

Empirische Studie:  
Unterschiede im Lernerfolg und  
Unterschiede im subjektiven Erleben  
des Unterrichts  
von Schülerinnen und Schülern  
im Informatik-Anfangsunterricht  
(11. Klasse Berufliches Gymnasium)  
in Abhängigkeit von der  
zeitlichen Reihenfolge der Themen  
(OOP-First und OOP-Later)

Dissertation

zur Erlangung des akademischen Grades eines  
Doktors der Naturwissenschaften (Dr. rer. nat.)

Fachbereich: Mathematik und Informatik

Institut: Informatik

Verfasser: Albrecht Ehlert

Jahr: 2012

1. Gutachter: Carsten Schulte, Freie Universität Berlin

2. Gutachter: Andreas Schwill, Universität Potsdam

Disputation: 13.01.2012

## **Kurzfassung**

In dieser empirischen Studie wurden zwei verschiedene fachdidaktische Vorgehensweisen zum Unterrichten der objektorientierten Programmierung (OOP) im Informatik-Anfangsunterricht verglichen: das so genannte OOP-First-Vorgehen (basierend auf den Ideen zu objects-first) mit dem so genannten OOP-Later-Vorgehen. Die Fachtermini „OOP-First“ und „OOP-Later“ wurden dabei hergeleitet bzw. definiert und eine neue Betrachtungsweise in die objects-first-Diskussion eingebracht: Das OOP-First-Vorgehen und das OOP-Later-Vorgehen unterscheiden sich (fast) nur in der sequenziellen Abfolge der Themen, die Themen selber sind vergleichbar.

Es wurden zwei Schulklassen (11. Klasse, Berufliches Gymnasium, Oberstufenzentrum Informations- und Medizintechnik) ein Jahr lang in zweierlei Hinsicht untersucht.

1. Der Lernerfolg nach einem Schuljahr, aufgeschlüsselt in verschiedene Themen:

- Einführung in die objektorientierte Programmierung (Klassenimplementierung und Objekterzeugung)
- Variablen (Attribute), fundamentale Datentypen und Steuerstruktur Sequenz
- Prozeduren (Methoden)
- Steuerstruktur Selektion und Steuerstruktur Iteration
- Spezielle Datentypen: Arrays und Strings
- Vererbung
- Assoziation
- Zusatzthemen: „Dynamische OOP“ und „Objektorientierte Modellierung“

Die Forschungsfrage lautete dabei: „Gibt es (signifikante) Unterschiede im Lernerfolg der Schülerinnen und Schüler im Informatik-Anfangsunterricht in Abhängigkeit von einem OOP-First- bzw. OOP-Later-Vorgehen?“.

2. Das subjektive Erleben (mit neun Momentaufnahmen innerhalb eines Schuljahres) der Schülerinnen und Schüler im Informatik-Anfangsunterricht, bezogen auf drei Dimensionen:

- Emotionale Dimension
- Kognitive Dimension
- Motivationale Dimension

Die Forschungsfrage lautete dabei: „Gibt es (signifikante) Unterschiede beim subjektiven Erleben des Informatik-Anfangsunterrichts der Schülerinnen und Schüler in Abhängigkeit von einem OOP-First- bzw. OOP-Later-Vorgehen?“.

In dieser Arbeit werden das Studiendesign hergeleitet, die Implementierung der Studie erläutert, die Resultate der empirischen Studie vorgestellt, Interpretationen der Ergebnisse vorgenommen und fachdidaktische Auswirkungen diskutiert (siehe auch: [Eh105] [Eh106] [Eh107a] [Eh107b] [Eh107c] [Eh108b] [ES07a] [ES07b] [ES09a] [ES09b] [ES09c] [ES09d] [ES10a] [ES10b]).

### **Schlagwörter:**

Informatikunterricht, objects first, objects later, empirische Studie, gymnasiale Oberstufe, OOP, OOP-First, OOP-Later, objektorientierte Programmierung.

## Abstract

Two different approaches to OOP in computer science education for beginners were compared in this empirical study: The objects-first approach vs. the objects-later approach.

First the technical terms OOP-first and OOP-later were defined and then a new angle on the discussion about objects-first vs. objects-later was introduced: Basically, the two different approaches only differ in the order of the topics presented, whereas the topics as such do not vary.

Two classes (grade 11) of Berlin's largest IT Secondary School (Oberstufenzentrum Informations- und Medizintechnik) were studied on an empirical basis over a period of one year in two different respects.

1. Learning success after one year of computer science education, broken down into various topics:

- Introduction to object oriented programming (declaring classes and instantiating objects)
- Fundamental data structures and control structure sequence
- Procedures (methods)
- Control structure selection and control structure loop
- Special data structures: arrays and strings
- Inheritance
- Association
- Additional topics: OOP (dynamically) and object oriented modeling (OOM)

The basic question to be answered here was: „Are the two approaches (OOP-first and OOP-later) really different in terms of learning success?“.

2. The personal (subjective) experience (snap-reading method applied nine times during the year) of the students concerning their computer science education, divided into three dimensions:

- Emotional dimension
- Cognitive dimension
- Motivational dimension

The basic question to be answered here was: „Are the two approaches (OOP-first and OOP-later) really different in terms of personal (subjective) experience?“.

In this paper the idea and the design of the study is explained, the way it was carried out is described, its results are presented, analyzed and, finally, discussed with regard to the methods of computer science education in general (q. v.: [Ehl05] [Ehl06] [Ehl07a] [Ehl07b] [Ehl07c] [Ehl08b] [ES07a] [ES07b] [ES09a] [ES09b] [ES09c] [ES09d] [ES10a] [ES10b]).

### Keywords:

Computer science education, objects first, objects later, empirical study, high school, CS0, CS1, computing education research, OOP-first, OOP-later, object oriented programming.

## Vorwort

Ich habe meinen Schuldienst als Lehrer im Jahr 1987 (siehe Lebenslauf im Anhang) am Oberstufenzentrum Nachrichtentechnik in Berlin-Wedding begonnen. Seit dem Jahr 2001 bin ich am neugegründeten Oberstufenzentrum Informations- und Medizintechnik ([www.oszimt.de](http://www.oszimt.de)) in Berlin-Neukölln tätig. In der Regel unterrichte ich zur Hälfte in der Berufsschule und zur anderen Hälfte am Beruflichen Gymnasium.

Von Anfang an habe ich in der Einführungsphase (im Rahmen des Unterrichts Technik/Informatik) am Beruflichen Gymnasium des OSZ IMT erlebt, dass sich viele meiner Schülerinnen und Schüler schwer getan haben mit der „Einführung in die (objektorientierte) Programmierung“. Nach einem Unterrichtsjahr konnten einige Schüler immer noch nicht den Unterschied zwischen einem Objekt und einer Klasse erläutern.

Als engagierter Lehrer erhoffte ich mir durch drei Fortbildungen und einen Workshop ab dem Sommer 2005 Lösungsansätze für die vorhandenen Probleme zu erhalten:

- „Fujaba-Einstieg (OOM)“ von Carsten Schulte und Ira Diethelm am 28. und 29.07.2005 an der FU Berlin
- „OOP von Anfang an: Objektorientierter Einstieg mit NIKI, dem JAVA-Roboter“ von Alexander Dietz und Bernd Kokavec am 26.10.2005
- „BlueJ-Workshop“ im Rahmen der 6. GI-Fachtagung von Alexander Dietz am 23.02.2006 in der HU Berlin
- „OOP mit BlueJ und Java im Informatikunterricht“, LISUM-Fortbildung von Alexander Dietz, Ralf Punkenburg und Mathias Müller am 06.03.2006

Alle Ansätze erschienen mir (bezogen auf meine Schüler) nicht sehr geeignet: Sie konnten m. E. die Lern-Probleme meiner Schüler nicht lösen. Dies war der Grund, warum es zu einer regen Diskussion mit meinem späteren Doktorvater (Prof. Dr. Schulte) gekommen ist, die am Ende in diese Arbeit mündete.

Im Laufe der empirischen Studie habe ich viele Vorträge auf Konferenzen gehalten und mehrere Papiere veröffentlicht. Die Rückmeldungen der Hörer und die Gutachten der Papiere haben zu einer Erhöhung der Qualität der Dissertation geführt.

Motivierend war auch, dass meine Forschungsarbeit auf internationalen Kongressen auf Interesse und Anerkennung stieß:

- „Highly relevant – this is perhaps the most discussed question in computing education research today, at least among practitioners“ (Review 2, ICER 2009)
- „Very significant. There has been too much debate and not enough experimentation“ (Review 1, ICER 2009)
- „It is pretty well linked to existing literature“ (Review 4, ICER 2009)
- „The approach used in this paper – changing the order in which material is presented – is both original and is a very interesting empirical approach“ (Review C, ITiCSE 2010)

Fünf Jahre nach meinen o. g. Fortbildungs-Erfahrungen besuchte ich am 23.02.2010 (im Rahmen des Informatiktages Berlin-Brandenburg) den Workshop „Einführung in die objektorientierte Programmierung“, weil dort auch häufig diskutierte didaktische Probleme wie OOP von Anfang an, Stellenwert der Modellierung und Wahl der Programmiersprache vorgestellt werden sollten. Ich wollte wissen, wie eine Fortbildung im Jahr 2010 die Frage nach „OOP-First“ beantworten würde. Zu meiner Überraschung wurde an dieser Stelle vom

vortragenden Kollegen J. Penon keine Antwort gegeben, sondern auf einen Fachmann im Publikum verwiesen: auf mich. So ändern sich die Zeiten.

Die vorliegende Dissertationsschrift hatte zeitlich gesehen für mich verschiedene Phasen. Nach dem Erwerben einer informatischen Fachkompetenz in über 20-jähriger Unterrichtspraxis, der Durchdringung der Probleme beim Informatik-Anfangsunterricht (speziell auch bei der OOP) ergaben sich für mich ab dem Sommer 2005 die hier vorliegenden Forschungsfragen. Darauf aufbauend entwickelte ich das empirische Forschungsdesign, die Fragebögen und die Tests. Der praktische Teil der Studie, also die Durchführung der empirischen Untersuchung, erfolgte zwei Jahre lang (von Sommer 2006 bis Sommer 2008) an meiner Schule in Berlin. Danach wurden die Daten ausgewertet und interpretiert. Parallel dazu habe ich in den letzten fünf Jahren Papiere für Konferenzen eingereicht, Vorträge gehalten und an dieser Dissertationsschrift geschrieben. Und dies alles neben der vollen Berufstätigkeit als Lehrer und als Fachbereichsleiter Informationstechnik am OSZ IMT.

Ich möchte mich bei einigen Personen des Oberstufenzentrums Informations- und Medizintechnik bedanken:

- Der Schulleitung (Herrn Brösemann und Herrn Seidel) und der Abteilungsleitung der gymnasialen Oberstufe (Herrn Ansorge und Frau Woffleben) für die Unterstützung bei der Studiendurchführung.
- Dem Informatik-Kollegen Francioni, der die OOP-First-Klasse betreute und bereit war, sequenziell die einzelnen Themen zu unterrichten und die Fragebögen und Tests in die Klasse zu geben.
- Den Deutsch-Kolleginnen Richter und Werner und den Deutsch-Kollegen Dr. Stang, Schifferings, Lehmann und Offenbacher (OSZ KIM) für das Korrekturlesen.

Für die Hilfe bei Fragestellungen zur Schreibweise in meinen englischen Beiträgen gilt mein Dank:

- Der Kollegin Denzau-Hoffmann (OSZ IMT), dem Kollegen Gretenkord (OSZ IMT), dem Kollegen Hannay (OSZ KIM) und der Sekretärin Schild (FU Berlin).

Natürlich gilt mein Dank auch allen Schülerinnen und Schülern (der Klassen OG61, OG62, OG71 und OG72), die freiwillig an dieser Studie teilgenommen haben.

Dann bedanke ich mich bei den Teilnehmern des Doktorandenkolloquiums IDDI, deren kritische Diskussionsbeiträge über mehrere Jahre hinweg sehr hilfreich waren und in dieser Arbeit berücksichtigt wurden:

- Hendrik Büdding, Ira Diethelm, Michael Dohmen, Stefan Freischlad, Ulrich Kiesmüller, Garbor Kiss, Maria Knobelsdorf, Lutz Kohl, Ralf Romeike, Kirsten Schlüter, Peer Stechert und Bernhard Wiesner.

Im Rahmen meiner Forschung gab es einen interessanten Austausch mit vielen Professoren, drei seien hier beispielhaft genannt. Ich spreche meinen Dank aus an die Professoren:

- Elmar Souvignier, Universität Münster
- Detlef Sembill, Universität Bamberg
- Werner Hartmann, Eidgenössische Technische Hochschule Zürich

Natürlich bedanke ich mich auch bei meinen Freundinnen, Freunden und der Familie, die mich unterstützt und meine eingeschränkte Zeit für soziale Kontakte akzeptiert haben.

Zu guter Letzt bedanke ich mich bei dem Menschen, ohne den es diese Arbeit nicht gäbe. Er hat mich am meisten gefördert und gefordert, hatte immer Zeit für mich und hat durch seine hohe Kompetenz, seine ruhige Art und sein motivierendes Verhalten dafür gesorgt, dass ich diese Arbeit bis zum Ende durchgehalten habe. Ich bedanke mich sehr herzlich bei meinem Doktorvater Prof. Dr. Carsten Schulte!

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Fachdidaktische Probleme und Relevanz der Studie	1
1.2	OOP-First und OOP-Later	7
1.2.1	Unterschiede zwischen objects-first und objects-later	7
1.2.2	Gleichartige Themen bei beiden Ansätzen	9
1.2.3	Definition: OOP-First und OOP-Later	12
1.2.4	Argumente für beide fachdidaktischen Vorgehensweisen	14
1.3	Zusammenfassung	16
<b>2</b>	<b>Stand der fachdidaktischen (objects-first) Diskussion, Forschungsfrage und Forschungshypothesen</b>	<b>17</b>
2.1	Deutsche Beiträge, speziell zur fachdidaktischen (objects-first) Diskussion	17
2.2	Internationale Beiträge, speziell zu bisherigen empirischen Ergebnissen der objects-first-Debatte (auch im Hinblick auf den Lernerfolg)	20
2.3	Beiträge zum subjektiven Erleben von (Informatik-) Unterricht	25
2.4	Forschungsfrage und Forschungshypothesen	28
2.4.1	Randbedingungen und Aufspaltung der Forschungsfrage	28
2.4.2	Objektiver Aspekt: Unterschiede im Lernerfolg	30
2.4.3	Subjektiver Aspekt: Unterschiede im emotionalen, kognitiven und motivationalen Erleben	31
2.4.4	Erstes Exposé	32
2.4.5	Ursprüngliche Thesen	33
2.4.6	Forschungshypothesen	38
2.5	Zusammenfassung	41

<b>3</b>	<b>Ziele, Forschungsdesign und Implementierung der empirischen Studie</b>	<b>43</b>
3.1	Ziele der empirischen Studie	43
3.2	Forschungsdesign der empirischen Studie	44
3.3	Implementierung der empirischen Studie	47
3.3.1	Randbedingung: Der Unterricht am Beruflichen Gymnasium des OSZ Informations- und Medizintechnik	48
3.3.2	Implementierung in Bezug auf das Forschungsdesign	49
3.3.3	Zeitliche Dimension der empirischen Studie (Untersuchungsplan)	51
3.3.4	Kommunikationsphase: Veröffentlichungen im Kontext der Studie	52
3.3.5	Variablen der empirischen Studie	53
3.3.6	Verteilung der Schüler auf die OOP-First- bzw. OOP-Later-Klasse	54
3.3.7	Pseudonymisierung der einzelnen Schülerinnen und Schüler	56
3.3.8	Bestimmung der Themen und ihrer zeitlichen Reihenfolge	57
3.3.9	Festlegung der Unterrichtsmethoden, der Arbeitsmittel und des Medieneinsatzes	61
3.3.10	Randbedingungen zur Auswahl der unterrichtenden Lehrer	63
3.3.11	Synchronisation der beiden Lehrer bzw. des Unterrichts	64
3.3.12	Vortest, Vergleichstest und Nachhaltigkeitstest	68
3.3.13	Fragebögen zum subjektiven Erleben	70
3.3.14	Vorstudie und deren Lehren für die Hauptstudie	71
3.3.15	Problemfelder: Lehrer, Vortest und fehlende Schüler	73
3.4	Zusammenfassung	76
<b>4</b>	<b>Lernerfolg: Lernziele, Indikatoren und Tests</b>	<b>77</b>
4.1	Beschreibung der Lernziele und Ableitung der Indikatoren	77
4.2	Beschreibung themenübergreifender Fachkompetenzen	90



4.3	Tests: Vergleichstest, Vortest und Nachhaltigkeitstest	93
4.3.1	Vergleichstest (Post-Test) in der Vorstudie und in der Hauptstudie in Bezug zu den abgefragten Themen	94
4.3.2	Vergleichstest (Post-Test) in der Hauptstudie	98
4.3.3	Vortest (Pre-Test) in der Hauptstudie	100
4.3.4	Nachhaltigkeitstest (Follow-Up-Test) in der Hauptstudie	102
4.3.5	Kritische Reflexion der Tests, speziell des Vergleichstests der Hauptstudie	105
4.4	Zusammenfassung	111
<b>5</b>	<b>Subjektives Erleben des Unterrichts: Dimensionen, Frage-Bereiche und Fragestellungen</b>	<b>113</b>
5.1	Vorgaben für den Fragebogen	113
5.1.1	Erlebnis-Dimensionen	113
5.1.2	Frage-Bereiche	115
5.1.3	Kategorisierung der Fragen mit Hilfe einer Matrix	115
5.1.4	Skalierung der Antwortmöglichkeiten	116
5.2	Fragestellungen	117
5.2.1	Emotionale Fragestellungen	117
5.2.2	Kognitive Fragestellungen	120
5.2.3	Motivationale Fragestellungen	123
5.2.4	Kategorisierungs-Matrix	124
5.3	Zusammenfassung	125
<b>6</b>	<b>Ergebnisse und Auswertung der empirischen Studie</b>	<b>127</b>
6.1	Statistische Grundlagen	127
6.2	Vorstudie: Ergebnisse und Interpretationen	130
6.2.1	Objektiver Aspekt	130
6.2.2	Subjektiver Aspekt	132
6.2.3	Zusammenfassung	142

6.3	Hauptstudie: Der Lernerfolg nach einem Schuljahr	143
6.3.1	Vortest (Pre-Test)	143
6.3.2	Vergleichstest (Post-Test)	144
6.3.3	Nachhaltigkeitstest (Follow-Up-Test)	146
6.3.4	Verlauf des Lernerfolgs	148
6.3.5	Ausgewählte Lerngruppen	149
6.3.6	Zusammenfassung	152
6.4	Hauptstudie: Das emotionale, kognitive und motivationale Erleben	153
6.4.1	Ergebnis-Überblick	153
6.4.2	Emotionale Dimension	155
6.4.3	Kognitive Dimension	161
6.4.4	Motivationale Dimension	171
6.4.5	Subjektives Erleben	174
6.4.6	Spezielle Lerngruppen	177
6.4.7	Zusammenfassung	179
6.5	Zusammenfassung	181
<b>7</b>	<b>Fachdidaktische Diskussion: Spezielle Fragestellungen, Auswertungen und Interpretationen</b>	<b>185</b>
7.1	Die Themen	185
7.1.1	Konnte die Themenreihenfolge wie geplant unterrichtet werden?	186
7.1.2	Konnten die Befragungen zu den Themen zeitlich wie geplant erfolgen?	186
7.2	Die Problemfelder	187
7.2.1	Waren die Lehrer wirklich gleichartig?	187
7.2.2	Hat der Vortest die Vermutung über die Vorkenntnisse der Schüler bestätigt?	192
7.2.3	Gab es Probleme mit fehlenden Schülern?	193

7.3	Der Lernerfolg	194
7.3.1	Welche Themen sind schwierig, welche einfach?	195
7.3.2	Sind prozedurale oder objektorientierte Themen schwieriger?	197
7.3.3	Warum sind Assoziationen und Felder die schwierigsten Themen?	198
7.3.4	Kann OOM in OOP-Kursen vermittelt werden?	200
7.4	Das subjektive Erleben	202
7.4.1	Hat das subjektive Erleben Einfluss auf den Lernerfolg?	203
7.4.2	Welche Themen werden als einfach, welche Themen als schwierig empfunden?	204
7.4.3	Hat das Schwierigkeits-Empfinden etwas mit dem tatsächlichen Schwierigkeitsgrad zu tun?	205
7.4.4	Gibt es Berührungspunkte zu vergleichbaren Untersuchungen?	207
7.4.5	Warum sind die Themen bei OOP-Later besser vernetzt?	209
7.4.6	Erleben die OOP-Later-Schüler den Unterricht wirklich subjektiv angenehmer?	210
7.5	Ergebnisse im Kontext zu den Forschungsfragen und Forschungshypothesen	213
7.5.1	Können die ursprünglichen Thesen bestätigt werden?	213
7.5.2	Können die Forschungsfragen beantwortet werden?	215
7.5.3	Können die Forschungshypothesen bestätigt werden?	217
7.5.4	Gibt es Schlussfolgerungen für den Informatik-Anfangsunterricht?	218
7.5.5	Gibt es interessante Nebenaspekte?	220
7.5.6	Gibt es Einschränkungen bei der Gültigkeit?	221
7.5.7	Gibt es weiterführende Forschungsfragen?	222
7.6	Zusammenfassung	224
	<b>Literaturverzeichnis (Quellenangabe)</b>	<b>225</b>
	<b>Abbildungsverzeichnis</b>	<b>235</b>

<b>Anhang</b>	<b>243</b>
A) Erstes Exposé (Sommer 2006)	243
B) Vorhandener Kernarbeitsplan vor der Studie (2006)	247
C) Vergleichstests vor der Studie (2004)	249
D) Fragebogen zum subjektiven Erleben der Schüler (Vorstudie 2006/2007)	259
E) Daten zum subjektiven Erleben der Schüler (Vorstudie 2006/2007)	260
F) Vergleichstest (Post-Test) inkl. Löser (Vorstudie 2006/2007)	269
G) Ergebnisse des Vergleichstests (Vorstudie 2006/2007)	283
H) Promotionsvorschlag (Februar 2008)	291
I) Auswahl der Stichproben (Schülerdaten, Hauptstudie 2007/2008)	302
J) Eingangsbefragung bezüglich der Computerkenntnisse (Hauptstudie 2007/2008)	303
K) Fragebogen zum subjektiven Erleben der Schüler (Hauptstudie 2007/2008)	304
L) Daten zum subjektiven Erleben der Schüler (Hauptstudie 2007/2008)	306
M) Vergleichstest (Post-Test) inkl. Löser (Hauptstudie 2007/2008)	315
N) Ergebnisse des Vergleichstests (Hauptstudie 2007/2008)	329
O) Nachhaltigkeitstest (Follow-Up-Test) inkl. Löser (Hauptstudie 2007/2008)	335
P) Ergebnisse des Nachhaltigkeitstests (Hauptstudie 2007/2008)	350
Q) Vortest (Pre-Test) 2008: erste Gruppe	358
R) Vortest (Pre-Test) 2008: zweite Gruppe	362
S) Ergebnisse des Vortests 2008	367
T) Informations- und Arbeitsblätter zu den Themen 8 und 9	369
U) Google-Ergebnisse: OOP first und OOP later (14.01.2010)	383
<b>Zusammenfassung der Ergebnisse</b>	<b>385</b>
<b>Lebenslauf</b>	<b>387</b>
<b>Eigenständigkeitserklärung</b>	<b>389</b>

# 1 Einführung

## 1.1 Fachdidaktische Probleme und Relevanz der Studie

Obwohl das objektorientierte Paradigma seit über zwei Jahrzehnten (vgl. antiquarisches Register des LOG-IN-Verlags [LI09]) in den Informatikunterricht in Schulen und Hochschulen Einzug gehalten hat, gibt es bis zum heutigen Tag immer wieder Beiträge, die auf die Fragestellung abzielen, wann und wie die objektorientierte Programmierung bzw. die objektorientierte Modellierung in den informatischen Lernablauf einfließen soll [ES09c, S. 121]. Während z. B. Ira Diethelm schon im Titel ihrer Dissertationsschrift [Die07] fordert: „Strictly models and objects first“ entscheidet sich Helmut Balzert wieder für einen strukturierten und prozeduralen Einstieg in die Programmierung (z. B. mit seinem Lehrbuch: „Java, der Einstieg in die Programmierung“ [Bal08]) fernab des objektorientierten Paradigmas (vgl. Abbildung 10).

Viele dieser Beiträge sind m. E. ein Indiz dafür, dass es keinen Konsens darüber gibt, wann und wie die objektorientierte Sichtweise den Informatik-Anfängern nahe gebracht werden soll, z. B.

- erst nach einem imperativen Einstieg,
- oder gleich von Anfang an, aber mit dem Schwerpunkt auf der objektorientierten Modellierung (OOM),
- oder gleich von Anfang an, aber mit dem Schwerpunkt auf der objektorientierten Programmierung (OOP).

Durch die folgenden angeführten Veröffentlichungen soll einerseits kurz veranschaulicht werden, wie vielschichtig die fachdidaktische Diskussion über objects-first ist. Es gibt z. B. Beiträge darüber,

- wie ein Informatik-Anfangsunterricht im Allgemeinen aussehen könnte,
- was die Vorteile (oder Nachteile) eines objects-first-Vorgehens sind,
- welches Software-Werkzeug oder welche Programmiersprache am besten geeignet ist,
- oder welche Beispiele bzw. Aufgaben, Materialien und didaktische Methoden Anwendung finden sollten.

Andererseits soll deutlich werden, dass Veröffentlichungen, die durch empirische Untersuchungen belegt sind, sich in der Minderheit befinden.

Die beiden ersten hier genannten Beiträge befassen sich allgemein mit den Problemen von Informatik-Anfangsunterricht.

Alexander Dietz, Jürgen Frank und Ralf Punkenburg [DFP11] geben 2011 für ihren Fortbildungskurs „Objektorientierte Programmierung mit Java im Informatikunterricht“ für Berliner Lehrkräfte folgende Inhaltsangabe:

„Im Anfangsunterricht fällt es Schülerinnen und Schülern oft schwer, Algorithmen im Kleinen und die Grundlagen von OOA, OOD, OOP (Klasse, Attribut, Methode, Beziehungen) zu verstehen, zu verbinden und anzuwenden. Diese Fortbildung thematisiert Möglichkeiten zur Bewältigung dieser Schwierigkeiten unter Verwendung der Programmiersprache Java anhand typischer Aufgabenstellungen aus den ersten beiden Lernjahren. Der Einsatz von Entwicklungsumgebungen, wie z. B. JavaKara, Greenfoot, BlueJ oder JavaEditor wird erprobt und diskutiert.“ [DFP11].

Das Autorenteam der Albert-Ludwigs-Universität Freiburg und der Eberhard-Karls-Universität Tübingen [BDua09] schreibt in seiner Zusammenfassung zum Vortrag „Auf dem Weg zu einer robusten Programmierausbildung“:

„Die gelungene Durchführung einer Vorlesung „Informatik I – Einführung in die Programmierung“ ist schwierig, trotz einer Vielfalt existierender Materialien und erprobter didaktischer Methoden. Gerade aufgrund dieser vielfältigen Auswahl hat sich bisher noch kein robustes Konzept durchgesetzt, das unabhängig von den Durchführenden eine hohe Erfolgsquote garantiert.“ [BDua09, S. 67].

Die nächsten Beiträge befassen sich mit der Frage, wann die objektorientierte Sichtweise in den Informatik-Anfangsunterricht einfließen sollte und welche Software-Werkzeuge geeignet sind.

Bernhard Koerber [Koe04] schreibt 2004 in einem Editorial zu einem LOG-IN-Doppelheft [LI04] mit dem Schwerpunktthema „Objektorientiertes Modellieren und Programmieren“:

„Sicherlich bleiben auch noch einige Fragen offen, beispielsweise welche Software-Werkzeuge die notwendigen Lernprozesse längerfristig gesehen tatsächlich unterstützen [...]. Und es muss weiterhin darüber nachgedacht werden, ob schon von der ersten Stunde an in die objektorientierte Sichtweise eingeführt werden sollte.“ [Koe04, S. 3].

Warum wird über diese Sachverhalte so viel diskutiert? Es scheint Schwierigkeiten beim Informatik-Anfangsunterricht (evtl. auch bei der Vermittlung der OOP) zu geben.

So schreibt z. B. Siegfried Spolwig [Spo01], eigentlich ein Befürworter des objects-first-Ansatzes, im Jahr 2001:

„In diesem Halbjahr hatte ich eine überwiegend schwache Gruppe und bin deshalb mit diesem Ansatz zum ersten Mal trotz sorgfältiger Vorbereitung grandios gescheitert, weil die Schüler völlig überfordert waren.“ [Spo01].

Um die Probleme in den Griff zu bekommen, gibt es inzwischen eine Vielzahl von Tools, die den Einstieg in den objektorientierten Unterricht für die Schülerinnen und Schüler vereinfachen sollen, z. B. BlueJ [BK06] oder Karel D. Robot, ein objektorientierter Delphi-Karel [Spo05].

Mit ihrem BlueJ-Tool wollen David Barnes und Michael Kölling [BK06] das objects-first-Vorgehen unterstützen:

„Ein Student kann als erste Aktivität ein Objekt erzeugen und seine Methoden aufrufen! Weil Benutzer Objekte direkt erzeugen und manipulieren können, können Konzepte wie Klassen, Objekte, Methoden und Parameter ohne weiteres direkt diskutiert werden, bevor die erste Zeile Java-Quelltext betrachtet werden muss.“ [BK06, S. 18].

Auch Marco Thomas [Tho04] lobt das BlueJ-Tool:

„Die bisherigen Erfahrungen zur Einführung in die Objektorientierung mit BlueJ sind insgesamt recht positiv.“ [Tho04, S. 31].

Gerhard Röhner [Röh09] schreibt im Dezember 2009 hingegen über viele Nachteile von BlueJ, u. a.:

„Ein wirklich gravierender Nachteil von BLUEJ ist jedoch die fehlende Unterstützung des Modellierungsprozesses.“ [Röh09, S. 61].

Danach preist er aber seinen Java-Editor [Röh10] als bestens geeignete Entwicklungsumgebung an. Ein Indiz dafür, dass die Werkzeug-Diskussion bis heute anhält.

Die Werkzeuge scheinen aber nicht alle Probleme zu lösen. So schreibt Alfred Heubaum [Heu04] schon im Jahr 2004:

„Man macht es sich zu leicht, wenn man [...] OOP als „Schlüssel zum Erfolg“ anpreist, auf die professionelle Software-Entwicklung verweist und ein analoges Vorgehen für den Unterricht fordert. Die angeblichen Vorteile müssen sich auch aus der (beschränkten) Sicht der Schüler und ihren Erfahrungen im Unterricht ergeben. [...] Die Schwierigkeiten bzw. Nachteile werden auch nicht dadurch behoben, dass man Werkzeuge wie BlueJ oder ein „didaktisches System“ einsetzt.“ [Heu04, S. 76].

Auch die Fragen nach dem Stellenwert der Modellierung und nach der Wahl der Programmiersprache kommen in der fachdidaktischen Diskussion bis zum heutigen Tag vor.

So wurde auf dem Informatiktag Berlin-Brandenburg 2010 ein Workshop „Einführung in die objektorientierte Programmierung“ angeboten. Im Inhaltsverzeichnis [PFS10] heißt es u. a.:

„Häufig diskutierte didaktische Probleme wie OOP von Anfang an, Stellenwert der Modellierung, Wahl der Programmiersprache usw. werden vorgestellt.“ [PFS10].

Obwohl die objektorientierte Modellierung oft als ein zentraler Bestandteil eines Informatik-Kurses angesehen wird, ergibt eine Umfrage an den Informatik-Instituten deutschen Universitäten, dass die Algorithmik als deutlich wichtiger erachtet wird als die OOM. Zendler und Spannagel [ZS08] schreiben 2008:

„Especially the concept *algorithm* seems to play the most important role in computer science education. Interestingly, the concept *model* has a low rank among the central concepts. Although it is part of the list of central concepts, it seems not be as important as it is often suggested.“ [ZS08, S. 12].

Auch international hält die Diskussion an, ob „Objects Early“ sinnvoll ist oder nicht (vgl. z. B. Owen Astrachan und weitere Autoren [ABua05]: „Resolved: Objects Early has Failed“) und wie eine „Einführung in die Programmierung“ aussehen sollte.

A. Robins, J. Rountree and N. Rountree [RRR03] vergleichen 2003 Literatur zu „Learning and Teaching Programming: A Review and Discussion“ und sie kommen zu dem Schluss:

„With respect to teaching novices in CS1 type courses the goal is to foster deep learning in students. 4.1: Many students make very little progress in a

first programming course. 4.2: Various suggestions regarding course design and delivery have been made in the literature.” [RRR03, S. 163].

Auch Arnold Pears und weitere Autoren [PSua07] sichten im Jahr 2007 in „A Survey of Literature on the Teaching of Introductory Programming“ vorhandene Literatur und Studien zu diesem Thema und kommen zu dem Ergebnis:

„The goal of this paper has been to give an overview of research issues relating to the teaching of introductory programming, with specific reference to curriculum, pedagogy, and languages and tools for supporting learning. It has provided material from the body of discipline-specific research in each of these areas. We conclude that despite the large volume of literature in this area, there is little systematic evidence to support any particular approach. For that reason, we have not attempted to give a canonical answer to the question of how to teach introductory programming.“ [PSua07, S. 211].

Und Kim Bruce [Bru05] schreibt 2005 in einem Artikel, in dem er eine Diskussion zusammenfasst, die von Mitgliedern der Mailingliste (SIGCSE-members@acm.org) der SIGCSE [SIG10] zum Thema „How to teach Computer Science 1“ geführt wurde, folgendes:

„We are in one of those periodic difficult times where we are struggling to find an effective way of teaching introductory courses that will provide a strong education for our students.“ [Bru05, S. 116].

Raymond Lister und andere Fachdidaktiker [LBua06] beziehen sich auch auf diese Mailingliste und den ausführlichen Streit im Jahr 2004 um die Vor- und Nachteile des objects-first-Vorgehens. Ihrer Meinung nach wird deutlich, dass viele unterschiedliche didaktische Dimensionen in der Diskussion angesprochen werden und dass viele der Argumente auf individueller Anschauung und Lehr-Erfahrung beruhen.

Dies meint auch Carsten Schulte [Sch03b] im Jahr 2003 in seiner Dissertation:

„Zum Thema Objektorientierung im Informatikunterricht liegen keine empirischen Untersuchungen, sondern nur Erfahrungsberichte von Lehrenden vor. Erfahrungsberichte sind zwar einerseits eine informative Quelle, andererseits kann aus verschiedenen Gründen die Gültigkeit der Aussagen nur schlecht abgeschätzt werden.“ [Sch03b, S. 13].

Inzwischen liegen einige empirische Untersuchungen vor (vgl. z. B. [Dec03], [Bri04], [Reg06], [Die07] und [VZG07]).

Dies wird von Ira Diethelm und Carsten Schulte bestätigt. Sie schreiben 2010 in einem Vorwort zu Beiträgen des GI-Workshops „Didaktik der Informatik – Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik“ [DDua10]:

„Die Didaktik der Informatik mausert sich. [...] Im Vergleich über die Jahre zeigt sich, wie auch im Titel zum Workshop angedeutet wird, die zunehmende empirische Fundierung, die einhergeht mit einer Reflexion der



Möglichkeiten, Aufgaben und notwendigen zukünftigen Ausrichtung informatikdidaktischer Forschung.“ [DDua10, S. 5].

Die empirische Fundierung nimmt zwar zu, ist aber weiterhin nicht die Regel.

Theme	Paper count
Ability/aptitude/understanding	66 (40%)
Teaching/learning/assessment techniques	57 (35%)
Teaching/learning/assessment tools	14 (9%)
Teaching/learning theories & models	9 (6%)
Curriculum	6 (4%)
Gender issues	3 (2%)
Educational technology	2 (1%)
Recruitment, progression, pathways	2 (1%)
Cheating & plagiarism	2 (1%)
Research	2 (1%)
Accessibility	1 (<1%)

Abbildung 1: Themen von 164 veröffentlichten Papieren mit Bezug zu Untersuchungen von 6 verschiedenen internationalen Konferenzen zur informatischen Bildung in den Jahren 2005-2008 [SSua09, S. 96, Table 3]

Klassifiziert man z. B. die 979 Papiere von sechs bedeutsamen internationalen Konferenzen zur informatischen Bildung (ICER, ITiCSE, SIGCSE, ACE, Koli Calling und NACCQ) aus den Jahren 2005-2008 nach Themen, dann befinden sich 164 Papiere (17%) im Kontext von Untersuchungen („experiments, studies, or analysis“). Erfolgt hier wieder eine Aufschlüsselung nach Themen (siehe Abbildung 1), dann ist erkennbar, dass empirische Studien zu fachdidaktischen Fragestellungen eher die Ausnahmen sind [SSua09].

Die bisherigen Zitate sollen einleitend darauf hindeuten, dass die didaktischen Fragestellungen zur OOP bzw. zu objects-first immer noch aktuell sind. Mehr Beiträge zur deutschen und internationalen Diskussion finden sich in den Kapiteln 2.1 und 2.2.

Offensichtlich werden bei dieser Diskussion viele verschiedene didaktische Dimensionen und fachdidaktische Probleme angesprochen: Reihenfolge und Schwerpunkte der Themen, angemessene Werkzeuge, Unterrichtsbeispiele, Lehr- und Lernformen bis hin zu der Frage, ob die objects-first-Idee überhaupt sinnvoll ist oder nicht. Die Antwort fällt schwer: Zum einen durch die Vielfalt und die inneren Abhängigkeiten der diskutierten didaktischen Aspekte, zum anderen durch die fehlende Abschätzung, welche Aspekte den zentralen Unterschied zwischen objects-first und objects-later ausmachen [ES09c, S. 121].

Zusammenfassend über alle Papiere kann man feststellen, dass es viele Veröffentlichungen zu „objects-first vs. objects-later“ gibt, die von Beispielen berichten, dass das eine oder andere Vorgehen gut funktioniert, aber es gibt so gut wie keine generalisierenden Aussagen bzw. Ergebnisse.

Und damit ergibt sich die Relevanz dieser Studie: Meines Erachtens gab es noch kein empirisches Untersuchungs-Szenario, das einen fairen Vergleich zwischen den beiden fachdidaktischen Vorgehensweisen ermöglicht hat [ES09a, S. 16].

Hier einige Beispiele für einen möglicherweise unfairen Vergleich [ES09c, S. 122]:

Adrienne Decker [Dec03] kommt 2003 zum Schluss, dass der objects-first-Ansatz überlegen ist. Allerdings hat sie den Ansatz mit einem so genannten „Weniger-Objekte-Ansatz“ verglichen und vor allem bezüglich der so vermittelten objektorientierten Kenntnisse (mit Hilfe eines objektorientierten Tests) untersucht – da ist es wenig

verwunderlich, dass in der Gruppe, in der mehr Zeit für die Objektorientierung aufgewendet wurde, auch mehr Wissen zum Thema entstanden ist.

Stuart Reges [Reg06] kommt 2006 dagegen in einem empirischen Vergleich zu dem Schluss, dass die traditionelle Art des Unterrichtens dem objects-first-Ansatz deutlich überlegen sei: Bessere Leistungen, bessere Zufriedenheit der Lernenden, bessere Bewertung des Lehrenden, weniger Abbrecher. Allerdings vergleicht er von ihm selbst durchgeführte Kurse mit denen seines Vorgängers, den die Universität entlassen hat, um ihn, Reges, mit der ausdrücklichen Maßgabe anzustellen, die Anfangskurse didaktisch zu verbessern. Da sein Vorgänger im Gegensatz zum ihm einen objects-first-Ansatz vertreten hat, sieht er hier die Hauptursache – es könnte allerdings auch an anderen Dingen liegen: Möglicherweise ist das didaktische Konzept insgesamt besser, der Lehrende fähiger, das Kursmaterial besser, die Studenten fähiger (oder besser motiviert) oder die gestellten Anforderungen beim Examen geringer.

In einer Studie von Vilner, Zur und Gal-Ezer [VZG07] an der „Open University of Israel“ finden sich keine signifikanten Unterschiede zwischen prozeduralen und objektorientierten Vorgehensweisen. Die Studenten der Universität sind in zwei verschiedenen CS1-Kursen, in einem prozeduralen C++-Kurs und in einem objektorientierten Java-Kurs. Die Untersuchung hat aber Schwächen, weil zwei unterschiedliche Programmiersprachen eingesetzt werden und nicht wissenschaftlich nachgewiesen wird, dass beide untersuchte Gruppen vergleichbar sind. Ähnliche Schwächen hat m. E. die empirische Untersuchung von Susan Wiedenbeck und weiteren Autoren [WRua99]: In dieser wird untersucht, inwiefern die Studenten nach zwei verschiedenen Programmierkursen (Prozedural: Pascal, Objektorientiert: C++) im nächsten Semester kurze oder lange objektorientierte Programme schreiben können. Für kurze Programme (mit nur einer Klasse in C++) findet sich kein Unterschied zwischen den beiden Gruppen. Bei längeren Programmen (mit mehreren Klassen in C++) sind die Studenten wesentlich besser, die zuerst das prozedurale Paradigma kennen gelernt hatten.

Diese Studien machen verschiedene Anforderungen an eine empirische Untersuchung des objects-first-Ansatzes deutlich. Um einen fairen Vergleich zu ermöglichen, müssen drei Bedingungen erfüllt sein [ES09a, S. 16]:

1. Es müssen die Unterschiede zwischen den beiden fachdidaktischen Vorgehensweisen (objects-first und objects-later) definiert bzw. operationalisiert werden. Sind es die Themen, das methodische Vorgehen, das eingesetzte Material, die Programmiersprache, die eingesetzten Tools oder die Programmierübungen, die den Hauptunterschied zwischen beiden fachdidaktischen Vorgehensweisen ausmachen und fokussiert werden sollten? Die Antwort wird im nächsten Kapitel 1.2 herausgearbeitet: Der zentrale Unterschied ist die unterschiedliche Sequenzierung der Lerninhalte.
2. Das unterschiedliche fachdidaktische Vorgehen muss zur gleichen Zeit erfolgen. Dafür müssen zwei Schulklassen zur Verfügung stehen, die (statistisch) vergleichbar sind und parallel beschult werden können. Diese Bedingungen liegen am Oberstufenzentrum Informations- und Medizintechnik in Berlin vor (siehe Kapitel 3.2 und 3.3).
3. Es müssen möglichst viele (Einfluss-) Variablen kontrolliert bzw. für beide Klassen gleichartig gestaltet werden, so dass sie den gleichen Einfluss auf beide Schülergruppen nehmen. Die Aufzählung der einzelnen Variablen (Lehrer, Medien, Unterrichts-Methoden etc.) und die Antwort auf die Frage, wie die Variablen gleichartig gestaltet werden, finden sich im Kapitel 3.3 bei der Implementierung der Studie.

In der inzwischen 20-jährigen Diskussion fehlte also ein direkter Vergleich zwischen objects-first und objects-later in zwei vergleichbaren Kursen bzw. Klassen im selben Jahrgang, an derselben Bildungseinrichtung, mit vergleichbaren Schülern, mit derselben Programmiersprache, mit derselben Entwicklungsumgebung, mit einem seriösen Untersuchungsdesign und mit reduzierter bzw. stark kontrollierter Variablenanzahl. Daraus ergibt sich die Relevanz dieser Studie.

## 1.2 OOP-First und OOP-Later

In diesem Kapitel werden die Begriffe OOP-First und OOP-Later erläutert bzw. definiert. Hierfür müssen erst einmal die folgenden Fragen geklärt werden:

- Was versteht man unter objects-first bzw. objects-later und worin bestehen die Unterschiede?
- Welches sind (in der Regel) die Themen im Informatik-Anfangsunterricht?
- Wie ist (in der Regel) die zeitliche Reihenfolge der Themen?

Sind Antworten gefunden, kann es zu einer Definition von OOP-First und OOP-Later und zu einer ersten Diskussion über die Vor- und Nachteile beider fachdidaktischer Vorgehensweisen kommen.

Die nächsten Kapitel (1.2.1 bis 1.2.4) sind teilweise wörtlich übernommen aus dem Beitrag von Carsten Schulte und mir für die INFOS 2009 [ES09c].

### 1.2.1 Unterschiede zwischen objects-first und objects-later

Im Folgenden soll der Unterschied zwischen objects-first und objects-later kurz anhand eines Unterrichtsbeispiels bzw. Unterrichtsthemas (Steuerstruktur Iteration) veranschaulicht werden.

Bei objects-first wird die objektorientierte Sichtweise gleich an den Anfang des Informatikunterrichts gestellt. Dabei gibt es verschiedene Möglichkeiten. In einer (nicht repräsentativen) internationalen Studie, an der auch viele deutsche Informatik-Lehrerinnen und -Lehrer teilgenommen haben, ergaben sich als Ergebnis einer empirischen Klassifizierung die folgenden drei Unterscheidungen des objects-first-Begriffs [BS07, S. 23]:

1. Objekte benutzen: Es werden zunächst vorhandene Objekte benutzt und manipuliert, bevor implementiert wird.
2. Klassen schreiben: Von Anfang an werden Klassen definiert, implementiert und instanziiert, um das Paradigma zu vermitteln. Frühe Programmier-Erfahrungen mit dem objektorientierten Paradigma stehen im Mittelpunkt.
3. Konzepte: In Bezug auf die Diskussion in Deutschland könnte man auch vom objektorientierten Modellieren sprechen. Zunächst werden die abstrakten und generellen Ideen des Paradigmas vermittelt, wobei die objektorientierten Modelle im Mittelpunkt stehen.

Gemeinsam ist den verschiedenen Varianten, dass das objektorientierte Paradigma an den Anfang gerückt wird.

Zuerst werden mit Hilfe von Klassen Objekte erzeugt und benutzt. Dabei werden entweder vorgefertigte Klassen benutzt und instanziiert, oder Klassen modelliert, implementiert und instanziiert. Je nach Variante werden also mehr oder weniger stark die im Klassen-Konzept enthaltenen Themen, wie z. B. Attribute und Methoden, allgemein eingeführt, um diese dann später zu konkretisieren [ES09c, S. 123].

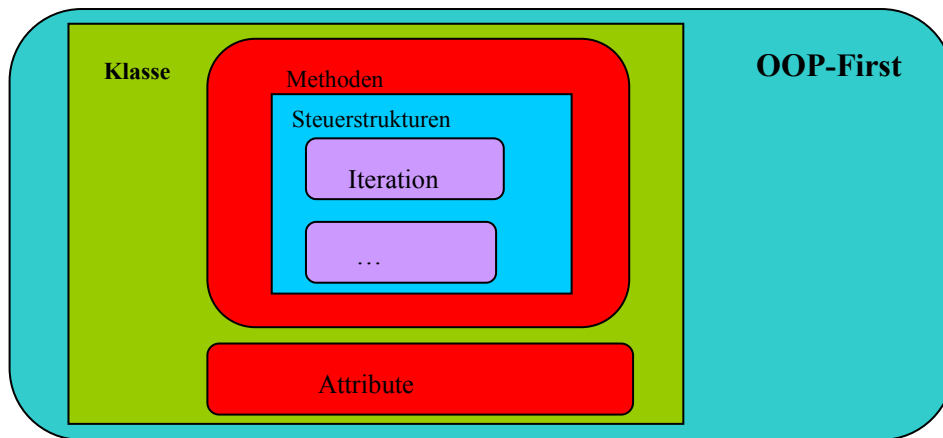


Abbildung 2: Thema Iteration beim OOP-First-Ansatz [ES09c, Abbildung 1, leicht modifiziert]

Die Abbildung 2 veranschaulicht diesen Zusammenhang am Beispiel der Steuerstruktur Iteration. Die Iteration ist Teil des Themas Steuerstrukturen, dieses ist ein Unterthema des Themas Methoden, welches mit dem Thema Attribute Teil des Themas Klasse ist.

Bei objects-later wird dagegen erst mit der prozeduralen bzw. imperativen Sichtweise in den Informatikunterricht eingestiegen. Wenn dann später im Unterricht auf die objektorientierte Sichtweise umgestiegen wird, kann auf vorhandenes Wissen aus der prozeduralen Programmierung zurückgegriffen werden.

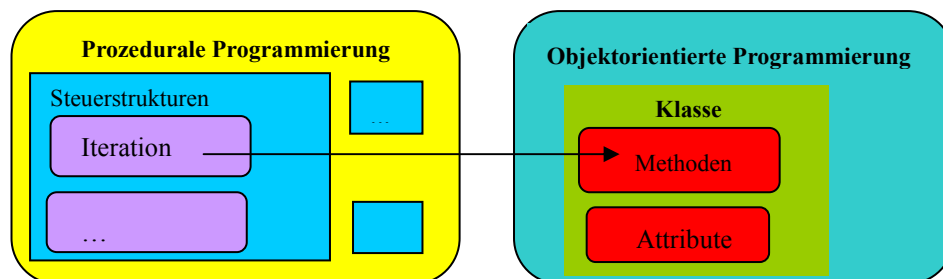


Abbildung 3: Thema Iteration beim OOP-Later-Ansatz [ES09c, Abbildung 2, leicht modifiziert]

Die Abbildung 3 veranschaulicht, dass beim objects-later-Ansatz beim Thema „Methoden einer Klasse“ z. B. auf dem vorhandenen Wissen über die Steuerstruktur Iteration aufgebaut werden kann.

Bei diesem objects-later-Ansatz gibt es für den Unterrichtenden Freiheitsgrade. Es stellt sich nämlich für ihn im Verlauf des Informatikunterrichts die Frage: Wann ist der Punkt gekommen, um von der prozeduralen Sichtweise auf die objektorientierte Sichtweise umzusteigen?

Zusammenfassend ist also der Hauptunterschied zwischen den beiden fachdidaktischen Vorgehensweisen, dass es bei objects-later im Gegensatz zu objects-first zu einem Paradigmenwechsel kommt – mit allen Vor- und Nachteilen.

## 1.2.2 Gleichartige Themen bei beiden Ansätzen

Das letzte Kapitel deutet darauf hin, dass der wesentlichste Unterschied vom objects-first-Ansatz zum objects-later-Ansatz die Reihenfolge der Lernthemen – und nicht das verwendete Werkzeug, die Unterrichtsmethode, die Programmiersprache etc. – ist [ES09c, S. 124]. Daher müssen für beide Ansätze gleichartige Themen benannt werden. Hierfür werden verschiedene Programmier-Lehrbücher (mit den verschiedensten Ansätzen) im Hinblick auf die einzelnen Programmier-Themen verglichen.

Hermes [Her96] schreibt schon 1996:

„Auf uns selbst angewiesen, bleibt wohl nichts anderes übrig, als die aktuelle Literatur zu sichten und eine Linie, einen roten Faden, in das Thema zu bringen.“ [Her96, S. 29].

Was steht aber in der aktuellen Literatur zur „Einführung in die Programmierung“? Wie sieht der rote Faden aus? Welches sind die Themen?

Auffällig ist, wenn man Informatik-Bücher vergleicht, wie unterschiedlich einerseits der rote Faden der Themen ist und wie unterschiedlich andererseits die (Haupt-) Kapitelüberschriften sind. In den Unterkapiteln finden sich aber dann doch vergleichbare Themen.

Adrienne Decker [Dec03] schreibt 2003:

„Students in both sections were taught the standard set of topics for a typical CS1, simple data types, assignment, basic arithmetic, selection, iteration, and basic use and operations with either arrays or collections. Both sections were using the same book [Sav01], which typical of many of the books currently in print presents the usual imperative concepts first, and then presents the notion of objects. Objects are presented and encapsulation is discussed briefly at the same time in the text. However, it is not until a few chapters later that the other key points of object-oriented programming, like inheritance or polymorphism, are presented. Both classes covered material up to, but not including the chapters on inheritance and polymorphism in the first semester.“ [Dec03, S. 241, Verweis von [16] auf [Sav01] geändert]

Das „Computing Curricula 2001 project“ schreibt in seinem Schlussbericht [CS01] unter der Überschrift „Introduction to Programming“:

„Topics include standard programming constructs, problem-solving strategies, the concept of an algorithm, and fundamental data structures (strings, arrays, and records) [...]“ [CS01, S. 166]

Bei der Revision dieses Reports im Jahr 2008 [CS08] wird unter dem Schlagwort „Fundamental Constructs“ u. a. Folgendes aufgeführt (CS08, S. 40): „variables“, „types“, „expressions“, „assignment“, „conditional and iterative control structures“ und „functions and parameter passing“.

Unter dem Thema „Data Structures“ finden sich u. a. [CS08, S. 41]: „representation of numeric data“, „arrays“, „representation of character data“ und „strings and string processing“.

Beim Thema „Object Oriented“ gibt es u. a. die folgenden Unterthemen [CS08, S. 42]: „object-oriented design“, „encapsulation and information-hiding“, „classes and subclasses“ und „inheritance“.

Exemplarisch für ein OOP-Later-Vorgehen soll hier das Inhaltsverzeichnis eines C#-Lehrbuchs [PW05] angeführt werden: Sprachgrundlagen von Visual C#.NET (Kapitel 6, enthält Variablen, Konstanten und Datentypen-Überblick), Datentypen und Operatoren (Kapitel 7), Kontrollstrukturen (Kapitel 8), Methoden (Kapitel 9), Komplexe Datentypen (Kapitel 10), Klassen und Objekte (Kapitel 11) und Vererbung (Kapitel 12).

Schwieriger wird es ein exemplarisches OOP-First-Vorgehen zu finden. Viele Lehrbücher mit dem Titel „Einführung in die objektorientierte Programmierung“ haben vor dem Kapitel „Klassen und Objekte“ doch die Kapitel „Datenstrukturen“ und „Kontrollstrukturen“. Den m. E. konsequentesten Weg gehen Barnes und Kölling [BK06]. Bei ihnen finden sich im 1. Kapitel („Objekte und Klassen“) u. a. die Themen „Methoden aufrufen“, „Parameter“ und „Datentypen“ und im 2. Kapitel („Klassendefinitionen“) etwas versteckt das Thema „Kontrollstruktur Auswahl“ (Kapitel 2.11: „Entscheidungen treffen: die bedingte Anweisung“) und richtig versteckt (gar nicht im Inhaltsverzeichnis aufgeführt, dort heißt es nur im Kapitel 4.8: „Komplette Sammlungen verarbeiten“) das Thema „Kontrollstruktur Schleife“ (Kapitel 4.8.1: „Die for-each-Schleife“, Kapitel 4.8.2: „Die while-Schleife“).

Ein spezielles Lehrbuch für Bildungseinrichtungen [Mis05] bezieht sich gar nicht auf ein spezielles Programmier-Paradigma, nennt sich einfach „Java 4 U: Programmentwicklung mit Java“ mit dem Kapitel „Java“ (Kapitel 1 enthält u. a. die Themen Datentypen, Variablen und Konstanten), dem Kapitel „Kontrollstrukturen“ (Kapitel 2 enthält u. a. die Themen Auswahl, Schleifen und Felder) und dem Kapitel „Das objektorientierte Konzept von Java“ (Kapitel 3 enthält u. a. die Themen Klassen, Methoden, modifizierte Attribute und Vererbung).

Helmut Balzert trennt die beiden Paradigmen auf in zwei unterschiedliche Lehrbücher.

Das Buch „Java: Der Einstieg in die Programmierung“ [Bal08] nimmt Bezug zur strukturierten und prozeduralen Programmierung und hat u. a. die Kapitel „Einfache Typen, ihre Werte und Operationen“ (Kapitel 3), „Kontrollstrukturen“ (Kapitel 4), „Felder“ (Kapitel 5) und „Prozeduren, Funktionen und Methoden“ (Kapitel 7).

Im Buch „Java 5: Objektorientiert programmieren“ [Bal06] lauten u. a. die Kapitel „Basiskonzepte im Detail“ (Kapitel 3), „Klassen benutzen und bereitstellen“ (Kapitel 4), „Assoziationen“ (Kapitel 5) und „Einfachvererbung“ (Kapitel 6).

Im Ergebnis finden sich bei allen Lehrbüchern immer wieder gleichartige Kapitel bzw. Sequenzen, kleine Unterschiede ausgenommen: Einmal finden sich die Steuerstrukturen im Kapitel „Anweisungen“ [KS07, S. 133-154], einmal ist der Datentyp Feld im Kapitel „Kontrollstrukturen“ [Mis05, S. 21-47], einmal entdeckt man die fundamentalen Datentypen im Unterkapitel „Variablen und Konstanten“ [Erl00, S. 33-64] und manchmal gibt es ein Zusatzthema, wie z. B. das Thema „Testen“ [Bal08, S. 239-256].

Somit können unabhängig von einem OOP-First- bzw. OOP-Later-Ansatz gleichartige Themen benannt werden [ES09c, S. 124]. Diese gleichartigen Themen sind:

- Einführung in die OOP, Klasse und Objekt
- Variablen (bzw. Attribute), Konstanten und fundamentale Datentypen
- Steuerstrukturen: Sequenz, Iteration und Selektion
- Prozeduren (bzw. Operationen, Methoden, Funktionen, Botschaften)

- Komplexe(re) Datentypen
- Vererbung
- Assoziation

Betrachtet man die typische zeitliche Reihenfolge der Themen bei einem „**OOP-First-Vorgehen**“ und einem „**OOP-Later-Vorgehen**“ (siehe Abbildung 4), sind die beiden Ansätze nur auf den ersten Blick sehr unterschiedlich (Hinweis: In der gesamten Arbeit wird OOP-First mit der Farbe „**Blassblau**“ und OOP-Later mit der Farbe „**Pflaume**“ gekennzeichnet.).

OOP-First	OOP-Later
Klasse und Objekt	Datentypen (und Variablen)
Attribute (inkl. Datentypen)	Steuerstrukturen
Methoden (inkl. Steuerstrukturen)	Prozeduren
Vererbung	Klasse und Objekt (inkl. Attribute und Methoden)
Assoziation	Vererbung und Assoziation

Abbildung 4: Typische Themenfolge von OOP-First und OOP-Later [ES09d, Folie 8, leicht modifiziert]

Auf den zweiten Blick erkennt man (siehe Abbildung 5), dass sie doch im Endeffekt die gleichen Themen, z. B. die Implementierung von Klassen und die Erzeugung von Objekten, genauso wie die Themen Daten- und Steuerstrukturen, enthalten.

OOP-First	OOP-Later
Klasse und Objekt	Datentypen (und Variablen)
Attribute (inkl. Datentypen)	Steuerstrukturen
Methoden (inkl. Steuerstrukturen)	Prozeduren
Vererbung	Klasse und Objekt (inkl. Attribute und Methoden)
Assoziation	Vererbung und Assoziation

Abbildung 5: Gleiche Themen bei OOP-First und OOP-Later [ES09d, Folie 8, leicht modifiziert]

Nur die Sichtweise ist eine andere:

- OOP-First: Das objektorientierte Paradigma steht von Anfang an im Mittelpunkt des Informatik-Anfangsunterrichts.
- OOP-Later: Das objektorientierte Paradigma baut auf dem prozeduralen bzw. imperativen Paradigma auf.

Die beiden Vorgehensweisen haben im Endeffekt also die gleichen Themen, nur die zeitliche Abfolge der Themen ist eine andere. Damit ist eine erste Definition von OOP-First und OOP-Later erfolgt, die im nächsten Kapitel vertieft werden soll.

Ein Hinweis soll aber vorher noch gegeben werden: Natürlich handelt es sich hier um eine Art Modellbildung, um die Anzahl der Variablen, die auf den Unterricht Einfluss nehmen (können), zu begrenzen. Für diese Studie werden ein OOP-First- und ein OOP-Later-Ansatz modellhaft definiert. Objects-first-Kurse in der Praxis werden sich abhängig von der Bildungseinrichtung, dem Lehrer, dem Land etc. teilweise von dem in dieser Studie untersuchten OOP-First-Kurs unterscheiden (z. B. gehen einige objects-first-Kurse nicht sequenziell sondern spiralförmig vor).

### 1.2.3 Definition: OOP-First und OOP-Later

Ausgehend von der Kategorisierung von Kapitel 1.2.1 (drei Varianten des objects-first-Begriffs) und den vorgegebenen Randbedingungen von Kapitel 1.1, kann das der Studie zugrundeliegende OOP-First-Unterrichtsvorgehen der zweiten objects-first-Variante zugeordnet werden. Der Fokus beim objects-first-Einstieg liegt auf der konkreten Programmierung von Klassen und der Erzeugung von Objekten. Daher wird über den Begriff „OOP“ (objektorientierte Programmierung) ein Hinweis auf diesen Programmierschwerpunkt gegeben.

Im Folgenden soll der Begriff *OOP-First* verwendet werden, wenn im Anfangsunterricht gleich mit dem Begriff Objekt bzw. Klasse eingestiegen wird und der Schwerpunkt eher in der Programmierung bzw. Implementierung als in der Modellierung liegt (dies ist auch die Abgrenzung zu objects-first, OOM und OO-First) [ES09c, S. 125]. Diese Schwerpunktbildung wird von Kortenkamp, Modrow, Oldenburg, Poloczek und Rabel [KMua09] unterstützt. Sie schreiben als Fazit in ihrem Beitrag zu „Objektorientierte Modellierung – aber wann und wie?“:

„Mit diesem Beitrag soll keineswegs die objektorientierte Modellierung aus der Schule verbannt werden. Es scheint aber sinnvoll, ihre Bedeutung etwas zu relativieren, wodurch sie eher (wieder) Ziel denn Ausgangspunkt der Beschäftigung mit Informatik wird.“ [KMua09, S. 46-47].

Die Begriffsbildung *OOP-First* wurde schon von anderen Autoren vorgenommen, so schreibt z. B. Ira Diethelm [Die07]:

„Eine andere Bedeutung von *objects-first* ist, mit objektorientierter Programmierung im Unterricht zu beginnen. Solche Konzepte kann man auch besser mit dem Namen *OOP-first* bezeichnen.“ [Die07, S. 22, kursive Schreibweise im Original].

Trotz des OOP-First-Vorgehens müssen natürlich an geeigneten Zeitpunkten auch „traditionelle Themen“ wie Datentypen und Steuerstrukturen vermittelt werden, da es sich um einen Informatik-Anfangsunterricht handelt.

Dagegen wird der Begriff *OOP-Later* gesetzt, wenn ein Teil der prozeduralen Themen (im Kontext zum imperativen Paradigma) vor dem Einstieg in die OO-Themen unterrichtet wird und auch hier wieder der Schwerpunkt eher in der Programmierung bzw. Implementierung als in der Modellierung liegt [ES09c, S. 125]. Es könnte sein, dass das OOP-Later-Vorgehen deckungsgleich ist mit dem Begriff des „traditionellen Unterrichtens“ (als Abgrenzung zu objects-first). Der traditionelle Programmierkurs steigt in der Regel (nach dem Hello-World-Programm) mit Variablen, Datentypen und Zuweisungsoperatoren in die Programmierung ein und arbeitet sich über kleine Programme mit Steuerstrukturen hin zu kleinen Programmen mit Prozeduren [ES09a, S. 16].



Für Heide und Helmut Balzert [BB04] ist dies eine von drei Möglichkeiten das Programmieren zu lehren:

„Es werden zunächst die leicht verständlichen Konzepte Variable, Zuweisung Ausdruck, einfache Typen und die Ablaufsteuerung (Sequenz, Verzweigung, Schleife) vermittelt. [...] Intellektuell anspruchsvoller ist die Vermittlung des Methoden- bzw. Prozedur- und Funktionskonzepts mit den verschiedenen Möglichkeiten der Parameterübergabe. Auch die Erarbeitung dieser Konzepte kann zunächst unabhängig von der Objektorientierung erfolgen.[...] Werden diese Grundkonzepte [...] beherrscht, sollte der Umstieg auf die Objektorientierung erfolgen[...].“ [BB04, S. 24].

OOP-Later besagt aber auch durch den Begriff *Later*, dass letztlich auch alle Themen der objektorientierten Programmierung unterrichtet werden, dass es also nach einer gewissen Zeitspanne zu einem Umstieg von prozeduralen (bzw. imperativen) auf objektorientierte Themen kommt.

Kim Bruce [Bru05] verwendet in einem Artikel zu der Diskussion, die von Mitgliedern der Mailingliste (sigcse-members@acm.org) der SIGCSE [SIG10] zum Thema „How to teach Computer Science 1“ geführt wurde, den Begriff „objects-late approach“ (als Gegenentwurf zu einem „objects-early approach“):

„The more “traditional” approach is to teach the first two-thirds of the course in the same way one might teach a procedural language (or as some faculty teach C++ as a “better” C), only introducing objectoriented concepts in the last few weeks of the course. Let’s call this the objects-late approach.“ [Bru05, S. 112].

Der Wahl der Begrifflichkeit *First* bzw. *Later* gibt dabei einen klaren Hinweis auf eine zentrale These dieser Arbeit: Die beiden Vorgehensweisen unterscheiden sich (fast) nur in der zeitlichen Reihenfolge der Themen. Die Themen selber sind die gleichen bzw. gleichartig.

Aus den in diesem Kapitel gegebenen Definitionen können jetzt beide Ansätze miteinander verglichen werden:

„Mit Hilfe dieser Definition kann nun (unter Fokussierung auf OOP-First und OOP-Later und der dort enthaltenen Programmier-Komponente) der objects-first-Ansatz mit dem objects-later-Ansatz - unter dem als zentral herausgearbeiteten Aspekt der unterschiedlichen Sequenzierung von Lerninhalten - verglichen werden.“ [ES09c, S. 125].

Dies ist auch Teil des Titels dieser Dissertationsschrift: „Unterschiede im Lernerfolg und Unterschiede im subjektiven Erleben des Unterrichts von Schülerinnen und Schülern im Informatik-Anfangsunterricht (11. Klasse Berufliches Gymnasium) in Abhängigkeit von der zeitlichen Reihenfolge der Themen (OOP-First und OOP-Later)“.

Ein kleiner Randhinweis: Wenn man in der Suchmaschine Google (Stand: 14.01.2010) den Suchbegriff „OOP-first“ bzw. „OOP first“ bzw. „OOP-First“ bzw. „OOP First“ eingibt, dann sind die ersten beiden Ergebnisse der ersten zehn Ergebnisse einer meiner Veröffentlichungen zugeordnet (siehe Anhang U).

Beim Suchbegriff „OOP-later“ bzw. „OOP later“ bzw. „OOP-Later“ bzw. „OOP Later“ sind es aber schon fünf der ersten zehn Ergebnisse. Für mich ein erstes Indiz dafür, dass der Begriff „OOP-Later“, gerade auch im Kontext zu dem Begriff „OOP-First“, von mir verstärkt in die fachdidaktische Diskussion eingebracht wird.

#### 1.2.4 Argumente für beide fachdidaktischen Vorgehensweisen

Aus der Sicht der Studie, die im Sinne eines (Entscheidungs-) Experiments aufgebaut ist (vgl. Kapitel 3.2), soll nicht das eine oder andere fachdidaktische Vorgehen präjudiziert werden. Für beide Vorgehensweisen lassen sich nämlich gute Argumente finden, die im Folgenden mit Hilfe von Quellen bzw. Zitaten angeführt werden sollen.

Für ein OOP-First-Vorgehen wird wie folgt argumentiert.

Crutzen und Hein [CH95] sehen das objektorientierte Denken als didaktische Basis der Informatik und plädieren dafür, objektorientiertes Denken so früh wie möglich in den Informatikunterricht zu integrieren.

Siegfried Spolwig [Spo95] meint, dass OOP-First kein unnötiges Umdenken (durch den Paradigmenwechsel) erfordert. Er schreibt:

„Warum soll man nicht von Anfang an bessere Konzepte der Software-Entwicklung nutzen, statt den Schülern einen Programmierstil beizubringen, den man ihnen später bei größeren Projekten wieder austreiben muss!“  
[Spo95, S. 45].

Diese Meinung teilt auch Joseph Bergin [Ber00], der schreibt, dass es schwierig sei, auf objektorientiertes Denken zu wechseln, wenn vorher prozedurales Denken vermittelt wurde.

Andreas Schwill [Sch95] sieht die OOP im Kontext zur natürlichen Denkweise der Schülerinnen und Schüler. Er schreibt in seinen Schlussbemerkungen zu „Programmierstile im Anfangsunterricht“:

„Wir haben versucht zu begründen, daß von den bisher erprobten Programmierstilen die objektorientierte Programmierung der geeignetste Ansatz ist, um Informatik im Anfangsunterricht zu vermitteln, weil sie fundamentale kognitive Prozesse widerspiegelt und damit der natürlichen Denkweise am nächsten kommt.“ [Sch95, S. 186].

Heide und Helmut Balzert [BB04] argumentieren auch mit dem Denken der Schülerinnen und Schüler, welches sich ihrer Meinung nach gleich an einer Objekt-Klassen-Kategorie orientieren sollte:

„Beginnt der Unterricht mit der Einführung von Objekten und Klassen, dann orientiert sich das Denken des Lernenden zuerst an diesen Kategorien. Dies ist ein großer Vorteil.“ [BB04, S. 23].

Carsten Schulte und ich [ES09c] greifen einen Gedanken von Elmar Souvignier auf, der die Motivation bei OOP-First als Argument anführt:

„Instruktionspsychologisch lässt sich [...] im Sinne einer Top-Down-Argumentation positiv anführen, dass den Lernenden die Ziele und

Anwendungsmöglichkeiten schnell deutlich werden, auf die sie hin arbeiten.  
Dies könnte sich motivational günstig auswirken.“ [ES09c, S. 125].

Für ein OOP-Later-Vorgehen wird wie folgt argumentiert.

Pallmer und Wachler [PW05] führen das (geringere) Abstraktionsniveau und den geschichtlichen Ablauf als Argumente für ein OOP-Later-Vorgehen an. Sie schreiben in ihrem C#-Lehrbuch:

„Obwohl C#.NET eine vollständig objektorientierte Programmiersprache ist, ist es nach Auffassung des Autors nicht unbedingt notwendig, das Erlernen dieser Sprache mit der Behandlung von Klassen und Objekten zu beginnen. Das dafür nötige Abstraktionsniveau kann von Anfängern kaum erwartet werden. Deshalb folgt diese Schulungsunterlage didaktisch dem Weg, den auch die Programmiersprachen in ihrer geschichtlichen Entwicklung genommen haben.“ [PW05, S. 7].

Das niedrigere Abstraktionsniveau am Anfang des Programmierunterrichts wird auch von Helmut und Heide Balzert [BB04] erwähnt:

„Das Vorgehen [...] ermöglicht es den Schülern schrittweise ein mentales Modell vom Programmieren aufzubauen, einschließlich der Handhabung von Compiler und Entwicklungsumgebung. Das Abstraktionsvermögen wird dabei nicht so stark beansprucht [...].“ [BB04, S. 24].

Balzerts, die weiter oben noch mit einem Argument für ein OOP-First-Vorgehen zitiert wurden, schreiben in der Einführung eines Artikels zum Thema „Modellieren oder Programmieren oder beides?“, dass OOP-First die Schülerinnen und Schüler überfordert:

„Beim Erlernen einer Programmiersprache (wie z. B. JAVA) kann dagegen, um die Lernenden nicht zu überfordern, nicht sogleich mit der Einführung des Objekt- und Klassenbegriffs begonnen werden; es sollen vielmehr zunächst – im herkömmlichen prozeduralen Stil – die Grundkonzepte von Algorithmen und elementaren Datenstrukturen erarbeitet werden.“ [BB04, S. 20].

Sie sprechen sich auch für einen Texteditor-Einstieg anstelle eines BlueJ-Einstiegs aus:

„Wir sind der Meinung, dass der Lernende zuerst mit dem JDK (Java Development Kit) und einem Texteditor beginnen sollte, um ein Gefühl für den minimalen Werkzeugeinsatz zu bekommen. [...] Als vorbildlich für den nächsten Schritt nach dem JDK halten wir BlueJ [...].“ [BB04, S. 24].

Alfred Hermes [Her96] hält 1996 ein Plädoyer für einen gleitenden Paradigmenwechsel und verspricht durch eine spezielle Unterrichtssequenz einen behutsamen Einstieg in die Welt der OOP mit einem thematisch integrierten Übergang von strukturierter zur objektorientierten Programmierung.

Schubert und Schwill [SS04] wollen auch nicht auf den „imperativen Baukasten“ verzichten:

*„Auf den imperativen Baukasten kann man bis auf weiteres nicht verzichten. Jedoch sollte man dann stets die objektorientierten Erweiterungen imperativer Sprachen wählen.“* [SS04, S. 173, kursive Schreibweise im Original].

Carsten Schulte und ich [ES09c] greifen einen Gedanken von Elmar Souvignier auf, der bei OOP-Later einen systematischeren Wissensaufbau sieht:

„Instruktionspsychologisch könnte man mit dem Vorteil eines systematischen Wissensaufbaus argumentieren.“ [ES09c, S. 125].

Es wird durch die Zitate deutlich, dass beide Vorgehensweisen sowohl Vorteile als auch Nachteile haben. Dies weiter herauszuarbeiten und zu belegen (mit den Schwerpunkten Lernerfolg und subjektives Erleben) ist ein Ziel dieser Studie (vgl. auch Kapitel 3.1: „Ziele der empirischen Studie“).

### **1.3 Zusammenfassung**

Seitdem das objektorientierte Paradigma in den Informatik-Anfangsunterricht Einzug gehalten hat, gibt es Beiträge bzw. Diskussionen (siehe auch nächstes Kapitel) um den „besten Weg“, also um die „richtige“ didaktische Vorgehensweise im Hinblick auf die Methoden, die eingesetzten Werkzeuge, die Themen, die Übungsaufgaben, die Programmiersprache etc.. Trotz der „objects-first-Schlachtrufe“ (z. B. von Diethelm [Die07] und Barnes und Kölling [BK06]) gibt es aber immer wieder kritische Stimmen, die auf die praktischen Schwierigkeiten bei der Umsetzung im Unterricht bzw. in den Kursen hinweisen (z. B. von Heide und Helmut Balzert [BB04]). Es gibt also weiterhin fachdidaktische Probleme, die teilweise noch ungelöst sind und für die Relevanz dieser Studie sprechen (siehe Kapitel 1.1).

Zusätzlich werden die Begriffe objects-first bzw. objects-later nicht selten unterschiedlich interpretiert. Deshalb wurde im letzten Kapitel versucht, die Unterschiede von objects-first und objects-later herauszuarbeiten (siehe Kapitel 1.2.1) und es wurde dabei festgestellt, dass bei beiden Ansätzen die Themen gleichartig sind (siehe Kapitel 1.2.2). Die gleichartigen Themen wurden im Kontext der Programmier-Lehrbücher herausgearbeitet bzw. benannt. Darauf aufbauend wurden die beiden Begrifflichkeiten OOP-First und OOP-Later eingeführt bzw. definiert (siehe Kapitel 1.2.3): Der Hauptunterschied beider fachdidaktischen Vorgehensweisen ist die sequenzielle Abfolge der Themen. Der Schwerpunkt beider Vorgehensweisen liegt eher in der Programmierung als in der Modellierung, ohne dass auf die Modellierung verzichtet wird. Am Ende wurden für beide fachdidaktischen Vorgehensweisen schlaglichtartig Argumente aufgeführt, die jeweils für das eine oder das andere Vorgehen sprechen (siehe Kapitel 1.2.4).

Ziel dieser Studie soll es sein, sich den beiden Vorgehensweisen „objektiv“ zu nähern und die Daten bzw. Ergebnisse des Experiments für oder gegen einzelne Bausteine des OOP-First- bzw. OOP-Later-Vorgehens entscheiden zu lassen.

Im Folgenden sollen aufgrund der nationalen und internationalen fachdidaktischen Diskussion die Forschungsfragen und Forschungshypothesen der Studie herausgearbeitet werden.

## 2 Stand der fachdidaktischen (objects-first) Diskussion, Forschungsfrage und Forschungshypothesen

Die objects-first-Diskussion ist weiterhin aktuell (vgl. z. B. [LBua06], Kapitel 1.1 und die folgenden Kapitel 2.1 und 2.2). Ein Hauptgrund hierfür (und für die Komplexität dieser Diskussion) ist, dass so viele verschiedene didaktische Dimensionen angesprochen werden und deswegen empirische Studien in Gefahr geraten, „Äpfel mit Birnen zu vergleichen“ [ES09a, S. 15].

In den nächsten beiden Unterkapiteln soll mit Hilfe deutscher Beiträge die objects-first-Diskussion dokumentiert und mit Hilfe internationaler Beiträge die bisherigen empirischen Ergebnisse zu dieser Diskussion vorgestellt werden. Im dritten Unterkapitel werden einige deutsche und internationale Beiträge zum subjektiven Erleben des (Informatik-) Unterrichts angeführt.

### 2.1 Deutsche Beiträge, speziell zur fachdidaktischen (objects-first) Diskussion

Im Folgenden sollen einige deutsche Beiträge zur objects-first-Debatte bzw. zum objektorientierten Paradigma und zur fachdidaktischen Diskussion (über Tools, Methoden etc.) chronologisch kurz genannt (und teilweise in ihren Widersprüchen kommentiert) werden.

Hermes [Her96] hält 1996 ein Plädoyer für einen gleitenden Paradigmenwechsel und verspricht durch eine spezielle Unterrichtssequenz einen behutsamen Einstieg in die Welt der OOP mit einem thematisch integrierten Übergang von strukturierter zur objektorientierten Programmierung.

Doberkat [Dob00] möchte 2000 den Beweis antreten, dass der objektorientierte Ansatz nicht nur in der Modellierung, sondern auch in der Implementierung Vorteile bringt. Seine Modellierung von „Hofzwerge“ (siehe Abbildung 6) ist aber nicht immer nachvollziehbar, da sich einige Unterklassen nur im Klassennamen unterscheiden, nicht aber in den Attributen oder Methoden.

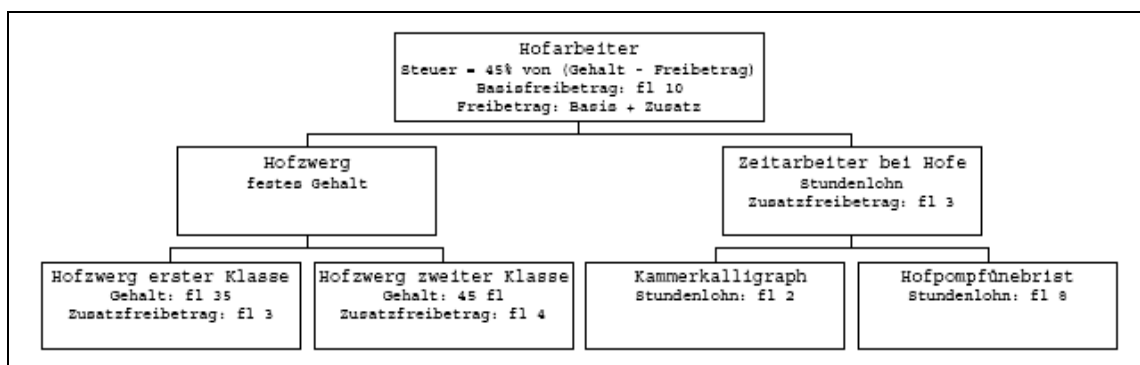


Abbildung 6: „Klassendiagramm“ für eine Hofzwerge-Modellierung [Dob00, S. 80, Abbildung 3]

Brinda [Bri00] erläutert 2000 im Zusammenhang mit dem objektorientierten Modellieren das Sammeln und Strukturieren von Übungsaufgaben im Informatikunterricht. Der Stellenwert der OOM gegenüber der OOP müsse deutlich werden. Sein „Schulmodell“ als Musterlösung einer OOM-Aufgabe erscheint relativ komplex und

dass im selben Klassendiagramm sowohl die Klasse „Elternbeirat“ als auch die Klasse „Lineal“ zu finden ist, ist m. E. diskussionswürdig.

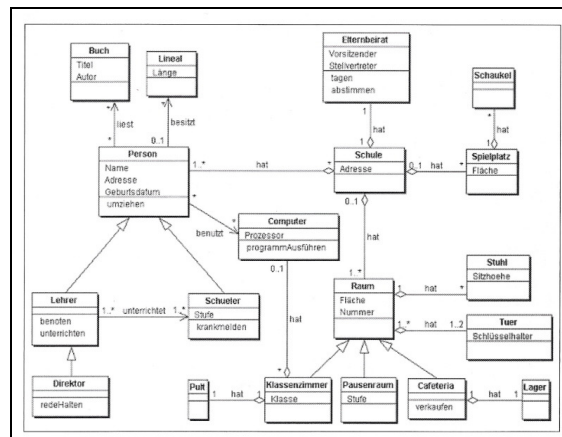


Abbildung 7: „Schulmodell“ als Musterlösung einer OOM-Aufgabe [Bri00, S. 45, Bild 9]

Baumann [Bau01] sieht dies 2001 ähnlich, indem er beklagt, dass Brinda die Programmierung ausklammert:

„Denn erst das konkrete Programm kann Lehrer und Schüler von der eingeführten [...] Modellierung überzeugen. Ein „Schulmodell“ beispielsweise, das im Klassendiagramm die Schaukel auf dem Spielplatz, das Pult im Klassenzimmer und das Lineal im Schulranzen aufführt [...], ist schlicht wirklichkeitsfremd. Denn es gibt keinen vernünftigen Programmzweck, der die Berücksichtigung solcher Einzelheiten rechtfertigen könnte.“ [Bau01, S. 10].

Baumann schildert dann ein Beispiel zum objektorientierten Entwurf in Java am Beispiel der Beziehungen zwischen Kunden, Bestellungen, Rechnungen und Waren und findet dabei m. E. eine gute Mischung aus Modellierung einerseits und konkreter Implementierung andererseits.

Engelmann [Eng03] hält 2003 ein Plädoyer für eine objektorientierte Betrachtungsweise in Informatik-Systemen und erläutert die so genannte Objekt-Attribut-Methode.

Auch Schulte [Sch03b] baut 2003 auf einen objektorientierten Zugang zur Informatik. Er entwickelt hierfür ein theoriegeleitetes Unterrichtskonzept (das life3-Unterrichtskonzept) für den Anfangsunterricht.

Wehrheim [Weh03] muss sich m. E. den Vorwurf gefallen lassen, dass in seinem Beitrag aus dem Jahr 2003 über „Objektorientiertes Modellieren mit DELPHI“ viel DELPHI (also Quelltexte) und keinerlei Modellierung (sondern ausschließlich Benutzungsoberflächen) zu sehen ist.

Auch Füller [Fül04] merkt 2004 zu Wehrheim an, dass das gewählte Beispiel (Memory-Spiel) keine objektorientierte Modellierung darstellt bzw. dass überhaupt keine Modellierung stattfindet.

Steinbrucker [Ste04] mahnt 2004 mit Hilfe von drei Ausrufungszeichen zur „Objektorientierung im Anfangsunterricht!!!“ Er beschreibt in diesem Zusammenhang eine Unterrichtsreihe zum Thema „Simulation einer Taschenlampe“, wobei sich für mich die Frage stellt, warum eine Taschenlampe objektorientiert simuliert bzw. modelliert werden muss (siehe Abbildung 8).

Heubaum [Heu04] schreibt 2004:

„Welchen Nutzen bringt nun der Übergang zum objektorientierten Programmieren? Zunächst fallen gewaltige Nachteile [...] auf. Die Programme sind [...] erheblich aufgebläht und [...] wesentlich komplizierter geworden. [...] Die Schwierigkeiten [...] werden auch nicht dadurch behoben, dass man Werkzeuge wie BlueJ [...] einsetzt.“ [Heu04, S. 76].

Heide und Helmut Balzert [BB04] meinen 2004:

„Beim Erlernen einer Programmiersprache (wie z. B. JAVA) kann dagegen, um die Lernenden nicht zu überfordern, nicht sogleich mit der Einführung des Objekt- und Klassenbegriffs begonnen werden; es sollen vielmehr zunächst – im herkömmlichen prozeduralen Stil – die Grundkonzepte von Algorithmen und elementaren Datenstrukturen erarbeitet werden.“ [BB04, S. 20].

Die Zeitschrift LOG IN [LI04] widmet sich im Jahr 2004 (im Heft 128/129) ganz dem Themenschwerpunkt „Objektorientiertes Modellieren und Programmieren“.

Auch die Diskussion, was objektorientierte Modellierung ist, bzw. was „richtige“ OOM ist, hält an. Spolwig [Spo04] kritisiert 2004 in diesem Kontext das Tool „Stifte und Mäuse“.

Lavergne [Lav04] spricht 2004 einerseits von einem gewachsenen Einfluss der OOP in der Software-Erstellung, andererseits meint er, dass sich zunehmend auch die Grenzen dieses Paradigmas gezeigt haben. Er erläutert in diesem Zusammenhang den Begriff der komponentenorientierten Programmierung.

Fischer und Knapp [FK05] widmen 2005 einen Artikel dem „richtigen“ Modellieren und schlagen bestimmte Formen als Festlegung zur Beschreibung von Klassen und Objekten vor, z. B. eine vereinfachte UML-Notation.

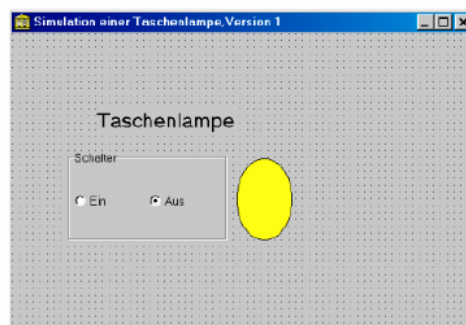


Abbildung 8: Objektorientierung im Anfangsunterricht (Modellierung einer Taschenlampe) [Ste04, S. 57]

Lavergne [Lav05] mahnt 2005 an, dass der objektorientierte Ansatz nur dann sinnvoll ist, wenn die beiden wesentlichen mit ihm verbundenen Eigenschaften – Vererbung und Polymorphie – tatsächlich benötigt werden. Er stellt die Frage: „Wann ist eine Klasse eine Klasse?“ und erläutert dann typische Fehler der OOP, wie erzwungene Objektorientierung und falsche Klassenbildung. Steinbrückers Analyse einer Taschenlampe wird als nicht seriös bezeichnet. Lavergne modelliert dann aber selber eine Ampel.

Humbert [Hum06] stellt 2006 das Problem bzw. das Problemlösen in den Mittelpunkt und dokumentiert Schemata (imperativ, objektorientiert, wissensbasiert und funktional) zum Problemlösen für den Einsatz im Unterricht.

Baumann [Bau07] bietet 2007 unter dem Thema „Objektorientiertes Modellieren“ Ausgaben, Eingaben, Algorithmen und Java-Quelltext an, nirgendwo sieht man aber ein Modell (und sei es nur ein Klassendiagramm).

Diethelm [Die07] stellt 2007 ein Unterrichtskonzept bzw. eine Unterrichtsmethodik für die objektorientierte Modellierung im Informatikunterricht vor und verkündet: „Strictly models and objects first“.

Hartmann, Näf und Reichert [HNR07] gehen in ihrem Buch „Informatikunterricht planen und durchführen“ nicht explizit auf Programmierparadigmen ein. Für sie gibt es nicht den einzigen, richtigen Zugang im Informatikunterricht:

„Längere Unterrichtssequenzen im Informatikunterricht müssen nicht zwingend einer Fachsystematik folgen. Entscheidend ist, dass der Unterricht eine klar erkennbare Strukturierung aufweist. Bottom-up, top-down, entlang der historischen Entwicklung, vom Naheliegenden zum Allgemeinen oder umgekehrt und andere Zugänge tragen zur Vielfalt des Unterrichts bei.“ [HNR07, S. 49].

Thile [Thi08] wagt sich 2008 an das Thema „Objektorientierte Programmierung von Schachproblemen“, allerdings nicht mit den Figuren als Objekten, sondern mit den Problemen als Objekten. Die abgeleitete Klasse Damenproblem erbt von der Klasse Turmproblem, die wiederum eine Unterklasse von Schachproblem ist.

Zendler und Spannagel [ZS08] schreiben 2008, dass sie durch eine Umfrage an deutschen Universitäten (Informatikinstitute) herausgefunden haben, dass die Algorithmik als deutlich wichtiger erachtet wird als z. B. die OOM.

Engbring [Eng09] schreibt 2009 am Anfang seines Artikels „Wozu objektorientiertes Programmieren?“:

„Man kann es drehen und wenden, wie man will: Kein Weg führt daran vorbei, dass im Informatikunterricht der gymnasialen Oberstufe programmiert wird. Aber wie soll programmiert werden?“ [Eng09, S. 25].

Er will aufzeigen, dass eine Entscheidung für die Objektorientierung nicht bedeutet, dass zentrale Inhalte der Algorithmik verloren gehen und ordnet die Objektorientierung in eine geschichtliche Ordnung ein.

## **2.2 Internationale Beiträge, speziell zu bisherigen empirischen Ergebnissen der objects-first-Debatte (auch im Hinblick auf den Lernerfolg)**

Wenn man die internationalen Beiträge zur objects-first-Debatte betrachtet, so ist m. E. erkennbar, dass es zum einen nach einer Anfangseuphorie Anfang der 90er Jahre (Umstieg vom prozeduralen Paradigma auf das objektorientierte Paradigma) zu einer Ernüchterung Anfang des neuen Jahrtausends gekommen ist und dass es zum anderen vergleichsweise wenig empirische Untersuchungen zu diesem Thema gibt.



Rick Decker und Stuart Hirshfield [DH94] schreiben 1994 einen Beitrag mit dem ironisch gemeinten Titel „The Top 10 Reasons Why Object-Oriented Programming Can't Be Taught in CS 1“. In diesem nennen sie die Hauptgründe, die gegen ein objects-first-Vorgehen vorgebracht werden und versuchen dann, diese Gründe zu widerlegen:

„Over the past year we have changed our CS 1 course from a standard Pascal-based, procedural programming course into one that emphasizes C++ and object-oriented programming (OOP). While our experience to date indicates that this was a good decision for both our students and our department, the decision did not come easy. We struggled long and hard with many, if not most, of the questions and issues that have come to be associated with teaching OOP to undergraduates. This paper recounts our struggles, and presents our responses to the more serious of the pedagogical questions that we considered. In hindsight, many of the reasons we came up with for not using OOP in CS 1 are seen to reflect our lack of understanding of the paradigm, our fear of the language, and our past experience teaching Pascal and the procedural paradigm.“ [DH94, S. 51].

Gut zehn Jahre später veröffentlicht Kim B. Bruce [Bru05] in einem Artikel eine Diskussion, die von Mitgliedern der Mailingliste (sigcse-members@acm.org) der SIGCSE [SIG10] zum Thema „How to teach Computer Science 1“ geführt wurde. Er schreibt u. a.:

„Looking back at introductory textbooks using Java, there seems to have been a gradual change in approach over time. The first wave of Java textbooks primarily followed the structure of C++ texts, presenting procedural programming early, while postponing the detailed discussion of object-oriented concepts until after arrays, for example. There has been an increasing trend in more recent texts, however, to introduce classes and objects very early, postponing or even omitting the introduction of some procedural control structures. Papers at SIGCSE and ITiCSE over the last several years have also reflected the same trend. However, there seems to be increasing concern among some faculty about whether this is the right direction, or even whether it is possible to teach this way successfully.“ [Bru05, S. 111].

Es würde seiner Meinung nach helfen, wenn es mehr empirische Untersuchungen gäbe, die den Erfolg von innovativen Lehr-Ansätzen im Informatik-Anfangsunterricht (Computer Science 1) kontrollieren [Bru05, S. 115].

Dies ist leichter gesagt als getan, wenn man sich die Komplexität der diskutierten Themen veranschaulicht [ES09a, S. 15]. Denn obwohl man z. B. „objects-first“ als eine konkrete fachdidaktische Idee ansehen könnte, stecken in dem Thema eine Vielzahl von didaktisch-methodischen Problemstellungen. Die folgende Auflistung von veröffentlichten Themen soll einen Eindruck vermitteln, wie viele Themen bzw. Fragestellungen der objects-first-Debatte innewohnen [ES09a, S. 15]:

- Mental Models and Programming Aptitude [CBL07]

- From Objects-First to Design-First with Multimedia and Intelligent Tutoring [MWua05]
- Day One of the Objects-First First Course: What To Do [BCua07]
- Implications of Perspective in Teaching Objects First and Object Design [Chr05]
- An Objective Comparison of Languages for Teaching Introductory Programming [MR06]
- Methodology First and Language Second: A Way to Teach Object-Oriented Programming [ZZ03]
- Relationship of Early Programming Language to Novice Generated Design [CMS06]
- A Study of the Development of Students' Visualizations of Program State during an Elementary Object-Oriented Programming Course [SKT08]
- Adopting XP Practices for Teaching Object Oriented Programming [KSD06]

Tew, McCracken und Guzdial [TMG05] untersuchen den Einfluss von Informatik-Einführungskursen mit verschiedenen fachdidaktischen Vorgehensweisen auf den darauf folgenden Kurs:

„Computer science has long debated what to teach in the introductory course of the discipline, and leaders in our field have argued that the introductory course approach is critical to student development. We investigated the impact of alternative approaches to introductory computing by considering the questions of what students bring to their second class in computing and how the outcomes differ depending on the students' alternative first course.“  
[TMG05, S. 25].

Sie kommen dabei zu dem Ergebnis, dass Unterschiede im Lernerfolg der Studenten auf lange Sicht hin (also im darauf folgenden Kurs) verschwinden:

„Our study showed significant differences in understanding of introductory concepts, such as iteration, conditionals, and arrays, at the beginning of the second course. However, by the end of the second course their understanding had converged.“ [TMG05, S. 25].

Sie schlussfolgern:

„Our results indicate that the traditional debates over the pedagogical approach (e.g. objects-first vs. objects-early) for the introductory CS course may not be as important as the community believes.“ [TMG05, S. 32].

Auch eine Studie von Vilner, Zur und Gal-Ezer [VZG07] an der „Open University of Israel“ findet keine signifikanten Unterschiede zwischen prozeduralen und objektorientierten Vorgehensweisen. Die Studenten der Universität waren in zwei verschiedenen CS1-Kursen, in einem prozeduralen C++-Kurs und in einem objektorientierten Java-Kurs:

„In our previous introductory course, the language introduced was C++, primarily the procedural facet of the language... In 2005, we designed a new introductory course, shifting from C++ to Java, and introducing the whole issue of OO Programming... The course covers all the topics covered in the

procedural course, plus the main topics of OO programming as encapsulation and information hiding, classes and objects, inheritance, polymorphism... During the first two weeks of the semester, we teach Karel J. Robot [BSua05], which is an environment designed to provide a means of introducing novice programmers to object-oriented programming. In the following weeks, we teach Java, using the BlueJ environment [BlueJ Website: <http://www.bluej.org>]. For the first three semesters when we offered the new course, we continued to offer the C++ course as well.“ [VZG07, S. 172p, Referenzen angepasst].

Am Ende der Kurse wurde ein dreistündiges Examen mit Papier und Stift geschrieben, dabei waren drei der vier Fragen gleich gestellt: Bei der ersten Frage sollten die Studenten eine rekursive Methode schreiben, bei der zweiten Frage einen effizienten Algorithmus entwickeln und bei der dritten Frage sollten die Studenten eine Klasse bzw. einen abstrakten Datentypen modellieren. Es gab bei der Auswertung keine signifikanten Unterschiede:

„Our study refers only to the first three questions, which were the same in both exams. We normalized the sum of the three questions to the scale 0-100. Among the students of the procedural course, 42% passed the exam (the normalized grade). The average of these students' grades was 74.6, with a standard deviation of 10.2. Among the students of the OO course, 45% passed the exam (the normalized grade). The average of these students' grades was 78.1, with a standard deviation of 9.3. However, there was no significant difference between the two groups. ( $p = 0.101$ )“ [VZG07, S. 174].

David Gries [Gri08] behauptet, dass „OO-first“ der einzig richtige Weg ist, wenn gewisse Prinzipien eingehalten werden:

„We believe that the principles discussed in this paper lead to the conclusion that OO first is the only way to go. But don't go this way without paying sufficient attention to: (1) the programming-skill aspect, (2) a model of objects and classes to which students can relate, (3) notation and terminology, (4) the order in which topics are introduced, and (5) an IDE that does not require an application.“ [Gri08, S. 35].

Im Gegensatz dazu berichtet Stuart Reges [Reg06], dass sehr viele Probleme der „Objects-first-CS1-Kurse“ dadurch gelöst wurden, dass die Informatik-Anfangskurse komplett wieder auf die prozedurale Programmierung umgestellt wurden:

„Our new version of CS1 looks a lot like a 1980.s course taught in Pascal. We have gone back to procedural style programming. I was motivated to do this after attempting and failing to teach a broad range of introductory students at the University of Arizona using an “objects early” approach. I found that my best students did just fine in the new approach, but the broad range of midlevel students struggled with the object concept.“ [Reg06, S. 294].

Chenglie Hu [Hu05] spricht schon im Jahr 2005 von der „Schädlichkeit“ eines objects-first-Vorgehens, er schreibt:

„An analysis is [...] presented showing the harmfulness of teaching objects-first.“ [Hu05, S. 209].

Adrienne Decker [Dec03] bezieht sich auf zwei CS1-Kurse, die (ähnlich wie in dieser empirischen Studie) fachdidaktisch unterschiedlich gestaltet wurden und untersucht, wie erfolgreich die jeweiligen Studenten bei dem darauf folgenden objektorientierten CS2-Kurs sind:

„There are many opinions as to the best way to present objects and the concepts of object-oriented programming within the CS1 and CS2 curriculum. This paper seeks to analyze the benefits for student performance using two different groups of students. Both groups were introductory programming students using Java and both groups were exposed to the idea of objects over the duration of the course. However, one group was exposed to an objects emphasized/objects thematic approach, while the other group was presented with a procedurally thematic approach. An experiment was conducted to see which group of students performed better in an object-oriented CS2.“ [Dec03, S. 238].

Sie kommt zu folgendem Ergebnis im objektorientierten CS2-Kurs:

„The results of the midterm exam are extremely encouraging, in that they support the idea that objects should be introduced early and emphasized after their introduction.“ [Dec03, S. 245].

Außerdem hat sie auch festgestellt, dass die Betonung der Objektorientierung keine Nachteile in den prozeduralen Wissensbeständen ergeben hat.

Pears, Seidman, Malmi, Mannila, Adams, Bennedsen, Devlin und Paterson [PSua07] geben einen Überblick über die Literatur und die vorhandenen Studien über „teaching of introductory programming“:

„The goal of this paper has been to give an overview of research issues relating to the teaching of introductory programming, with specific reference to curriculum, pedagogy, and languages and tools for supporting learning. It has provided material from the body of discipline-specific research in each of these areas. [...]. For that reason, we have not attempted to give a canonical answer to the question of how to teach introductory programming. Rather, the paper seeks to provide support for a wide variety of approaches, and to help teachers to provide scholarly motivations for course design choices with respect to learning situation, instructional approach, selection of a language, and use of support tools and technologies. [...] After decades of exploratory research on teaching programming, it has become clear that studying individual learning settings is no longer sufficient. Our literature review leads us to believe that larger-scale systematic studies are vital to a better

long-term understanding of how to teach this key element of the field. [...]. In addition, they should provide strategically significant new empirical results, allowing the community to use research to begin to resolve key questions regarding what to teach beginning programmers and how to teach programming.“ [PSua07, S. 211].

Zusammenfassend kann also festgestellt werden, dass weiterhin keine generalisierenden Aussagen zu objects-first vs. objects-later möglich sind, da es so viele fachdidaktische Unterthemen bzw. Fragestellungen in dieser Debatte gibt.

## **2.3 Beiträge zum subjektiven Erleben von (Informatik-) Unterricht**

Zusätzlich zu dem (objektiven) Aspekt des Lernerfolgs wird in dieser Arbeit ein weiterer Aspekt in die objects-first-Diskussion eingebracht (siehe auch Kapitel 2.4.3 und Titel der Dissertationsschrift), nämlich der Aspekt des subjektiven Erlebens von Informatik-Anfangsunterricht durch die Schülerinnen und Schüler. Dazu gehören z. B. die emotionale Befindlichkeit, das Wohlfühlen und die Stimmungslage. Es wird untersucht, ob das subjektive Erleben bei beiden fachdidaktischen Vorgehensweisen gleich ist. Der Einfluss des subjektiven Erlebens (z. B. Wohlfühlfaktor (comfort level)) der Schüler bzw. Studenten im Informatikunterricht wird nämlich in den Studien relativ selten fokussiert, erst recht nicht in Abhängigkeit von der fachdidaktischen Vorgehensweise.

Sembill [Sem92] prägt 1992 den Begriff der „Emotionalen Befindlichkeit“ von Schülerinnen und Schülern im Unterricht. Er schreibt:

„Emotionale Befindlichkeit kann als emotional-motivational geprägtes, subjektives und situationsspezifisches Erleben eines Zustandes verstanden werden. Dieses Erleben bezieht sich auf Kognitionen i.e.S. Emotionale Befindlichkeit kann sowohl als Auslöser, Begleiterscheinung und / oder Folge dieser kognitiven Prozesse i.e.S. auftreten.

In einer ganzheitlichen Betrachtungsweise des psychophysischen Person-Umwelt-Bezuges wird für die beteiligten Konstrukte Emotion, Motivation und Kognition ein Wirkungsgefüge angenommen [...].“ [Sem92, S. 118].

Er begründet, dass die emotionale Befindlichkeit von Schülerinnen und Schülern Einfluss auf ihr Denken, Handeln und Lernen im Unterricht hat und schreibt zusammenfassend zum Thema „Problemlösefähigkeit, Handlungskompetenz und Emotionale Befindlichkeit“:

*„Das unterstreicht die Notwendigkeit, Emotionale Befindlichkeit nicht allein unter einer psychologischen Perspektive zu betrachten, sondern konstruktiv in didaktische oder fachdidaktische Bemühungen einzubeziehen [...].“* [Sem92, S. 284, kursive Schrift im Original].

Sembill und Seifried (vgl. z. B. [Sem05] und [Sei04]) unterteilen das Erleben von Unterricht in die folgenden drei Aspekte: emotionaler Aspekt, motivationaler Aspekt und kognitiver Aspekt.

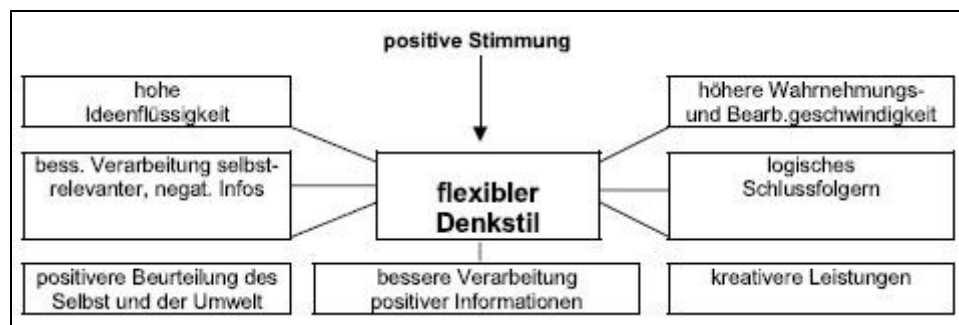


Abbildung 9: Wirkungen von positiven Stimmungen auf kognitive Prozesse [EH08, S. 61, Abb. 1]

Heidrun Edlinger und Tina Hascher [EH08] schreiben im Jahr 2008 in ihrer Zusammenfassung zum Thema „Überlegungen zur Wirkung von Emotionen auf schulisches Lernen und Leisten“:

„Aus den bisher referierten Ergebnissen lassen sich durchaus förderliche Effekte von positiven Emotionen ableiten (vgl. Abbildung 9): die bessere Verarbeitung von positiven Informationen sowie die positivere Beurteilung von sich selbst und der Umwelt. [...] So erhöht positive Stimmung die Wahrnehmungs- und Bearbeitungsgeschwindigkeit und sogar die Leistung beim logischen Schlussfolgern.“ [EH08, S. 60, Referenz angepasst].

Sie schreiben weiter:

„Die Annäherung an das Thema Emotionen und Lernen in der Schule über die experimentelle Stimmungsforschung erbringt aus unserer Sicht fruchtbare Erkenntnisse. Trotzdem steht außer Frage, dass empirische Untersuchungen, die die Wirkung von Emotionen auf das schulische Lernen und Leisten erfassen, noch weiter forciert werden müssen.“ [EH08, S. 68].

Jenna Müllener [Mül08] aus der Schweiz berichtet über aktuelle Forschungen, die zeigen, dass das individuelle Wohlbefinden einen großen Einfluss auf das erfolgreiche Lernen hat und sie fordert, dass Lernen in einem guten Lernklima geschieht:

„Lange Zeit waren sich die Forscher einig, dass die Prinzipien der Motivationsforschung für den Lernerfolg ausschlaggebender sind als das gute Klima, d.h. das Wohlbefinden der Lernenden bei ihrer Arbeit. Aktuelle Hirnphysiologen und Forscherinnen zeigen, dass das individuelle Wohlbefinden einen enorm grossen [sic!] – ja den grössten [sic!] – Einfluss auf das erfolgreiche Lernen hat.“ [Mül08, S. 17].

Dies befindet sich im Einklang mit den Forschungsergebnissen von Sembill und Seifried (vgl. [Sei04], [Sem92] und [Sem05]): Die emotionale Befindlichkeit hat Einfluss auf die Lehr-Lern-Prozesse.

Peter Hubwieser [Hub07] verweist auf die Bedeutung der (möglichst intrinsischen) Motivation und ihre Bedeutung für die Lernleistung:

„Die Beziehung zwischen Motivation und Lernen ist immer *wechselseitig*: Motivation ist Voraussetzung, aber auch Ergebnis von Lernprozessen. Erfolg beim Lernen kann über Motivationsverstärkung nach den Erkenntnissen von Fraser et al. (1987) [FWua87] die Lernleistung mehr als verdoppeln.“ [Hub07, S. 16, kursives Wort im Original, Querverweis hinzugefügt].

In der englischsprachigen Literatur ist oft vom „Emotional Climate“ oder vom „Comfort Level“ die Rede.

Wilson und Shrock [WS01] analysieren 2001 in einer Studie zwölf Faktoren, die Einfluss nehmen könnten auf den Erfolg eines Informatik-Anfangskurses:

„This study was conducted to determine factors that promote success in an introductory college computer science course. The model included twelve possible predictive factors including math background, attribution for success/failure (luck, effort, difficulty of task, and ability), domain specific self-efficacy, encouragement, comfort level in the course, work style preference, previous programming experience, previous non-programming computer experience, and gender.“ [WS01, S. 184].

Einen dieser Faktoren bezeichnen sie als „Comfort Level“:

„Comfort level - a continuous variable derived from seven questions on the questionnaire regarding asking and answering questions in class, in lab, and during office hours; anxiety level while working on computer assignments; perceived difficulty of course; perceived understanding of concepts in the course as compared with classmates; and perceived difficulty of completing the programming assignments.“ [WS01, S. 185].

Dabei kommen sie zu dem Ergebnis, dass nur drei von den zwölf Faktoren signifikant Einfluss nehmen auf den Erfolg des Kurses, davon ist der Faktor „Comfort Level“ der wichtigste:

„Comfort level in the computer science class was the best predictor of success in the course. Math background was second in importance in predicting success in this computer science class. It is most interesting, in this study, that comfort level was found to be more important than math background.“ [WS01, S. 187].

Auch McKinney und Denton [MD04] untersuchen in einer Studie subjektive Faktoren, die Einfluss auf den Erfolg von Computer-Anfangskursen nehmen. Sie finden eine Korrelation zwischen den Faktoren Interesse, „gefühlte Kompetenz“, Arbeitsaufwand und „wenig Leistungsdruck“ und dem Lernerfolg der Teilnehmer an einem Informatik-Kurs:

„This study provides quantitative results that confirm that interest, perceived

competence, effort, lack of pressure, and value are significantly correlated with CS1 course grades.“ [MD04, S. 239].

Philip Ventura [Ven04] untersucht in seiner Dissertation Einflussfaktoren für den Erfolg von Informatik-Anfangskursen und kommt zum Ergebnis, dass „student effort“ und „comfort level“ die größten Einflussfaktoren sind. Bei Bergin und Reilly [BR05] ist es von 15 untersuchten Faktoren die Kombination aus den Faktoren „student’s perception of their understanding of the module“, „gender“, „mathematics score“ und „comfort level“.

Die angeführten Studien bzw. Zitate (scheinen zu) belegen, dass subjektive Aspekte wie „emotionale Befindlichkeit“, „Wohlfühlfaktor“, „empfundene Schwere der Themen“ etc. wichtige Faktoren im Informatik-Anfangsunterricht sind. Das ist der Grund dafür, dass in dieser Studie auch die subjektiven Aspekte fokussiert werden. In den Debatten um objects-first bzw. objects-later werden zwar manchmal diese Faktoren als Argumente angeführt, aber mir ist keine Studie bekannt, die diese subjektiven Aspekte bei einem Vergleich zwischen einem OOP-First-Vorgehen und einem OOP-Later-Vorgehen heranzieht [ES10a, S. 108].

Weiterführende Informationen, inwiefern das subjektive Erleben bzw. die emotionale Befindlichkeit Platz findet im Studiendesign und wie die Datenerhebung konkret implementiert wird, finden sich u. a. im Kapitel 2.4.3 („Subjektiver Aspekt: Unterschiede im emotionalen, kognitiven und motivationalen Erleben“), Kapitel 3.2 („Forschungsdesign der empirischen Studie“) und Kapitel 3.3.13 („Fragebögen zum subjektiven Erleben“).

Nachdem der Stand der nationalen und internationale Diskussion über objects-first bzw. objects-later (im Kontext zum objektiven Aspekt des Lernerfolgs und zum Aspekt des subjektiven Erlebens von Unterricht) aufgezeigt worden ist, kann im nächsten Kapitel die Forschungsfrage dieser Arbeit abgeleitet werden.

## **2.4 Forschungsfrage und Forschungshypothesen**

### **2.4.1 Randbedingungen und Aufspaltung der Forschungsfrage**

Für diese Promotionsschrift gelten einige Randbedingungen. Zum einen existiert die Promotionsordnung des Fachbereichs Mathematik und Informatik der Freien Universität Berlin [FUB07], zum anderen gibt es eine Beschreibung der Handhabung des Promotions-Verfahrens auf einer Webseite des Fachbereichs [Pre09]. Dort heißt es z. B. unter der Überschrift „Der wissenschaftliche Beitrag“:

„Kern einer Promotion ist der Beitrag zum Stand der Wissenschaft, den sie liefert. An Umfang und Qualität dieses Beitrages wird die Arbeit gemessen.“  
[Pre09].

Der hier vorliegende wissenschaftliche Beitrag hat folgendes Ziel: Er soll dabei helfen, einige praxisrelevante Probleme im Informatik-Anfangsunterricht besser als bisher zu lösen. Die Relevanz muss hierfür belegt werden (siehe vorhergehendes Kapitel) und der Nutzen empirisch nachgewiesen werden (siehe die empirischen Ergebnisse dieser Arbeit, speziell die Auswertungen in dem Kapitel 6 („Ergebnisse und Auswertung der empirischen Studie“) und in dem Kapitel 7 („Fachdidaktische Diskussion: Spezielle Fragestellungen, Auswertungen und Interpretationen“)). Am Ende soll die Frage beantwortet werden [Pre09]: „Welches (fachdidaktisches) Problem kann (mit dem vorliegenden wissenschaftlichen Beitrag) um wie viel besser gelöst werden?“.



Lauer [Lau75] schreibt schon 1975:

„A thesis proposal in computing science should address at least the following six points:

1. A statement of the problem and why it should be solved
2. Reference to and comments upon relevant work by others on the same or similar problems
3. The candidate's ideas and insights for solving the problem and any preliminary results he may have obtained
4. A statement or characterisation of what kind of solution is being sought
5. A plan of action for the remainder of the research; and
6. A rough outline of the thesis itself.“ [Lau75, S. 279].

Im ersten Exposé (siehe Anhang A) und im Promotionsvorschlag (siehe Anhang H) wird auf diese Punkte eingegangen.

Punkt 1 wird im Kapitel 1.1 („Fachdidaktische Probleme und Relevanz der Studie“) untersucht, auf den Punkt 2 wird im Kapitel 2.1 („Deutsche Beiträge, speziell zur fachdidaktischen (objects-first) Diskussion“) bzw. 2.2 („Internationale Beiträge, speziell zu bisherigen empirischen Ergebnissen der objects-first-Debatte“) eingegangen, die Punkte 3 und 4 werden u. a. im Kapitel 2.4 („Forschungsfrage und Forschungshypothesen“) und Kapitel 3.1 („Ziele der empirischen Studie“) behandelt und die Punkte 5 und 6 finden sich im Kapitel 3.2 („Forschungsdesign der empirischen Studie“), im Kapitel 3.3.2 („Implementierung in Bezug auf das Forschungsdesign“) und im Kapitel 3.3.3 („Zeitliche Dimension der empirischen Studie (Untersuchungsplan)“) wieder.

Die Forschungsfrage baut auf dem Stand der fachdidaktischen Wissenschaft (siehe Kapitel 2.1 und Kapitel 2.2) auf, die sich mit den Unterschieden zwischen objects-first und objects-later und den Vor- und Nachteilen einzelner fachdidaktischer Vorgehensweisen beschäftigt. Was aber in der inzwischen 20-jährigen Diskussion fehlt, ist ein direkter Vergleich zwischen objects-first und objects-later in zwei vergleichbaren Kursen desselben Jahrgangs, an derselben Bildungseinrichtung, mit vergleichbaren Schülern bzw. Studenten, mit derselben Programmiersprache, mit einem seriösen Untersuchungsdesign und bei reduzierter bzw. stark kontrollierter Variablenanzahl.

Somit wird die Forschungsfrage der vorliegenden Arbeit, die indirekt im Titel dieser Dissertationsschrift aufgeführt ist, wie folgt formuliert:

„Gibt es Unterschiede im Lernerfolg und Unterschiede im subjektiven Erleben des Unterrichts von Schülerinnen und Schülern im Informatik-Anfangsunterricht (11. Klasse Berufliches Gymnasium) in Abhängigkeit von der zeitlichen Reihenfolge der Themen (OOP-First und OOP-Later)? *Wenn ja, welche?*“

Somit besteht die Forschungsfrage eigentlich aus zwei Fragen, denn diese empirische Studie will den Fokus auf zwei verschiedene Aspekte des Schüler-Lernprozesses richten: den subjektiven Aspekt und den objektiven Aspekt [ES09b, Folie 11].

Die Forschungsfrage zum objektiven Aspekt lautet:

*„Gibt es Unterschiede im Lernerfolg von Schülerinnen und Schülern im Informatik-Anfangsunterricht in Abhängigkeit eines OOP-First- bzw. OOP-Later-Vorgehens? Wenn ja, welche?“*

Die Forschungsfrage zum subjektiven Aspekt lautet:

*„Gibt es Unterschiede im subjektiven Erleben des Unterrichts von Schülerinnen und Schülern im Informatik-Anfangsunterricht in Abhängigkeit eines OOP-First- bzw. OOP-Later-Vorgehens? Wenn ja, welche?“*

Im Folgenden sollen beide Aspekte und ihr möglicher Zusammenhang näher erläutert werden.

## 2.4.2 Objektiver Aspekt: Unterschiede im Lernerfolg

Der objektive Aspekt setzt den Fokus auf den Lernerfolg der Schülerinnen und Schüler nach einem ganzen Schuljahr. Überprüft wird dies mit zwei klassischen (Lernziel-) Tests: Einem Vergleichstest (Post-Test) direkt nach dem Schuljahr und einem Nachhaltigkeitstest (Follow-Up-Test) acht Wochen später.

Vorher müssen für die Studie aber verschiedene Fragen beantwortet werden:

- Wie lauten die einzelnen Themen?
- Was sind die Lernziele?
- Wie werden aus den Lernzielen die Indikatoren (die Testfragen) abgeleitet?
- Wie ist die Verteilung der Fragen im Hinblick auf die Anforderungsbereiche?
- Welchen Umfang darf der Test haben?
- Soll der Test auch mit „echter Programmierung“ im Computerlabor stattfinden?
- Wie werden die Kriterien für einen hochwertigen Test erfüllt?
- Ist es ein Problem, dass einige Themen schon länger zurück liegen als andere?

Das gesamte Kapitel 4 („Lernerfolg: Lernziele, Indikatoren und Tests“) gibt Antworten auf diese Fragen und stellt die Testfragen (Vor-Test, Vergleichs-Test und Nachhaltigkeits-Test) vor (siehe auch Kapitel 3.3: „Implementierung der empirischen Studie“, speziell Kapitel 3.3.12).

Durch den Vergleichs- und den Nachhaltigkeits-Test soll herausgefunden werden, ob es (signifikante) Unterschiede bei den Ergebnissen der Tests in Abhängigkeit von der fachdidaktischen Vorgehensweise gibt (vgl. Kapitel 6.3). Damit wird eine Antwort auf die Forschungsfrage zum objektiven Aspekt gegeben. Es geht bei Unterricht immer darum, ob und was gelernt wird. Auch wenn diese Arbeit noch auf der Idee von „Lernzielen“ („Input-Orientierung“) beruht und nicht Kompetenzen („Output-Orientierung“) formuliert, so sind die Ergebnisse eines abschließenden Tests eine Aussage zu der Frage: „Was ist konkret vermittelt worden?“.

Eine eindeutige Antwort (vgl. Kapitel 7.3) darauf würde m. E. die „objects-first-or-not-first-Debatte“ versachlichen. Ich sehe Tendenzen, dass in regelmäßigen Abständen jemand einen Beitrag veröffentlicht, dass er „so oder so“ vorgegangen sei bzw. „ein neues tolles Tool“ benutzt und damit „sehr gute Erfahrungen“ gemacht habe und die Schüler „sehr angetan“ gewesen seien und „viel gelernt“ hätten.

Es wäre einerseits hilfreich für die Debatte, wenn diese empirische Studie zu einem eindeutigen Ergebnis käme und wenn man dann darüber diskutieren könnte, warum es (signifikante) Unterschiede bei den Lernergebnissen gibt.

Es wäre aber andererseits genauso hilfreich, wenn keine oder nur partielle Unterschiede gefunden würden und sich die teilweise sehr emotional geführte Debatte dadurch als überflüssig erweise.

Unabhängig von dem objektiven Aspekt könnte es noch einen anderen Aspekt geben, nämlich das subjektive Erleben der Schüler in Abhängigkeit von der fachdidaktischen Vorgehensweise. Warum dieser subjektive Aspekt wichtig ist, wird im nächsten Kapitel erläutert.

### 2.4.3 Subjektiver Aspekt: Unterschiede im emotionalen, kognitiven und motivationalen Erleben

Zusätzlich zu dem objektiven Aspekt des Lernerfolgs wird in dieser Arbeit ein weiterer Aspekt in die Diskussion eingebracht (siehe Kapitel 2.3), nämlich der Aspekt des subjektiven Erlebens von Informatik-Anfangsunterricht durch die Schülerinnen und Schüler. Dazu gehören z. B. die emotionale Befindlichkeit (vgl. [Sem92]) und das Wohlfühlen (vgl. [WS01]) bzw. das Wohlbefinden (vgl. [Mül08]). Es wird untersucht, ob das subjektive Erleben bei beiden fachdidaktischen Vorgehensweisen gleich ist. Dabei wird die Einteilung von Sembill und Seifried (vgl. z. B. [Sem05] und [Sei04]) übernommen, die das Erleben von Unterricht untersucht haben, bezogen auf die folgenden drei Aspekte:

- Emotionaler Aspekt
- Kognitiver Aspekt
- Motivationaler Aspekt

Seifried und Klüber [SK06] verfassen 2006 einen Beitrag, in dessen Zentrum der Vergleich des emotionalen, kognitiven und motivationalen Erlebens des Unterrichts in Abhängigkeit von fachdidaktischen Schwerpunktsetzungen steht. Die Ausführungen basieren dabei auf in kurzer zeitlicher Taktung erhobenen Prozessdaten zum subjektiven Erleben des Unterrichts aus Sicht von Schülerinnen und Schülern. Die Erlebensdaten („Innensicht“) werden dann mit einer Unterrichtsbeobachtung („Außensicht“) in Verbindung gebracht.

Die vorliegende Studie orientiert sich an dieser Sichtweise, allerdings ist die zeitliche Taktung wesentlich länger. In einem Schuljahr werden nach jedem Thema (ca. alle 3-4 Wochen) die gleichen Fragen zur emotionalen Befindlichkeit bzw. zum emotionalen, kognitiven und motivationalen Erleben der letzten Unterrichtsstunden zu diesem Thema gestellt.

Die „Außenansicht“ ist dabei wie folgt:

- Das subjektive Erleben wird in den Kontext der beiden fachdidaktischen Vorgehensweisen gestellt und somit auch auf die Reihenfolge der unterrichteten Themen bezogen.
- Das subjektive Erleben wird mit dem objektiven Aspekt, also den Testergebnissen, in Verbindung gebracht.

Im Vordergrund stehen dabei z. B. die folgenden Fragestellungen [ES09b, Folie 12]:

- Wie erleben die Schülerinnen und Schüler das unterrichtete Thema, den Unterricht und ihren Lernprozess?
- Wie erleben die Schülerinnen und Schüler das Thema, z. B. im Hinblick auf den Schwierigkeitsgrad?
- Wie groß ist ihr Interesse und ihre Motivation im bzw. am Informatikunterricht?

Diese Fragestellungen sind wichtig, da das emotional-motivationale Erleben des Unterrichtsgeschehens und die Intensität und Qualität der Motivation den Lernerfolg mitbestimmen [Sem05]. Die Theorie, dass ein positives subjektive Erleben des Unterrichts positive Auswirkungen auf den Lernerfolg hat, ist vielfach belegt (siehe Kapitel 2.3).

Das gesamte Kapitel 5 („Subjektives Erleben des Unterrichts: Dimensionen, Frage-Bereiche und Fragestellungen“) gibt Antworten auf diese Fragen und stellt am Ende den Fragebogen vor (siehe auch Kapitel 3.3: „Implementierung der empirischen Studie“, speziell Kapitel 3.3.13).

Durch den Fragebogen soll herausgefunden werden, ob es (signifikante) Unterschiede bei dem subjektiven Erleben des Unterrichts in Abhängigkeit von der fachdidaktischen Vorgehensweise gibt (vgl. Kapitel 6.4). Damit wird eine Antwort auf die Forschungsfrage zum subjektiven Aspekt gegeben. Obwohl im Unterricht immer die Hauptpriorität auf dem Lernerfolg liegt, wäre es trotzdem schön, wenn der Unterricht positiv erlebt werden würde, was wieder positive Rückwirkung auf den Lernerfolg haben könnte.

Da die Fragebögen nach jedem Thema von den Schülerinnen und Schülern ausgefüllt werden und die Tests für den Lernerfolg auch themenbasiert sind, können Korrelationen zwischen subjektivem Erleben (siehe Kapitel 6.4) und Lernerfolg (siehe Kapitel 6.3) durch diese Studie herausgefunden werden (siehe Kapitel 7.4.1).

#### 2.4.4 Erstes Exposé

Sehr interessant ist ein erstes Exposé für die FU Berlin aus dem Jahre 2006 (siehe Anhang A). Daraus wird auch der dynamische Prozess dieser Dissertation ersichtlich.

Der dort genannte Arbeitstitel lautet noch: „Fachdidaktische Untersuchung zur OOP im Anfängerunterricht bei relativ leistungsschwachen Schülern“.

Der Hinweis auf die leistungsschwachen Schüler ist aus dem Dissertationstitel verschwunden. Ursache war das Problem, wie man wissenschaftlich nachweist, dass das Berufliche Gymnasium am OSZ IMT eher von leistungsschwachen Schüler besucht wird (im Vergleich zu bodenständigen Gymnasien). Es wird jetzt nur auf die Tatsache hingewiesen, dass 31 von 44 Schülern von Gesamt- bzw. Realschulen kommen (siehe Abbildung 26).

Als Anlass für die Untersuchung wird meine Erfahrung genannt, dass sich speziell leistungsschwache Schüler mit der objektorientierten Modellierung und der objektorientierten Programmierung schwer tun.

Bei der Hauptstudie werden spezielle Teilgruppen untersucht, u. a. die Gruppe der leistungsschwächsten Schüler. Als Idee wird angeführt, ob es vielleicht für dieses Schülerklientel leichter ist (und von mehr Lernerfolg gekrönt), wenn der Informatikunterricht imperativ beginnt und erst nach den „Kontroll- und Datenstrukturen“ auf die „Begrifflichkeiten der OOP“ eingegangen wird.

Hier finden sich also schon die beiden Ideen des subjektiven und objektiven Aspekts.

Bei den Lerninhalten ist von 20 Modulen die Rede. Der Begriff Modul wird später durch den Begriff Thema (Begründung: siehe Kapitel 3.3.8) ersetzt und die Anzahl der Themen auf neun gesenkt.

Im Jahre 2006 wurden von mir auch erste Thesen formuliert (siehe nächstes Kapitel). Auch wenn diese Arbeit sich im Laufe der Zeit immer mehr geöffnet hat zu einer „objektiveren Betrachtung“ beider fachdidaktischen Vorgehensweisen, möchte ich nicht verschweigen, dass meine ersten Ideen als ein Plädoyer für ein OOP-Later-Vorgehen angesehen werden können.

Die Freie Universität Berlin bat mich im Februar 2008, ein weiteres Exposé als so genannten Promotionsvorschlag zu erstellen und das Thema der Promotion

„[...] einer kleinen aber feinen Schar fachkundiger und kritischer Personen vorzustellen [...].“ [Pre09].

Den Vortrag [Ehl08a] hielt ich dann vor einigen Professoren, Studenten, Angestellten und Doktoranden der FU Berlin. Der Promotionsvorschlag findet sich in Anhang H wieder und ist ein weiteres Indiz für den dynamischen Aspekt dieser Arbeit.

## 2.4.5 Ursprüngliche Thesen

Zunächst sah ich die Ursache für die Lernschwierigkeiten meiner Schüler im Informatik-Anfangsunterricht einerseits im OOP-First-Ansatz und andererseits in der Vermittlung des objektorientierten Paradigmas. Unterstützung fand ich bei Helmut Balzert, der auch schlechte Erfahrungen mit seinen Erstsemester-Studenten gemacht hatte (siehe Abbildung 10):



Abbildung 10: Anfrage an Prof. Balzert und seine Antwort [Ehl06, Folien 14 und 15]

Auch erschien es mir, dass die Vermittlung der OOP teilweise ein Selbstzweck war. In meinen ersten Vorträgen (z. B. [Ehl06]) stellte ich provokativ die Fragen (siehe auch Abbildung 11 und Abbildung 12):

- „Ist anstelle des Problemlösungsziels der Informatik jetzt das OOP-Vermittlungsziel getreten?“
- „Wird das OO-Paradigma zum „OO-Paradogma“?“

Auch Humbert [Hum06] sieht weiterhin das Problemlösen als zentrale Kategorie des Informatikunterrichts. Er gibt Schemata (z. B. imperativ und objektorientiert) zum Problemlösen für den Einsatz im Unterricht vor [Hum06, Anhang E].

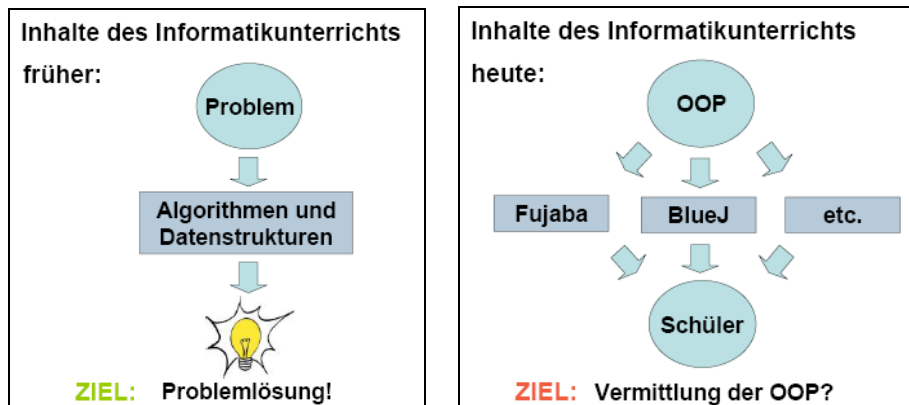


Abbildung 11: Inhalte des Informatikunterrichts (früher und heute) [Ehl06, Folien 6 und 7]

Zu der Vermittlungsziel-Problematik kommt m. E. ein weiteres Problem hinzu: Mit einem OOP-First-Einstieg werden den Schülern zu viele Fachtermini auf einmal genannt. Dies kann zu einer Überforderung führen.

**Gedanken:**

- Siegfried Spolwig: (Probleme im OOP-Anfängerunterricht) ... lassen eher Mängel in der didaktisch-methodischen Aufbereitung vermuten.
- m. E. „Anleitung zum Unglücklichsein“!
- Alle Hilfen sind Ausdruck eines Vermittlungsproblems!
- Das OOP-Paradigma wird teilweise zum „Paradogma“!
- Wozu muss eine Taschenlampe modelliert werden?

Abbildung 12: Erste Gedanken zu den Problemen im OOP-Anfangsunterricht [Ehl06, Folie 8, leicht modifiziert]

Balzert [BB04] schreibt dazu:

„Da aber zu einem und demselben Zeitpunkt nicht alles auf einmal vermittelt werden kann, bleibt zunächst ein „Erkenntnisnebel“ durch unbekannte Konzepte bestehen, der sich erst nach und nach lichtet.“ [BB04, S. 24].

Eine Seite vorher schreibt er zu einem Unterricht, welcher mit der Einführung von Objekten und Klassen beginnt:

„Nachteilig dabei ist, dass am Anfang vieles als gegeben und als noch nicht erklärt hingenommen werden muss.“ [BB04, S. 23].

Korber [Koe04] schreibt:

„Der Weg vom Problem zum Programm über die Phasen des Abstrahierens, Modellierens und Codierens gehört zu den Grundpfeilern informatischer Bildung. Neu ist dagegen die Sichtweise auf die Objekte der Realität. [...] Standen früher vor allem Algorithmen im Zentrum des Informatikunterrichts,

so besteht durch die Orientierung auf Objekte nunmehr nahezu ein Zwang, sich mit dem Modellieren dieser Objekte zu beschäftigen. Das bedeutet aber nicht, dass die Umsetzung in ein ablauffähiges Programm im Unterricht vernachlässigt werden darf.“ [Koe04, S. 3].

Die verschiedenen lernunterstützenden Tools (Fujaba, BlueJ etc., siehe auch Abbildung 11 und Abbildung 16) waren für mich nur Indikatoren dafür, dass es Schwierigkeiten bei der Vermittlung des objektorientierten Paradigmas gibt (siehe auch Abbildung 12). Auch die Themen- und Modellierungsbeispiele in fachdidaktischen Veröffentlichungen waren m. E. nicht immer nachzuvollziehen (siehe auch Kapitel 2.1). Mir erschloss sich nicht unbedingt der Sinn, eine Taschenlampe (siehe Abbildung 8 und Abbildung 12) zu modellieren.

Ich habe aus diesen kritischen Überlegungen heraus dann eigene Thesen entwickelt. Meine spätere „offenere“ Sichtweise resultiert aus jahrelanger Arbeit an dem Thema. Allerdings formulierte ich schon im Jahr 2006 die Ansicht, dass jede fachdidaktische Vorgehensweise ihre Vor- und Nachteile hat und es keinen Königsweg geben wird (siehe auch Abbildung 13).

**Resümee:**

Wir haben m. E. teilweise ein Vermittlungsproblem bei der „OOP im Anfängerunterricht“.

Jedes fachdidaktische Vorgehen hat seine Vor- und Nachteile.

Es gibt somit keinen Königsweg.

**Aber wir sollten offen sein für die Probleme der Schüler und die Schüler wieder in den Mittelpunkt unserer Überlegungen stellen (nicht die OOP)!**

Abbildung 13: Erstes Resümee am Beginn der Studie [Ehl06, Folie 21, leicht modifiziert]

Die gesamte Arbeit hat also eine dynamische Komponente und dies empfinde ich als positiv. Auch wenn sich einige Thesen als nicht haltbar erwiesen, möchte ich sie trotzdem nicht unerwähnt lassen. Die Thesen befinden sich dabei noch auf einem populärwissenschaftlichen Niveau, es ist aber schon erkennbar, dass daraus Forschungshypothesen gewonnen werden können, die dann in statistische Hypothesen überführt werden können bzw. müssen [BL03, S. 30]. Mehr dazu findet sich im Kapitel 2.4.6 („Forschungshypothesen“).

**These1:**

**Viele Themen der „OOM, OOA, OOD bzw. OOP“ sind auf einem höheren „Lern-Schwierigkeits-Niveau“ als z. B. das Thema „Kontrollstrukturen“ in der imperativen Programmierung.**

**These2:**

**Das von der allgemeinen Lerntheorie abgeleitete Prinzip „vom Einfachen zum Schwierigen“ sollte auch bei der „OOP im Anfängerunterricht“ beachtet werden. Dies gilt umso mehr, je leistungsschwächer die Schüler sind.**

Abbildung 14: These 1 und These 2 aus dem Jahr 2006 [Ehl06, Folie 9 und 10, leicht modifiziert]

**These3:**

**Die Abfolge der Lerneinheiten muss so erfolgen, dass die Aussage aus These 2 berücksichtigt wird: Der Unterricht führt vom Leichten zum Schwierigen, und es werden nicht zu viele Dinge auf einmal unterrichtet.**

**These4:**

**Ein Anfänger-Unterrichts-Einstieg mit „Programmieren im Kleinen“ ohne OOP-Bezug steigert deutlich den „Wohlfühlfaktor“ im Unterricht und somit auch die Motivation bzw. die Erfolgserlebnisse des einzelnen (leistungsschwachen) Schülers.**

Abbildung 15: These 3 und These 4 aus dem Jahr 2006 [Ehl06, Folie 11 und 12, leicht modifiziert]

Die ersten Thesen stammen aus der Anfangszeit der Studie im Jahr 2006 und wurden im Rahmen eines Vortrages auf dem MNU-Kongress in Berlin (siehe Abbildung 14 und Abbildung 15) vorgestellt.

Eine der ersten Forschungsfragen lautete damals: „Wie erleben eigentlich die Schülerinnen und Schüler den Einstieg in die OOP mit dem OOP-First- und mit dem OOP-Later-Ansatz?“.

Eine mögliche Antwort auf diese Frage soll im Folgenden durch das Beispiel eines Einstiegs in die OOP mit dem BlueJ-Tool gegeben werden [Ehl07b, S. 18-19].

Barnes und Kölling [BK06] schreiben zu ihrem BlueJ-Buch:

„Dieses Buch ist eine Einführung in die objektorientierte Programmierung für Programmieranfänger.“ [BK06, S. 16].

Das erste BlueJ-Beispiel [BK06, S. 32] ist das Projekt „Figuren“, welches m. E. eine Vielzahl von Fachtermini enthält: Klassen, Objekte, Zugriffsspezifizierer, Methoden, Parameter und Attribute (vgl. Abbildung 16 und Abbildung 17).

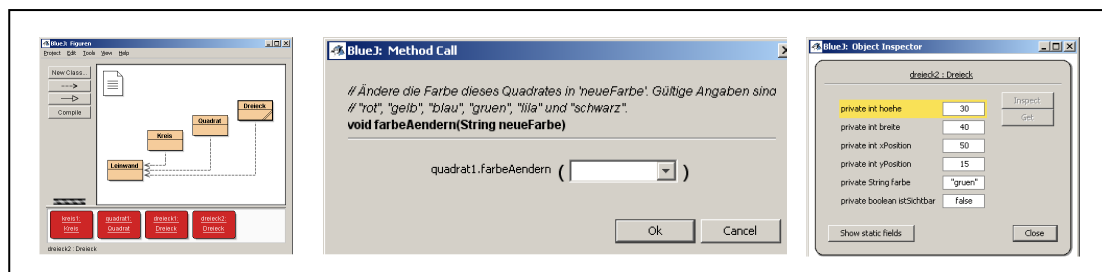


Abbildung 16: Beispiel für die Komplexität der ersten BlueJ-Übung [vgl. Ehl07b, S. 18]

Wird diese Entwicklungsumgebung für den OOP-First-Einstieg genutzt, dann wissen die Schülerinnen und Schüler noch nichts über Datentypen, Variablen, Konstanten, Schleifen und Verzweigungen [Ehl07b, S. 19]. In einem der ersten Vorträge [Ehl07c] veranschaulichte ich das Problem durch ein Bild mit zwei Jugendlichen vor einem Holzstapel (siehe Abbildung 17) nach dem Motto: „Viel Holz auf einmal zum Lernen“.



Abbildung 17: Visualisierung der Anzahl der BlueJ-Teilthemen bei OOP-First [Ehl07c, Folie 6]

Der Lehrer wird dieses Beispiel anders präsentieren als ein Lehrer, der über einen OOP-Later-Weg bei dem Übergang zur objektorientierten Programmierung dieses Beispiel einsetzt und schon viele Begrifflichkeiten bei den Schülern voraussetzen kann, wie z. B. die Begriffe Datentypen, Variablen, Methoden und Parameter [Ehl07b, S. 18].

Die OOP-Later-Schüler mit ihrem Vorwissen werden evtl. das Unterrichtsthema bzw. das Unterrichtsbeispiel leichter erleben. Visualisiert habe ich diesen möglichen Sachverhalt in den Vorträgen [z. B. Ehl07c] durch ein



neues Bild (siehe Abbildung 18), auf dem der Holzstapel deutlich kleiner geworden ist, nach dem Motto: „Weniger Holz auf einmal zum Lernen!“.



Abbildung 18: Visualisierung der Anzahl der BlueJ-Teilthemen bei OOP-Later [Eh107c, Folie 29]

Der folgende Abschnitt ist teilweise wörtlich übernommen aus dem Promotionsvorschlag (siehe Anlage H)). In diesem sind die folgenden drei Thesen formuliert.

- These 1: Das Thema OOP gleich zu Beginn der ersten Informatik-Stunden wird von den Schülerinnen und Schülern als schwieriger erlebt, als wenn es erst nach einer längeren „Einführung in die prozedurale Programmierung“ erfolgt (vgl. Abbildung 19 und Abbildung 20).

Die von der allgemeinen Lerntheorie herstammenden didaktischen Prinzipien „Vom Leichten zum Schwierigen“, „Neue Informationen mit altem Wissen in Verbindung setzen“ und „Nicht zu viele neue Informationen auf einmal“ sollten auch bei der Programmierung im Anfangsunterricht eingehalten werden (vgl. auch [Küc06]). Dies gilt umso mehr, je leistungsschwächer die Schüler sind. Es spricht somit m. E. einiges dafür, vor der Einführung in die OOP erst einmal grundlegendes Wissen wie Datentypen, Selektionen und Iterationen zu vermitteln, da dann der Unterricht vom Leichten zum Schwierigen führt, auf Bekanntem aufbaut und nicht zu viele neue Begriffe auf einmal einführt.

- These 2: Ein Unterrichts-Einstieg mit „Programmieren im Kleinen“ ohne OOP-Bezug steigert den „Wohlfühlfaktor“ im Unterricht und somit auch die Motivation bzw. die Erfolgserlebnisse des einzelnen (leistungsschwachen) Schülers und verbessert somit auch die Lernergebnisse. Parallel dazu wird der Schwierigkeitsgrad und die Komplexität einiger Themen verringert (vgl. Abbildung 19 und Abbildung 20).

Es werden aber in der fachdidaktischen Diskussion auch viele Argumente für das OOP-First-Vorgehen angeführt (eine Art Antithese), z. B.: Durch die OO-Begrifflichkeiten von der ersten Stunde an wird das objektorientierte Paradigma viel besser von den Schülern verinnerlicht. Durch die Vielzahl der Begrifflichkeiten (inkl. der prozeduralen Begrifflichkeiten) werden diese unterschiedlichen Themen viel häufiger angesprochen bzw. immer wieder angesprochen (im Sinne eines Spiralmodells) und somit besser gelernt.

Wahrscheinlich ist (als eine Art Synthese), dass beide Vorgehensweisen spezielle Vor- und Nachteile haben.

Wenn man die Programmier-Lehrbücher (mit den verschiedensten Ansätzen) im Hinblick auf die einzelnen Programmier-Themen betrachtet, finden sich immer wieder gleichartige Kapitel bzw. Sequenzen, kleine Unterschiede ausgenommen (siehe Kapitel 1.2.2). Dies ist ein erster Beleg für die nächste These.

- These 3: Der OOP-First-Ansatz unterscheidet sich von dem OOP-Later-Ansatz überwiegend nur in der unterschiedlichen Reihenfolge der Themen. Daher können für beide Ansätze gleichartige Themen benannt werden (vgl. Abbildung 19 und Abbildung 20).

Abbildung 19 soll die drei Thesen veranschaulichen: In den sechs OOP-First-Themen finden sich vier prozedurale Themen, die dem OO-Unterricht vorgelagert werden können (Zahlen wurden willkürlich gewählt). Die Themen bleiben gleichartig, der Gesamt-Schwierigkeitsgrad der Themen sinkt allerdings.

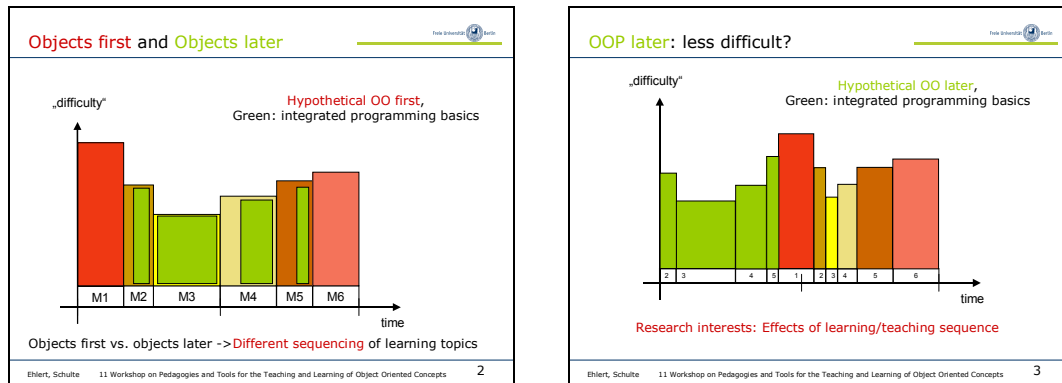


Abbildung 19: Erste schematische Veranschaulichung der 3 Thesen [ES07b, Folie 2 und Folie 3]

Auch die Abbildung 20 soll die drei Thesen veranschaulichen: Der OOP-First-Einstieg mit seiner Vielzahl an Themen befindet sich auf einem relativ hohen Schwierigkeits- und Komplexitätsgrad. Verlagert man nun einige prozedurale Themen an den Anfang, dann gestaltet sich der spätere OOP-Einstieg für die Schüler leichter. Trotzdem sind die Themen gleichartig.

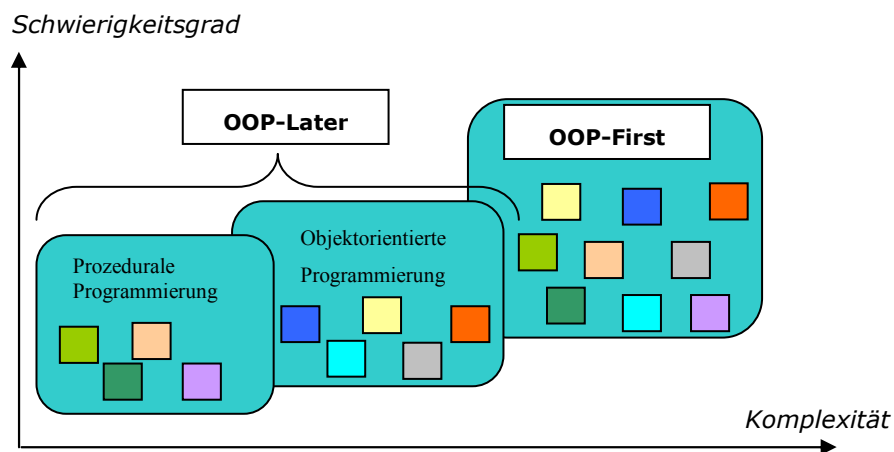


Abbildung 20: Zweite schematische Veranschaulichung der 3 Thesen [Ehl08b, Folie 13, leicht modifiziert]

Damit sind die ursprünglichen Thesen inkl. Begründungszusammenhang dokumentiert (siehe auch Anhang H). Diese inhaltlichen Hypothesen müssen jetzt methodisch in Forschungshypothesen umgewandelt werden.

## 2.4.6 Forschungshypothesen

Im Folgenden sollen aus den beiden Forschungsfragen (siehe Kapitel 2.4.1) und den Thesen aus dem Jahr 2006 (die man als inhaltliche Forschungshypothesen bezeichnen könnte) die statistischen Forschungshypothesen

(Alternativhypothesen und Nullhypothesen) entwickelt werden [BL03, S. 14-60]. Um die Gliederungsstruktur dieses Kapitels deutlicher zu machen, werden manche Fachbegriffe kursiv hervorgehoben.

Eine der Ursprungsthesen braucht bzw. kann nicht durch statistisches Datenmaterial belegt werden: „Der OOP-First-Ansatz unterscheidet sich von dem OOP-Later-Ansatz überwiegend nur in der unterschiedlichen Reihenfolge der Themen. Daher können für beide Ansätze gleichartige Themen benannt werden.“ (vgl. These 3, Kapitel 2.4.5).

Diese These ist ja zum einen als Kernelement der Definition des Unterschieds zwischen OOP-First und OOP-Later (siehe Kapitel 1.2.3) schon in die Arbeit (und in den Dissertations-Titel) eingeflossen, zum anderen kann man ja an dieser Stelle schon auf ein Ergebnis der Studie (siehe Kapitel 7.5.1: „Können die ursprünglichen Thesen bestätigt werden?“) verweisen: Es konnten neun Themengebiete generiert werden (siehe Kapitel 1.2.2 und Kapitel 3.3.8), die sowohl beim OOP-First-Ansatz als auch beim OOP-Later-Ansatz zur Geltung kamen. Nach einem Jahr sind in der Hauptstudie alle neun Themengebiete unterrichtet worden, nur die zeitliche Reihenfolge der Themen variierte.

Bei den anderen Hypothesen muss entschieden werden, ob gerichtete oder ungerichtete Forschungshypothesen formuliert werden. Dies hat nämlich statistische Auswirkungen.

Bortz und Lienert [BL03] schreiben:

„Die Art der Hypothese, ob gerichtet oder ungerichtet, muss bereits vor der Gewinnung von Beobachtungs- oder Versuchsdaten festgelegt werden. Im anderen Fall könnte man sie so formulieren, wie es die Ergebnisse nahelegen. Wie wir noch sehen werden, ist es leichter, eine gerichtete Hypothese als zutreffend anzunehmen als eine ungerichtete [...]“ [BL03, S. 31].

Die Ursprungsideen für diese Studie sind alles *gerichtete Alternativhypothesen* ( $H_1: \mu_A > \mu_B$ ):

1. Forschungshypothese: „Der Lernerfolg von Schülerinnen und Schülern ist bei einem OOP-Later-Vorgehen größer als bei einem OOP-First-Vorgehen“ (vgl. erste Forschungsfrage, Kapitel 2.4.1).

Hinweis: Der Lernerfolg muss aufgeschlüsselt werden, z. B. nach Themen.

2. Forschungshypothese: „Das subjektive Erleben von Schülerinnen und Schülern ist bei einem OOP-Later-Vorgehen „besser“ als bei einem OOP-First-Vorgehen“ (vgl. zweite Forschungsfrage, Kapitel 2.4.1).

Hinweis: Hierfür muss noch definiert werden, in welchen Items sich subjektives Erleben ausdrückt und was „besser“ bedeutet. Dazu gehören aber sicher die folgenden Hypothesen:

2.1. Forschungshypothese: „Der „Wohlfühlfaktor“ von Schülerinnen und Schülern ist bei einem OOP-Later-Vorgehen „größer“ als bei einem OOP-First-Vorgehen“ (vgl. These 2, Kapitel 2.4.5).

2.2. Forschungshypothese: „Das „Schwere-Empfinden“ der Schülerinnen und Schüler bezogen auf die einzelnen Themen ist bei einem OOP-Later-Vorgehen „kleiner“ als bei einem OOP-First-Vorgehen“ (vgl. These 2, Kapitel 2.4.5).

- 2.3. Forschungshypothese: „Das Thema „Einführung in die OO(P)“ wird von den Schülerinnen und Schülern bei einem OOP-Later-Vorgehen „leichter“ erlebt als bei einem OOP-First-Vorgehen“ (vgl. These 1, Kapitel 2.4.5).
- 2.4. Forschungshypothese: „Objektorientierte Themen befinden sich (in der Regel) für die Schülerinnen und Schüler auf einem „höheren Schwierigkeitsgrad“ als prozedurale Themen“ (vgl. These 2 und Abbildung 20, Kapitel 2.4.5).

Dementsprechend lauten die *Nullhypothesen* ( $H_0: \mu_A \leq \mu_B$ ):

1. Nullhypothese: „Der Lernerfolg von Schülerinnen und Schülern ist bei einem OOP-Later-Vorgehen „kleiner oder gleich“ als bei einem OOP-First-Vorgehen.“
  2. Nullhypothese: „Das subjektive Erleben von Schülerinnen und Schülern ist bei einem OOP-Later-Vorgehen „gleich oder schlechter“ als bei einem OOP-First-Vorgehen.“
- Somit gibt es auch die Nullhypothesen 2.1., 2.2., 2.3. und 2.4.

Bortz und Lienert [BL03] schreiben:

„Ein einseitiger Test wird also eher signifikant als ein zweiseitiger Test. Dies begründet noch einmal [...] die Notwendigkeit, die Art der Alternativhypothese – gerichtet oder ungerichtet – vor der Datenerhebung festzulegen. Das „Umwecheln“ von einer ungerichteten  $H_1$  auf eine gerichtete  $H_1$  angesichts der erhobenen Daten ist nicht statthaft, vor allem wenn dadurch ein nicht signifikanter zweiseitiger Test zu einem signifikanten einseitigen Test wird. Wenn man die Richtung der Alternativhypothese vor Untersuchungsbeginn nicht sinnvoll begründen konnte, muss man konsequenterweise auch zweiseitig testen.“ [BL03, S. 46].

Obwohl für alle Alternativhypothesen fachdidaktische sinnvolle Begründungen gegeben wurden (siehe Kapitel 2.4.5), wird bei der statistischen Auswertung der Daten dieser Studie (siehe Kapitel 6) zweiseitig getestet im Sinne einer „objektiveren Betrachtung“ beider fachdidaktischen Vorgehensweisen.

Der zweiseitige Test würde von den folgenden *ungerichteten Alternativhypothesen* ( $H_1: \mu_A \neq \mu_B$ ) ausgehen:

- „Der Lernerfolg von Schülerinnen und Schülern bei einem OOP-Later-Vorgehen unterscheidet sich von dem Lernerfolg bei einem OOP-First-Vorgehen.“
- „Das subjektive Erleben von Schülerinnen und Schülern bei einem OOP-Later-Vorgehen unterscheidet sich von dem subjektiven Erleben bei einem OOP-First-Vorgehen.“

Dementsprechend lauten dann die *Nullhypothesen* ( $H_0: \mu_A = \mu_B$ ):

- „Der Lernerfolg von Schülerinnen und Schülern bei einem OOP-Later-Vorgehen ist gleich dem Lernerfolg bei einem OOP-First-Vorgehen.“
- „Das subjektive Erleben von Schülerinnen und Schülern bei einem OOP-Later-Vorgehen ist gleich dem subjektiven Erleben bei einem OOP-First-Vorgehen.“

## 2.5 Zusammenfassung

Die deutschen und die internationalen Beiträge (siehe Kapitel 2.1 und Kapitel 2.2) zur objects-first-Diskussion dokumentieren, dass es bis zum heutigen Tag eine offene Diskussion über angemessene didaktisch-methodische Konzepte zur Vermittlung des objektorientierten Paradigmas gibt. Diese umfasst auch die Frage, ob die Idee der „objects first“ überhaupt sinnvoll ist oder nicht. Gerade empirische Untersuchungen zu diesem Thema sind zum einen eher selten, zum anderen fehlt es an allgemein verbindlichen (generalisierenden) Aussagen bzw. Ergebnissen. Oft sind die Variablen zu speziell oder sie variieren zu stark. Bortz und Lienert [BL03] schreiben:

„Der Wert einer Forschungsarbeit hängt von der Eindeutigkeit der Ergebnisinterpretation (interne Validität) und von der Generalisierbarkeit der Untersuchungsergebnisse ab (externe Validität).“ [BL03, S. 17].

Die offene fachdidaktische Diskussion und der Mangel an generalisierbaren Ergebnissen bzw. Aussagen sind die Grundlage für die Forschungsfrage (siehe Kapitel 2.4.1). Diese wird unterteilt in einen objektiven Aspekt (Unterschiede im Lernerfolg, siehe Kapitel 2.4.2) und in einen subjektiven Aspekt (Unterschiede im emotionalen, kognitiven und motivationalen Erleben des Unterrichts, siehe Kapitel 2.4.3).

Parallel zu der Forschungsfrage wurde anhand von Beiträgen und Vorträgen aus der Anfangszeit der Studie (ab dem Jahr 2006) dokumentiert, welches die ersten Ideen, Ziele und Thesen waren (siehe Kapitel 2.4.4 und Kapitel 2.4.5). Ausgehend von diesen Ursprungsideen bzw. Ursprungthesen wurden dann die Forschungshypothesen entwickelt (siehe Kapitel 2.4.6), die Bezug sowohl auf den objektiven als auch auf den subjektiven Aspekt nehmen.

Speziell in dem Kapitel 7.5.1 („Können die ursprünglichen Thesen bestätigt werden?“) und in dem Kapitel 7.5.2 („Können die Forschungsfragen beantwortet werden?“) werden die Ergebnisse der Studie bezogen auf die Forschungsfrage bzw. die Forschungshypothesen zusammengefasst.



## 3 Ziele, Forschungsdesign und Implementierung der empirischen Studie

Im Kapitel 3 werden die Ziele, das Forschungsdesign und die Implementierung der empirischen Studie erläutert, wobei der Schwerpunkt bei den Details der Implementierung liegt.

### 3.1 Ziele der empirischen Studie

Das Studienziel hat sich im Laufe der Studie leicht verändert. War am Anfang die Idee vorhanden, dass ein OOP-First- gegenüber einem OOP-Later-Vorgehen mehr Vor- als Nachteile haben könnte (siehe ursprüngliche Thesen), ist spätestens bei der Hauptstudie die Idee eines reinen Entscheidungsexperiments (siehe Forschungsdesign im nächsten Kapitel 3.2) zum Tragen gekommen.

Die Ziele sind wie folgt:

- Es soll der Nachweis erbracht werden, dass die hier erfolgte Definition (siehe Kapitel 1.2.3) bzw. Modellbildung von OOP-First und OOP-Later im Unterricht Bestand hat (Nachweis: siehe Kapitel 7.5.1).
- Es sollen die Forschungsfragen beantwortet werden (Fragen: siehe Kapitel 2.4, Antworten: siehe Kapitel 7.5.2).
- Es sollen der Lernerfolg nach einem Schuljahr ermittelt und mögliche Unterschiede zwischen den Schulklassen diskutiert werden. Dazu gehört auch eine Reflexion über den Lernerfolg bezogen auf die einzelnen Themen (Auswertung: siehe Kapitel 6.3 und Kapitel 7.3).
- Es sollen das subjektive Erleben der Schülerinnen und Schüler ermittelt und mögliche Unterschiede zwischen den Schulklassen diskutiert werden (Auswertung: siehe Kapitel 6.4 und Kapitel 7.4).
- Es sollen die speziellen Unterschiede und speziellen Schwierigkeiten bzw. Stärken beider Vorgehensweisen (OOP-First und OOP-Later) herausgearbeitet werden (Auswertung: siehe Kapitel 6 und Kapitel 7).
- Es sollen Hinweise gegeben werden (wenn Schwierigkeiten erkannt bzw. benannt wurden), welche Änderungsmöglichkeiten es im unterrichtlichen Geschehen geben könnte um den Unterricht zu verbessern (Hinweise: siehe Kapitel 7.5).
- Es sollen die Fragestellungen auch auf spezielle Lerngruppen (z. B. leistungsschwache oder leistungsstarke Schülerinnen und Schüler) bezogen werden (Auswertung: siehe Kapitel 6.4.6).
- Es soll mit der Studie (bzw. den Ergebnissen der Studie) die Diskussion um objects-first vs. objects-later versachlicht werden.

Das letzte Ziel ist dabei eher ein Wunsch.

## 3.2 Forschungsdesign der empirischen Studie

Dieses Kapitel ist teilweise wörtlich von einem Beitrag zur INFOS 2009 [ES09c] übernommen. Forschungsdesign-Merkmale, die für die Studie zutreffen bzw. ausgewählt werden, sind kursiv gekennzeichnet.

Schnell, Hill und Esser [SHE08] betonen in ihrem Kapitel „Forschungsdesign und Untersuchungsformen“, dass sich die empirische Arbeit im Wesentlichen auf die Überprüfung von Hypothesen bezieht. Dazu müssten eine Reihe von Entscheidungen getroffen werden, wann, wo, wie und wie oft die beobachtbaren empirischen Indikatoren an welchen Objekten erfasst werden [SHE08, S. 211]. Sie schreiben:

„Die Gesamtheit dieser Entscheidungen bezeichnet man als Untersuchungsanordnung oder „Forschungsdesign“. Der (logische) Aufbau des Forschungsdesigns ist entscheidend für den Grad an Gewissheit, mit dem die Frage nach dem Zusammenhang zwischen zwei Ereignissen (Ursache und Wirkung; unabhängige und abhängige Variable) beantwortet werden kann.“ [SHE08, S. 211].

Das zugrunde liegende *Forschungsdesign* hat zum *Ziel* (vgl. auch letztes Kapitel), Zusammenhänge (*Kausalitäten*) zwischen Variablen aufzudecken.

Sie schreiben weiter:

„Ganz allgemein kann man Untersuchungsanordnungen, die der Überprüfung von Hypothesen dienen, Experimente nennen.“ [SHE08, S. 212].

Bei dem vorliegenden *Experiment* sollen also die Hypothesen überprüft werden (vgl. Kapitel 2.4.6), die aus den Forschungsfragen entwickelt wurden (siehe Kapitel 2.4):

- Erzielt das OOP-First-Vorgehen oder das OOP-Later-Vorgehen bessere Lernergebnisse (unterteilt in einzelne Themengebiete)?
- Ist das subjektive Erleben der Schüler positiver beim OOP-First-Vorgehen oder beim OOP-Later-Vorgehen (unterteilt in einzelne Items)?

Durch die *ungerichtete* Formulierung der *Hypothesen* (vgl. Kapitel 2.4) kann von einem *Entscheidungs-Experiment* gesprochen werden: „Ist das OOP-First-Vorgehen oder das OOP-Later-Vorgehen besser?“.

Bortz und Lienert [BL03] fordern für eine *hypothesenprüfende Untersuchung*, welches die Zielgrößen und Einflussgrößen sind:

„Aus dem Studienprotokoll einer hypothesenprüfenden Untersuchung muss deutlich werden, welche Zielgröße als sog. abhängige Variable [...] untersucht und welche Einflussgrößen als sog. unabhängige Variable [...] hierbei geprüft werden sollen.“ [BL03, S. 16].

Bei der Untersuchung ist die *unabhängige Variable* die fachdidaktische Vorgehensweise. Sie unterscheidet sich in den beiden Schulklassen (fast) nur durch die unterschiedliche Reihenfolge der (insgesamt neun) Themen.



Die *abhängige Variable* ist in diesem Fall der Lernerfolg bzw. das subjektive Erleben des Unterrichts durch die Schülerinnen und Schüler.

Experimente können in Labor- und Feldexperimente eingeteilt werden, Schnell, Hill und Esser [SHE08] nehmen dabei Bezug auf die *interne* und *externe Validität*:

„Echte Experimente führen in der Regel durch ihre Möglichkeiten der Kontrolle von Störfaktoren zu hoher interner Validität. Da in der Praxis der notwendige Einsatz der Kontrolltechniken zumeist mit einer Durchführung des Experiments im Labor einhergeht, kann aber die externe Validität leiden. [...] Hingegen wird dem Feldexperiment, das in einer für die Versuchspersonen gewohnten Umgebung stattfindet, eine tendenziell höhere externe Validität zugute gehalten ...“ [SHE08, S. 225-226].

Das (Entscheidungs-)Experiment ist somit ein *Feldexperiment*, wobei das „Feld Schule“ durchaus auch (im positiven Sinne) Züge eines Labors hat.

Das Experiment wird an einer *kleinen Gesamtheit* durchgeführt, in diesem Fall zwei Parallelklassen der Einführungsphase der gymnasialen Oberstufe (11. Klasse) des Beruflichen Gymnasiums am Oberstufenzentrum Informations- und Medizintechnik in Berlin.

Schnell, Hill und Esser [SHE08] schreiben:

„Will man experimentelle Resultate auf größere Personengruppen (Grundgesamtheiten) generalisieren, so sind spezielle Auswahlverfahren (insbesondere Zufallsauswahlen) zur Rekrutierung der Versuchspersonen unumgänglich.“ [SHE08, S. 228].

Dies ist in dieser Studie nicht der Fall. Die Schüler kommen von sich aus an das OSZ IMT, sie werden nicht speziell ausgewählt. Daher muss bei der Auswertung der Ergebnisse vorsichtig mit einer Generalisierung (auf alle Schüler Deutschlands) umgegangen werden.

Ausgewählt wird aber die Verteilung der vorhandenen Schüler auf die zu bildenden Klassen. Diese Verteilung geschieht über eine *geschichtete Stichprobenauswahl* bzw. *geschichtete Stichprobenerhebung*. Hierfür werden einzelne Schüler-Merkmale (wie z. B. Geschlecht, Alter, etc.) in homogene Subpopulationen, also in die beiden Schulklassen, aufgliedert [BL03, S. 24].

Auch Helmut Kromrey [Kro06] nennt in seinem Kapitel „Forschungsplanung und Forschungsdesign“ den Vorteil geschichteter Stichproben. Er mahnt aber, dass diese Schichtung nur als eine Art Vorgruppierung gesehen werden darf:

„Danach entscheidet ein Zufallsverfahren, welche Personen aus jeder Schicht der Experimentalgruppe und welche der Kontrollgruppe zugewiesen werden. Auf diese Weise erreicht man zweierlei: zum einen die Bildung unmittelbar äquivalenter Experimental- und Kontrollgruppen hinsichtlich der Schichtungsmerkmale, zum anderen durch das zufällige Zuweisen – das sog. Randomisieren – die Ausschaltung der Gefahr systematischer

Ergebnisverzerrungen durch Faktoren, die dem Forscher vorab nicht bekannt sind.“ [Kro06, S. 99, kursives Wort im Original].

Auch Bortz und Lienert [BL03] legen Wert auf die randomisierte Gruppenbildung. Sie sprechen von einem *kontrollierten Experiment*, da einer *Experimentalgruppe* (EG) eine *Kontrollgruppe* (KG) gegenübersteht [BL03, S. 18]. Ob nun die OOP-First-Klasse die Experimentalgruppe und die OOP-Later-Klasse die Kontrollgruppe ist, oder das Verhältnis genau umgekehrt ist, spielt keine Rolle. Die Studie ist *teilweise randomisiert*, da die Verteilung der Probanden (Schüler) auf EG und KG (fast) zufällig erfolgte, allerdings ist sie weder ein Blindversuch noch ein Doppelblindversuch, da sowohl die Schüler (Probanden) als auch die Lehrer (Versuchsleiter) wussten, ob sie zur KG oder zur EG gehörten [BL03, S. 19].

Laut Jacobs [Jac98] ist eine Untersuchung, ob eine Unterrichtsmethode I besser ist als eine Unterrichtsmethode II (verwendet in zwei Parallelklassen: Klasse A und Klasse B mit Vor- und Nachtests) ein *quasiexperimenteller Versuchsplan*.

Helmut Kromrey [Kro06] spricht auch von *Quasi-Experimenten*, z. B. wenn es keine „Vorher-Messung“ gibt:

„Untersuchungsanordnungen, die sich an der *Experimentallogik* orientieren, jedoch nicht alle Bedingungen des klassischen Experiments erfüllen können, werden als *Quasi-Experimente* bezeichnet.“ [Kro06, S. 102, kursive Schreibweise im Original].

Ein Vortest soll neben der Überprüfung der Ausgangslage auch die Gleichartigkeit der Gruppen kontrollieren. Es kann aber durchaus sinnvoll sein, auf den Vorgang des Messens vor dem eigentlichen „treatment“ zu verzichten. Helmut Kromrey [Kro06, S. 99] schreibt dazu, dass dadurch verhindert wird, dass allein schon durch die Messung Veränderungen in Gang gesetzt werden (Versuchskaninchen-Effekt). Bei der Studie wird daher die „Vorher-Messung“ (Vortest) an einer nicht am Experiment teilnehmenden Gruppe vorgenommen (siehe Kapitel 4.3.3).

Damit ist schon ein großer Teil der Forschungsplanung dargestellt worden. Jürgen Friedrichs [Fri90] schreibt in einer Zusammenfassung:

„Unter Forschungsplanung wurden zwei Gruppen von Arbeitsschritten verstanden: die Konzeptualisierung und die Festlegung des Untersuchungsplans. Die Konzeptualisierung richtete sich u. a. darauf, das Problem zu strukturieren, Variablen zu isolieren, Hypothesen zu formulieren und bestehende Theorien zur Klärung der Sachverhalte heranzuziehen. Hierzu gehört auch die Frage, welche Indizes, Skalen, Tests etc. entwickelt werden müssen ...“ [Fri90, S. 158].

Die Implementierung der Studie im Kontext zum Forschungsdesign (siehe Kapitel 3.3.2 und Abbildung 24) sieht sowohl drei Tests vor (*Vortest*, *Vergleichstest* und *Nachhaltigkeitstest* (siehe Kapitel 4)), als auch den Einsatz eines *Fragebogens* zum subjektiven Erleben, der noch entwickelt werden muss (siehe Kapitel 5).

Der gesamte *Untersuchungsablauf* wird in einer Vorstudie im Schuljahr 2006/2007 getestet.

Nach Nieschlag, Dichtl und Hörschgen [NDH02] werden bei dem *Untersuchungsablauf* idealtypisch *fünf Phasen* unterschieden:

- *Definitionsphase* (Problemformulierung),
- *Designphase* (Konzeptualisierung),
- *Feldphase* (Datenerhebung),
- *Analysephase* (Auswertung und Interpretation der Ergebnisse) und
- *Kommunikationsphase* (Transfer der Ergebnisse).

Diese Phasen finden sich sowohl in der Vorstudie als auch in der Hauptstudie im Schuljahr 2007/2008 wieder (siehe Untersuchungsplan in Kapitel 3.3.3 und Abbildung 25). Das nächste Kapitel ist z. B. Teil der Designphase.

### **3.3 Implementierung der empirischen Studie**

Bei der konkreten Implementierung muss das ausgewählte Forschungsdesign umgesetzt werden (siehe Kapitel 3.3.2). Dann werden mit Hilfe eines Untersuchungsplans die Meilensteine der Untersuchung, also die zeitlichen Abläufe der Studie, festgelegt (siehe Kapitel 3.3.3). Mit Hilfe vieler Entscheidungen bzw. Festlegungen werden die folgenden Fragen beantwortet:

- Welches sind die abhängigen und unabhängigen Variablen? Wie wird dafür gesorgt, dass die meisten Variablen möglichst unabhängig sind (siehe Kapitel 3.3.5)?
- Wie wird die Gleichartigkeit der Zusammensetzung beider Klassen gewährleistet (siehe Kapitel 3.3.6)?
- Welche Themen sollen unterrichtet werden, wie ist die geplante zeitliche Reihenfolge (siehe Kapitel 3.3.8)?
- Wie wird für gleichartigen Methoden- und Medieneinsatz gesorgt (siehe Kapitel 3.3.9)?
- Wie werden der Unterricht bzw. die Lehrer sinnvoll synchronisiert (siehe Kapitel 3.3.11)?
- Welche Tests für den objektiven Aspekt der Studie (im Kontext auch zur zeitlichen Dimension) sollen stattfinden (siehe Kapitel 3.3.12)?
- Wann und wie werden die Fragebögen zum subjektiven Aspekt der Studie benutzt (siehe Kapitel 3.3.13)?

Um diese Entscheidungen bzw. Festlegungen zu überprüfen wird erst eine Vorstudie (im Schuljahr 2006/2007) durchgeführt. Die dort gesammelten Erfahrungen sollen dann zu einer Verbesserung des Designs der Hauptstudie (im Schuljahr 2007/2008) führen (siehe Kapitel 3.3.14).

Schließlich verbleiben zwei Problemfelder (siehe Kapitel 3.3.15), die mit den folgenden Fragen einhergehen:

- Wie wird dafür gesorgt, dass die Wahl zweier unterschiedlicher Lehrer möglichst wenig Einfluss auf die Untersuchungsergebnisse hat?
- Kann ein Vortest geschrieben werden, ohne Einfluss auf die Untersuchung zu nehmen? Wenn nein, wie kann der Kenntnis-Stand der Schülerinnen und Schüler am Anfang der Untersuchung belegt werden?

Bei den folgenden Kapiteln wird bei der Implementierung der Schwerpunkt auf die Implementierung der Hauptstudie gelegt, die Erfahrungen mit der Vorstudie sind bereits eingearbeitet (siehe auch Kapitel 3.3.14).

### 3.3.1 Randbedingung: Der Unterricht am Beruflichen Gymnasium des OSZ Informations- und Medizintechnik

Das OSZ IMT ([www.oszimt.de](http://www.oszimt.de), geprüft: 03.05.2011) ist die größte IT-Schule Berlins (ca. 1000 Rechner, 30 Techniklabore, 160 Lehrer und 3000 Schüler) [Zor09].



Abbildung 21: Gebäudeansicht des Oberstufenzentrums Informations- und Medizintechnik in Berlin (<http://www.oszimt.de/0-schule/mitteilungen/05-06/2006-03-06.html>, Bild: schule.jpg, geprüft: 29.04.2010)

Dort unterrichtete ich ausschließlich Informatik, speziell Software-Engineering, Datenbanksysteme, Robotik und Programmierung (inkl. Zertifizierung). An der Schule bin ich der Leiter des Fachbereiches Informationstechnik (siehe auch: [www.ehlert.oszimt.de](http://www.ehlert.oszimt.de) (geprüft: 03.05.2011) und Lebenslauf im Anhang) mit ca. 20 Kolleginnen und Kollegen.



Abbildung 22: Der erste Abiturjahrgang am OSZ IMT im Jahr 2005 (<http://www.oszimt.de/1-bildung/allgemein/og/abi/abi2005.html>, Bild: abi2005.jpg, geprüft: 29.04.2010)

Die Schule hat eine kleine gymnasiale Oberstufe (Berufliches Gymnasium), die ab der 11. Klasse beginnt. In dieser so genannten Einführungsphase haben die Schüler das Fach Technik, welches zur Hälfte die „Einführung in die (objektorientierte) Programmierung“ vorsieht.



Abbildung 23: Einzel- und Partnerarbeit im Computerlabor am OSZ IMT (<http://www.oszimt.de/2-faecher/allgemein/og/og-ti-k3.html>, Bild: TI-K3-CD1.jpg, geprüft: 29.04.2010)

Die Vorstudie dieser empirischen Untersuchung wird mit zwei Parallelklassen (schulinterne Bezeichnung: OG61 und OG62) in der Einführungsphase im Schuljahr 2006/2007 durchgeführt, die Hauptstudie im Schuljahr 2007/2008 mit den Klassen OG71 und OG72.

Für diese empirische Studie gelten einige Randbedingungen, die überwiegend von dem Bundesland (Berlin), der konkreten Schule (OSZ IMT), dem Bildungsgang (Berufliches Gymnasium), der Klassenstufe (11), dem konkreten Unterrichtsfach (Technik) und den konkreten Schülern vorgegeben werden [ES09c, S. 126]:

- Es geht es um einen Informatik-Anfangsunterricht (nach Vorgaben des Stoffverteilungsplans bzw. Kernarbeitsplans der Schule (siehe Anhang B), damals noch mit Lernziel- statt Kompetenz-Formulierungen) in einer 11. Klasse (Einführungsphase) im Fach Technik des Beruflichen Gymnasiums des Oberstufenzentrums Informations- und Medizintechnik (OSZ IMT) in Berlin mit überwiegend männlichen Schülern.
- Die Schülerinnen und Schüler haben in der Regel keine Programmier-Erfahrungen und kommen überwiegend von Real- bzw. Gesamtschulen (MSA-Abschluss).
- Die Schülerinnen und Schüler sollen neben der Modellierung auch relativ schnell Programme in Java schreiben, der Fokus des Unterrichts auf die OO-Programmierung ist also vorgegeben. Die Informatik-Kolleginnen und -Kollegen haben vereinbart, dass der objektorientierte Einstieg mit dem BlueJ-Tool erfolgt.
- Die Schülerinnen und Schüler sollen in einem Schuljahr (à 3h / Woche) von Datentypen, Steuerstrukturen bis hin zur Vererbung in der OOP im Rahmen einer „Einführung in die (objektorientierte) Programmierung“ unterschiedliche Themen durchdringen. Dazu steht zu jeder Zeit ein Computerraum zur Verfügung.

Die Kolleginnen und Kollegen sind sich aber uneins, in welcher zeitlichen Reihenfolge die Themen dabei unterrichtet werden sollen, ob also der OOP-First-Weg oder der OOP-Later-Weg beschritten werden soll. Einig ist man sich aber, dass sich die Schüler so oder so schwer tun bei der „Einführung in die Programmierung“. Ob dies überwiegend ein Problem von objektorientierten Inhalten oder ein allgemeines Problem ist, bleibt dabei aber strittig und die Kolleginnen und Kollegen sind sehr interessiert an den Ergebnissen dieser empirischen Studie.

### 3.3.2 Implementierung in Bezug auf das Forschungsdesign

Die erste Frage des Entscheidungsexperiments lautet:

„Erzielt das OOP-First-Vorgehen oder das OOP-Later-Vorgehen bessere Lernergebnisse (unterteilt in einzelne Themengebiete)?“

Aus dieser Frage wird ersichtlich, dass es verschiedene Tests geben muss, welche das Erreichen von Lernzielen (siehe Kapitel 4.1) abprüfen. Im Kapitel 3.3.12 wird begründet, warum ein Vortest, ein Vergleichstest und ein Nachhaltigkeitstest geschrieben werden und warum folgende zeitliche Festlegungen vorgenommen werden (siehe Abbildung 24):

- Der *Vortest* (Pre-Test) zur Ermittlung des Ursprungswissens der Schülerinnen und Schüler soll vor dem Schuljahr 2007/2008, also im Sommer 2007 erfolgen. Warum es aber zu einer indirekten Messung des Vorwissens durch einen Test im Sommer 2008 gekommen ist, erläutert das Kapitel 3.3.15.
- Der so genannte *Vergleichstest* (Post-Test) wird am Ende des Schuljahres 2007/2008, also im Sommer 2008 geschrieben.

- Der *Nachhaltigkeitstest* (Follow-Up-Test) wird nach den Sommerferien 2008, also im Spätsommer 2008 geschrieben.

Die zweite Frage des Entscheidungsexperiments lautet:

„Ist das subjektive Erleben der Schüler positiver beim OOP-First-Vorgehen oder beim OOP-Later-Vorgehen (unterteilt in einzelne Items)?“

Aus dieser Frage wird ersichtlich, dass ein *Fragebogen* entwickelt werden muss (siehe Kapitel 5), der die subjektiven Aspekte des Unterrichtsgeschehens abfragt. Hier wird im Kapitel 3.3.13 die folgende zeitliche Festlegung vorgenommen (siehe Abbildung 24): Nach jedem der einzelnen Themen (also neunmal im Schuljahr) wird das Stimmungsbild der Schülerinnen und Schüler erhoben.

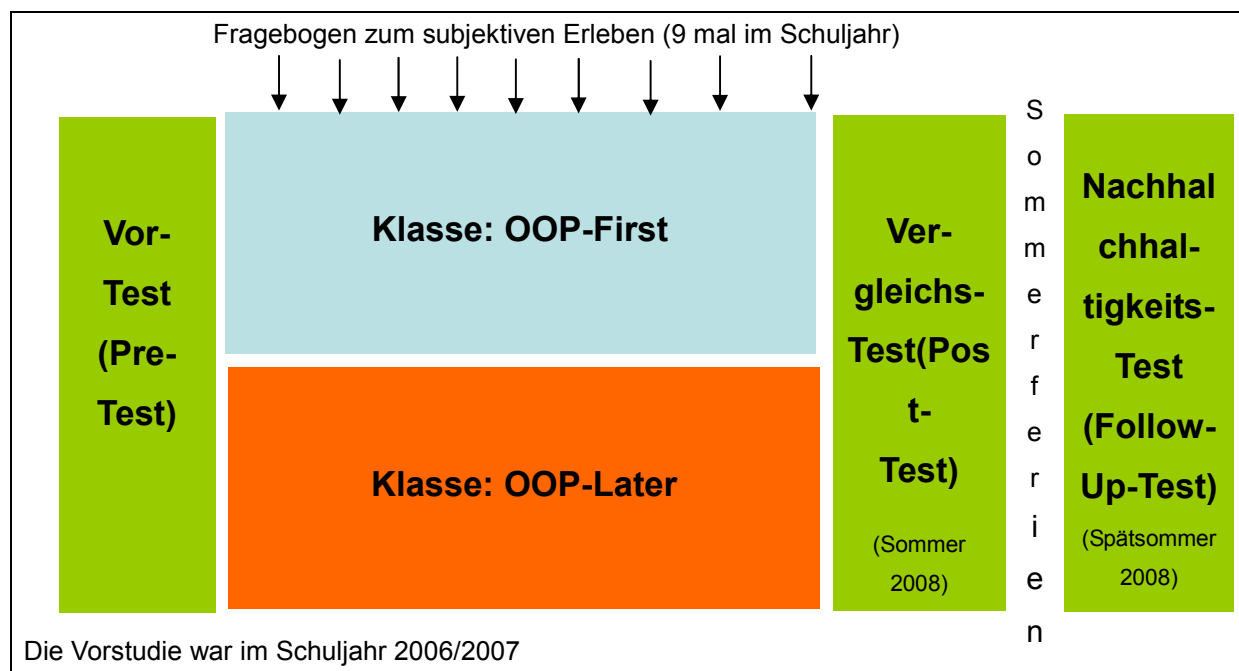


Abbildung 24: Design der Hauptstudie (vgl. z. B. [ES09c, Abb. 3, S. 126])

Bezug nehmend auf den konkreten Unterricht gibt es die folgenden Festlegungen (vgl. [ES09c, S. 126], [ES10a, S. 108-109]):

- In zwei Schulklassen wird ein Schuljahr lang (ca. 40 Unterrichts-Wochen, 3h die Woche) das Thema „Einführung in die (objektorientierte) Programmierung“ unterrichtet. Dazu gehören die Programmier-Grundlagen von Java genauso wie eine Einführung in das objektorientierte Paradigma. Der Hauptfokus liegt dabei eher auf der Programmierung als auf der Modellierung.
- Die Zusammensetzung der beiden Klassen ist gleichartig, was das Alter der Schülerinnen und Schüler, das Geschlecht, die Vornote in Mathematik und Deutsch, den Schulzweig, aus dem sie kommen und ihre Nationalität anbelangt (siehe Kapitel 3.3.6). Allerdings ist der Informatiklehrer nicht derselbe (siehe Kapitel 3.3.10, 3.3.11 und 3.3.15).
- Beide Klassen erhalten im Laufe eines Schuljahres neun gleichartige Themen unterrichtet (siehe Kapitel 3.3.8), aber die zeitliche Reihenfolge der Themen ist unterschiedlich: Daher werden sie als OOP-First-Klasse und OOP-Later-Klasse gekennzeichnet.

- Der Fokus während der unterschiedlichen fachdidaktischen Vorgehensweisen in den beiden Klassen liegt zum einen auf einem subjektiven Aspekt (mit einem speziellen Fragebogen) und zum anderen auf einem objektiven Aspekt (mit drei verschiedenen Tests) des Lerngeschehens.

Um die Implementierung zu überprüfen wird eine Vorstudie im Schuljahr 2006/2007 durchgeführt. Die Hauptstudie schließt sich im Schuljahr 2007/2008 an. Die Abbildung 24 zeigt im Überblick das Design der Hauptstudie, welches in den folgenden Kapiteln weiter entwickelt wird.

### 3.3.3 Zeitliche Dimension der empirischen Studie (Untersuchungsplan)

Das zeitliche Vorgehen der Studie (siehe Abbildung 25) hält sich an die Phasen des Untersuchungsplans (vgl. Kapitel 3.2) nach Nieschlag, Dichtl und Hörschgen [NDH02].

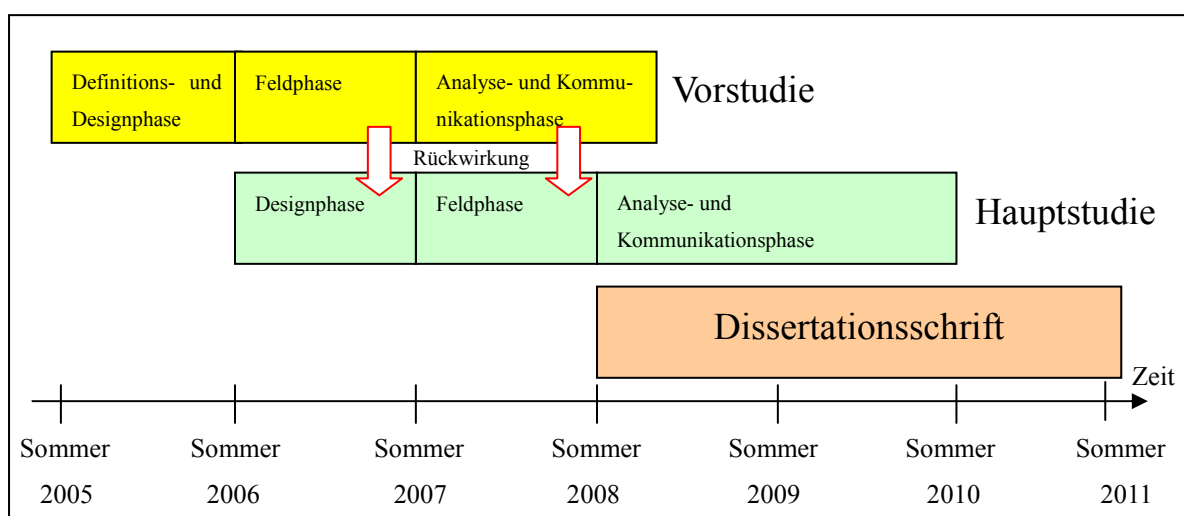


Abbildung 25: Phasendarstellung der Vorstudie und der Hauptstudie (vgl. auch [NDH02, Abb. 55])

Rückblickend kann man diesen Phasen konkrete Zeitdaten zuordnen:

Seit dem Sommer 2005 habe ich Gedanken und Probleme formuliert und bin deshalb auch zu 4 Fortbildungen gegangen (*Definitionsphase*). Über den daraus resultierenden Kontakt und die Diskussion mit Prof. Carsten Schulte wurde die Idee zu dieser Studie im Januar 2006 geboren. Somit konnte bis zum Sommer 2006 ein erstes Konzept für die Vorstudie erstellt werden (*Designphase*). Während die Datenerhebung der Vorstudie ein Jahr lang lief, konnte parallel dazu das Konzept für die Hauptstudie erstellt werden, wobei die *Feldphase* der Vorstudie Einfluss hatte auf die *Designphase* der Hauptstudie. Auch bei der Auswertung und Interpretation der Vorstudien-Daten (*Analysephase* der Vorstudie) wurde schnell klar, dass diese *Analysephase* Einfluss auf die parallel laufende *Feldphase* der Hauptstudie nimmt. Z. B. wurde auf eine verstärkte Lehrersynchronisation geachtet und der Vergleichs- bzw. Nachhaltigkeitstests vollständig überarbeitet (siehe Kapitel 3.3.11 und 3.3.12).

Gleichzeitig wurden die Ergebnisse der Vorstudie transferiert und veröffentlicht (*Kommunikationsphase*). Seit dem Ende der *Feldphase* der Hauptstudie im Spätsommer 2008 war der Untersuchungsplan der Vorstudie beendet, es konnten die neuen Ergebnisse ausgewertet, interpretiert, transferiert und veröffentlicht werden (*Analyse- und Kommunikationsphase* der Hauptstudie). Zur gleichen Zeit (Juli 2008) wurde die Dissertationsschrift begonnen und im Sommer 2011 beendet.

### 3.3.4 Kommunikationsphase: Veröffentlichungen im Kontext der Studie

Passend zum Untersuchungsplan bzw. zum zeitlichen Ablauf der Studie wurden von mir (teilweise unter Mitwirkung von Prof. Schulte) Beiträge veröffentlicht bzw. Vorträge gehalten. Diese Beiträge spiegeln gut den zeitlichen Verlauf meiner Arbeit wieder. Im Folgenden unterscheidet sich zwischen Vorträgen (auf Konferenzen bzw. auf Doktorandenkolloquien) und Veröffentlichungen.

1. Folgende Vorträge (oft mit einem entsprechenden Papier bzw. Konferenzbeitrag) wurden gehalten:
  - 5. Berliner MNU-Kongress 2006: *Objektorientierte Programmierung im Anfängerunterricht nicht(!) von Anfang an?! [Ehl06]*
  - 7. GI-Tagung der Fachgruppe "Informatik-Bildung in Berlin und Brandenburg" 2007 in Potsdam: *Objects Really First? [Ehl07c]*
  - 11th Workshop TLOOC at European Conference on Object-Oriented Programming (ECOOP) 2007 in Berlin: *Learners Views on Objects-First and Objects-Later - Results of an Exploratory Study [ES07b]*
  - 6. Berliner MNU-Kongress 2008: *OOP first oder OOP later - Ergebnisse der Hauptstudie [Ehl08b]*
  - International Computing Education Research Workshop (ICER) 2009 in Berkeley, Kalifornien, USA: *Empirical Comparison of Objects-First and Objects-Later [ES09b]*
  - 13. GI-Fachtagung „Informatik und Schule - INFOS 2009“ in Berlin: *Unterschiede im Lernerfolg von Schülerinnen und Schülern in Abhängigkeit von der zeitlichen Reihenfolge der Themen (OOP-First bzw. OOP-Later) [ES09d]*
  - 15th Annual Conference on Innovation and Technology in Computer Science Education 2010 (ITiSE'10) in Bilkent, Ankara, Türkei: *Empirical Comparison of Objects-First and Objects-Later*, gehalten von Prof. Schulte am 29.06.2010 [ES10b]
2. Veröffentlichungen findet man unter (siehe auch Literaturverzeichnis (Quellenangabe)):
  - Ehlert, A. und Schulte, C.: *Learners Views on Objects-First and Objects-Later - Results of an Exploratory Study*, ECOOP 2007 in Berlin [ES07a]
  - Ehlert, Albrecht: *Studie: Objects-First- und Objects-Later-Einstieg*, Praxisband der 12. GI-Fachtagung "Informatik und Schule - INFOS 2007" in Siegen [Ehl07b]
  - Ehlert, A. und Schulte, C.: *Empirical comparison of objects first and objects later*, ICER'09 in Berkeley, Kalifornien, USA [ES09a]
  - Ehlert, A. und Schulte, C.: *Unterschiede im Lernerfolg von Schülerinnen und Schülern in Abhängigkeit von der zeitlichen Reihenfolge der Themen (OOP-First bzw. OOP-Later)*, Tagungsband der INFOS 2009 in Berlin [ES09c]
  - Ehlert, A. und Schulte, C.: *Comparison of OOP First and OOP Later – First Results Regarding the Role of Comfort Level*, ITiCSE '10 in Bilkent, Ankara, Türkei [ES10a]

Sehr hilfreich für mich waren die Diskussionen im Internationalen Doktorandenkolloquium. Ausgehend vom Doktorandenforum der INFOS 2005 in Dresden fand das Internationale Doktorandenkolloquium der Didaktik



der Informatik (IDDI) seit dem Jahr 2006 im halbjährlichen Rhythmus statt. Es verbindet allgemeine mit individuellen Fragestellungen zum Ablauf von Dissertationsprojekten und in den Kurzvorträgen werden neue Forschungserkenntnisse aus den Promotionsvorhaben der Teilnehmerinnen und Teilnehmer vorgestellt und im Kolloquium diskutiert [Ste07, Vorwort, S. 3].

3. Teilnahme an den folgenden Doktoranden-Kolloquien (in der Regel mit einem eigenen Beitrag):

- 1. Internationales Doktorandenkolloquium der Didaktik der Informatik, FU Berlin, März 2006: *Erste Thesen*
- 3. Internationales Doktorandenkolloquium der Didaktik der Informatik, ETH Zürich, März 2007: *Vorstellung des Studiendesigns*
- 4. Internationales Doktorandenkolloquium der Didaktik der Informatik, INFOS, Universität Siegen, September 2007: *Objects-First- und Objects-Later-Einstieg*
- 5. Internationales Doktorandenkolloquium der Didaktik der Informatik, Universität Münster, Februar 2008: *Kurz-Vorstellung des Inhaltsverzeichnisses*
- 6. Internationales Doktorandenkolloquium der Didaktik der Informatik, DDI, Universität Erlangen, September 2008: *Problem mit dem Post-Test*
- 8. Internationales Doktorandenkolloquium der Didaktik der Informatik, INFOS, Freie Universität Berlin, September 2009, der Vortrag erfolgte auf der INFOS selber (s. o. und [ES09d]).

### 3.3.5 Variablen der empirischen Studie

Sachs und Hedderich klassifizieren in ihrem Buch „Angewandte Statistik“ [SH09, S. 15-20] einerseits Variablen, die Einfluss auf die Studie bzw. Datenerhebung ausüben (Störvariablen und unabhängige Variablen) und andererseits Variablen, die beeinflusst werden (abhängige Variablen). Die *unabhängige Variable*, die gut kontrollierbar ist, ist bei der Problemstellung der Studie die fachdidaktische Vorgehensweise. Das OOP-First- bzw. OOP-Later-Vorgehen ist somit Ursache für die *abhängige Variable*, in diesem Fall also der Lernerfolg bzw. das subjektive Erleben (Wirkung). Die dazwischen liegenden Variablen (Unterrichtsmethodik, Medien, Lehrer, Computereinsatz etc.) müssen insofern kontrolliert werden (also gleichartig gestaltet werden), dass sie möglichst wenig (am besten: keinen) bzw. gleichartigen Einfluss auf die abhängige Variable nehmen. Sachs und Hedderich [SH09, S. 15-17] definieren so genannte Störvariablen, die nicht gut kontrollierbar sind. Bei den oben genannten Variablen ist m. E. der Lehrer am wenigsten gut kontrollierbar. Er muss daher als „Störgröße“ gesehen werden, die unbedingt soweit wie möglich kontrolliert werden muss.

In den nächsten Kapiteln werden die folgenden Fragen im Hinblick auf die Variablen diskutiert bzw. beantwortet:

- Wie wird dafür gesorgt, dass die Themen gleichartig sind (siehe Kapitel 3.3.8)? Ziel ist es, dass sich diese Einflussvariable nur in der zeitlichen Abfolge der Einflussnahme unterscheidet und nur dadurch unterschiedlichen Einfluss auf die abhängige Variable nimmt.

- Wie wird dafür gesorgt, dass die beiden unterschiedlichen Lehrer möglichst gleichartig sind bzw. gleichartig im Unterricht vorgehen (siehe Kapitel 3.3.11)? Ziel ist es, dass diese Störvariable gleichartigen Einfluss auf die abhängige Variable nimmt.
- Wie wird dafür gesorgt, dass die Methodik, die Medien, die Arbeitsmittel etc. gleichartig sind (siehe Kapitel 3.3.9)? Ziel ist es, dass diese Variablen gleichartigen Einfluss auf die abhängige Variable nehmen.

Bevor auf die Sicherstellung der Gleichartigkeit der Einfluss nehmenden Variablen eingegangen wird, muss zuerst dafür gesorgt werden, dass die Schülerverteilung der beiden Klassen gleichartig ist.

### 3.3.6 Verteilung der Schüler auf die OOP-First- bzw. OOP-Later-Klasse

Ziel beim Setzen der Variablen ist es, möglichst viele Einflussvariablen (bis auf die unabhängige Variable) gleichartig zu gestalten. Dies bedeutet für die beiden Schulklassen:

- Die Mädchen-Jungen-Verteilung sollte gleich sein.
- Das Durchschnittsalter sollte gleich sein.
- Der Migrantenanteil sollte gleich sein.
- Das Mischungsverhältnis im Kontext zu den vorhergehenden Schultypen (Realschule, Gesamtschule, Gymnasium) sollte gleichartig sein.
- Die letzte Mathematik-Note sollte im Durchschnitt gleich sein. Speziell diese Note wird als guter Indikator für einen Informatik-Lernerfolg gesehen (vgl. z. B. [WS01] und [BR05]).

Alle Schüler, die von ihren bisherigen Schulen auf die gymnasiale Oberstufe des Beruflichen Gymnasiums des OSZ IMT wechseln wollen, müssen bei der Anmeldung nicht nur ihre Standarddaten wie Name, Vorname, Geburtsdatum etc. angeben, sondern auch ihr letztes Zeugnis vorlegen. So können für die Zusammensetzung der Klassen Daten im Hinblick auf das Alter, das Geschlecht, den vorherigen Schulzweig, die Nationalität, die Zensuren etc. abgefragt werden (siehe Anhang I). Nun werden (z. B. mit Hilfe von EXCEL) die Schüler erst „geschichtet“ und dann so „randomisiert“ den beiden Klassen zugeordnet (vgl. Kapitel 3.2), dass sich eine gleichartige Zusammensetzung ergibt (siehe Abbildung 26).

Daten	m	w	Geburts-jahr	Math.-Note	Deutsch-Note	National.: Deutsch	Andere National.	Gym-nasium	Real-schule	Gesamt-schule	Wieder-holer
OG71	20	2	1989,45	2,64	2,64	22	0	5	12	3	2
OG72	20	2	1989,68	2,64	2,82	21	1	5	11	5	1

Abbildung 26: Klassenstatistik für die Hauptstudie 2007/2008 (generiert aus den Daten aus Anhang I)

Dabei ist zufällig die OG71 die OOP-First-Klasse und die OG72 die OOP-Later-Klasse. Welche der beiden Klassen die Experimentalklasse und welche die Kontrollklasse ist, kann dabei offen bleiben.

Es kann die Gleichartigkeit der Klassen wie folgt begründet werden:

- Beide Klassen bestehen aus zwei Schülerinnen und 20 Schülern.

- Die Schülerinnen und Schüler beider Klassen sind im Durchschnitt (im Sommer 2007) 18 Jahre alt.
- Die Mathematik-Durchschnittsnote ist dieselbe, die Deutsch-Durchschnittsnote differiert nur um 0,18 Notenpunkte.
- Von den 44 Schülern sind 43 deutscher Nationalität.
- Das Verhältnis von den Schularten (Gymnasien / Realschule / Gesamtschule / „OSZ-Wiederholer“) kann man m. E. mit 5/12/3/2 contra 5/11/5/1 als ähnlich betrachten.

Ob dies zu einer Gleichartigkeit auch im Hinblick auf die Computervorkenntnisse geführt hat, soll über eine Eingangsbefragung überprüft werden (siehe Anhang J). Dabei ergaben sich (für die Klassen der Hauptstudie) die folgenden prozentualen Ergebnisse:

- Fast alle Schüler verfügen über einen eigenen Computer (OG71: 93% / OG72: 96%).
- Fast alle Schüler verfügen über einen Internetanschluss (OG71: 96% / OG72: 100%).
- Eine Spielekonsole hat eine Mehrzahl der Schülerinnen und Schüler (OG71: 68% / OG72: 58%).
- Die meisten Schüler hatten schon so genannten „Informatikunterricht“ (OG71: 79% / OG72: 83%).

Bei den nächsten Fragen konnte immer einer von fünf Bereichen von „sehr viel/sehr fundiert“ (Zahl 1) bis „sehr wenig/keine“ (Zahl 5) angekreuzt werden. Die folgenden Durchschnittszahlen beruhen auf der Entscheidung, dass den fünf Bereichen die Zahlen von 1 bis 5 zugeordnet werden.

- Downloads aus dem Internet geschehen in beiden Klassen etwas mehr als durchschnittlich (OG71: 2,8 / OG72: 2,5).
- Die Schüler recherchieren viel im Internet (OG71: 2,0 / OG72: 2,0).
- Sie verbringen eher durchschnittlich viel Zeit mit der Spielekonsole (OG71: 2,9 / OG72: 2,7).
- Ihre Textverarbeitungs-Kenntnisse sind durchschnittlich bis fundiert (OG71: 2,5 / OG72: 2,5).
- Ihre Kalkulationsprogramm-Kenntnisse sind nur durchschnittlich ausgeprägt (OG71: 3,6 / OG72: 3,0).
- Ihre Präsentationsprogramm-Kenntnisse sind guter Durchschnitt (OG71: 2,9 / OG72: 2,6).
- Sie haben fast keine Datenbankprogramm-Kenntnisse (OG71: 4,6 / OG72: 4,3).
- Programmier-Kenntnisse sind sehr selten anzutreffen (OG71: 4,3 / OG72: 4,2).
- Webseiten-Programmier-Kenntnisse sind selten vorhanden (OG71: 3,9 / OG72: 4,0), aber ein bisschen häufiger anzutreffen als Programmier-Kenntnisse.
- Die Schülerinnen und Schüler freuen sich auf den Informatikunterricht (OG71: 1,6 / OG72: 1,4).

Frage	1	2	3	4	5	6	7	8	9	10	11	12	13	14
OG71	93/7/0	96/4	68/32	79/21	2,8	2,0	2,9	2,5	3,6	2,9	4,6	4,3	3,9	1,6
OG72	96/4/0	100/0	58/42	83/17	2,5	2,0	2,7	2,5	3,0	2,6	4,3	4,2	4,0	1,4

Abbildung 27: Ergebnisse im Hinblick auf die Computervorkenntnisse (siehe Fragebogen in Anhang J)

Diese Zahlen geben keinen Anlass, etwas an der Schülerverteilung in den beiden Klassen zu ändern.

Eine Sicherung wird aber trotzdem eingebaut. Um sich nicht nur auf die Mathematik-Noten anderer Schulen (also den Schulen, die zuletzt von den Schülerinnen und Schülern besucht wurden) zu verlassen, sondern auch den aktuellen Mathematik-Leistungsstand zu ermitteln, wird in beiden Klassen in der Hauptstudie der selbe Mathematik-Lehrer eingesetzt. Seine Mathematik-Durchschnittsnoten für beide Klassen geben dann einen Hinweis auf die mathematische Leistungsfähigkeit der Klassen. Und diese soll (z. B. nach Wilson und Shrock [WS01]) korrelieren mit der informatischen Leistungsfähigkeit. Im Endeffekt geht es also um den Nachweis der Gleichartigkeit der beiden Klassen auch im Hinblick auf die mathematisch-informatischen Fähigkeiten.

Obwohl die meisten Schülerinnen und Schüler angeben, schon Informatikunterricht gehabt zu haben, war anscheinend das Thema Programmieren nicht Teil dieses Unterrichts (siehe Programmier-Kenntnisse). Auf mündliche Nachfrage wurde dies in beiden Klassen bestätigt. Der Informatikunterricht war oft bezogen auf das Erlernen bzw. Bedienen von Office-Anwendungen. Das ist insofern von Bedeutung, da diese Arbeit ja von einem Anfangsunterricht (im Kontext zur Programmierung) ausgeht und damit diese Eingangs-Voraussetzung bestätigt wird.

Positiv ist es, dass sich so viele Schülerinnen und Schüler auf den Informatikunterricht freuen.

### 3.3.7 Pseudonymisierung der einzelnen Schülerinnen und Schüler

Im Bundesdatenschutzgesetz [BDSG09] wird unter §3 („Weitere Begriffsbestimmungen“) im Abschnitt 6 die Anonymisierung von personenbezogenen Daten definiert:

„Anonymisieren ist das Verändern personenbezogener Daten derart, dass die Einzelangaben über persönliche oder sachliche Verhältnisse nicht mehr oder nur mit einem unverhältnismäßig großen Aufwand an Zeit, Kosten und Arbeitskraft einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können.“ [BDSG09, §3(6)].

Bei den Fragebögen und bei der Auswertung der Daten soll überprüft werden, ob es Zusammenhänge zwischen einzelnen Merkmalen der Schüler (z. B. Geschlecht) und den Daten der Fragebögen oder den Ergebnissen der Tests gibt. Daher kommt eine Anonymisierung der personenbezogenen Daten für die Untersuchung nicht in Frage.

Es wird die so genannte „Pseudonymisierung“ angewendet, die wie folgt definiert ist:

„Pseudonymisieren ist das Ersetzen des Namens und anderer Identifikationsmerkmale durch ein Kennzeichen zu dem Zweck, die Bestimmung des Betroffenen auszuschließen oder wesentlich zu erschweren.“ [BDSG09, §3(6a)].

Damit ist nach außen eine fast vollständige Anonymisierung erreicht, nach innen können aber die Daten vertieft ausgewertet werden. Die Schüler müssen natürlich über diesen Sachverhalt informiert werden und bei sensiblen Fragen die Möglichkeit erhalten, auch gar nicht zu antworten.

Das Ersetzen des Namens geschieht wie in Abbildung 28 dargestellt.

	Code
<b>Erster</b> Buchstabe des Vornamens der <b>Mutter</b>	
<b>Zweiter</b> Buchstabe des Vornamens der <b>Mutter</b>	
<b>Erster</b> Buchstabe des Vornamens des <b>Vaters</b>	
<b>Zweiter</b> Buchstabe des Vornamens des <b>Vaters</b>	

Abbildung 28: Pseudonymisierung des Schülernamens

Ein Beispiel: Der Schüler Karl Lange mit seiner Mutter Sonja und seinem Vater Helmut bekommt also den Code „SOHE“. Von Vorteil ist, dass nicht der Versuchsleiter einen Code vorgibt (den die Schüler schnell vergessen können), sondern dass die Schüler den Namen selbst bestimmen und somit jederzeit wieder herleiten können. Der Lehrer als Versuchsleiter muss allerdings die Schüler einmal nach ihrem Code fragen und die Zuordnung notieren. Damit wird den Schülern auch bewusst, dass sie nicht anonym an der Studie teilnehmen.

### 3.3.8 Bestimmung der Themen und ihrer zeitlichen Reihenfolge

Im Kapitel 1.2.2 wurden sieben gleichartige Themen herausgearbeitet (vgl. auch [ES09c, S. 124]), die in der Regel in Büchern für Programmier-Anfänger aufgeführt sind:

- A. Einführung in die OOP (Klasse und Objekt)
- B. Variablen (bzw. Attribute), Konstanten und fundamentale Datentypen
- C. Steuerstrukturen: Sequenz, Iteration und Selektion
- D. Prozeduren (bzw. Operationen, Methoden, Funktionen, Botschaften)
- E. Komplexe(re) Datentypen
- F. Vererbung
- G. Assoziation

Diese Themen der Fachliteratur sollen jetzt in Abhängigkeit von schulischen Randbedingungen und fachdidaktischen Entscheidungen in Unterrichtsthemen überführt werden. Die Themen werden dahingehend diskutiert, welche inhaltliche Dimension sie bei einem OOP-First- und bei einem OOP-Later-Vorgehen (auch im Kontext zu schulischen Randbedingungen) besitzen. Fertig herausgearbeitete Unterrichts-Themen sind in diesem Kapitel kursiv gekennzeichnet und mit einem Kürzel versehen (T1, T2 etc.). Im Kapitel 4.1 werden dann für die einzelnen Themen die Lernziele formuliert.

Das Thema A wird für den Unterricht in zwei Themen aufgeteilt. Dabei wird u. a. eine Randbedingung des OSZ IMT berücksichtigt, dass nämlich die Einführung in die OOP mit dem BlueJ-Tool erfolgt. Der erste Teil des Themas A wird also wie folgt definiert:

*Thema T1: Einführung in die OOP (mit BlueJ)*

Hier werden die Begriffe Klasse, Objekt, Attribut und Methode zum ersten Mal erwähnt. Leipholz-Schuhmacher [Lei04] hat schon 2004 berichtet, dass die Bedienung der BlueJ-IDE selbst für mit dem Rechner unerfahrene Schülerinnen und Schüler einfach ist und durch die Übung „Figures“ die Lerngruppe schnell mit dem Unterschied zwischen Klasse und Objekt vertraut wird. Andererseits sind in dieser Übung m. E. eigentlich drei mächtige Einzelthemen enthalten, nämlich Klasse bzw. Objekt (später T3), Attribute (später T2) und Methoden (später T4), so dass die einzelnen Themen teilweise nur oberflächlich behandelt werden (im Sinne eines Spiralmodells).

Wie erfolgt aber bei einem OOP-Later-Vorgehen der Einstieg in die Programmierung? Oft wird mit Hilfe einer textorientierten IDE das „Hallo-Welt-Programm“ vorgeführt und dann relativ schnell auf Variablen, Konstanten und ihre Datentypen eingegangen. Es werden Wertzuweisungen und evtl. schon Ein- und Ausgaben getätigt, z. B. nach dem so genannten EVA-Prinzip. Die dahinterliegende Steuerstruktur Sequenz kann benannt werden und evtl. auch schon mit Hilfe eines Struktogrammes (nach DIN 66261) dargestellt werden:

T2-OOP-Later: Einführung in den Java-Editor, Variablen, fundamentale Datentypen, Steuerstruktur Sequenz (inkl. Struktogramm)

Soll dies auch ein eigenständiges Thema bei dem OOP-First-Vorgehen sein, liegt der Schwerpunkt dann auf dem Umstieg vom BlueJ-Tool auf die IDE und auf dem Thema Attribute, die z. B. mit Hilfe des Konstruktors initialisiert werden. Die Steuerstruktur Sequenz hätte hier einen anderen Kontext, zum Beispiel wäre sie hier eine Abfolge von Methodenaufrufen auf einem Objekt:

T2-OOP-First: Einführung in den Java-Editor, Attribute, fundamentale Datentypen, Steuerstruktur Sequenz (inkl. Struktogramm)

Spätestens an dieser Stelle wird klar, warum von dem ursprünglichen Begriff der Module Abstand genommen wurde. Module sind inhaltlich und zeitlich abgeschlossen, sie laufen also exakt gleich ab. Aufgrund der zeitlich verschiedenen Abfolge der einzelnen Themen, können aber die Lerneinheiten nicht exakt gleich sein, weil ja das jeweilige Vorwissen ein anderes ist. Deshalb wurde vom Begriff des Moduls auf den Begriff des (gleichartigen) Themas umgestiegen.

Thema B muss also für beide Vorgehensweisen wie folgt formuliert werden:

*Thema T2: Einführung in den Java-Editor, Variablen bzw. Attribute, fundamentale Datentypen, Steuerstruktur Sequenz (inkl. Struktogramm)*

Als nächste Themen bei einer imperativ-prozeduralen Vorgehensweise kommen jetzt in fast allen Lehrbüchern die beiden Themen „Steuerstruktur Selektion“ und „Steuerstruktur Iteration“. Das o. g. Thema C wird also in zwei Themen aufgeteilt. Auch hier wird bei den beiden unterschiedlichen Vorgehensweisen der Fokus ein anderer sein. Die OOP-Later-Klasse wird in der Main-Methode kleine algorithmische Probleme lösen, die OOP-First-Klasse wird die Logik wahrscheinlich in die Methoden bzw. im Fall der Selektion vielleicht auch in die „Set-Verwaltungsmethoden“ schreiben:

*Thema T5: Steuerstruktur Selektion (inkl. Struktogramm)*

*Thema T6: Steuerstruktur Iteration (inkl. Struktogramm)*

Bei der OOP-First-Klasse wird der Begriff der Methode (Thema D) von Beginn an verwendet, er kann vor den Steuerstrukturen unterrichtet werden. Bei der OOP-Later-Klasse wird dieses Thema in der Regel erst nach den Steuerstrukturen unterrichtet. In Java wäre es z. B. denkbar in der „Main-Klasse“ (mit der main-Methode) weitere statische Methoden zu implementieren und so beispielhaft einmal eine „Eingabe-Verarbeitungs-Ausgabe-Sequenz“ zu realisieren. Dann wäre das Thema Methoden eigentlich erst einmal das Thema Prozeduren. Oder man wartet mit dem Thema bis nach dem OOP-Einstieg. Dann kann gleich der Methoden-Begriff eingeführt werden:

*Thema T4: Methoden*

Was jetzt noch fehlt ist eine Vertiefung des Begriffs „Klasse und Objekt“ (ein Teil des o. g. Themas A). Es soll eine Trennung von Klassen-Implementierung (Datei, Fachkonzept) und der Objekterzeugung in einem Anwendungsprogramm (Main-Methode, Benutzungsoberfläche) erfolgen:

*Thema T3: Klassenimplementierung (im Fachkonzept) und Objekterzeugung im Anwendungsprogramm (Benutzungsoberfläche)*

Auf der Ebene der so genannten OOP-Themen verbleiben jetzt noch die Themen „Vererbung (Thema F)“ und „Assoziation (Thema G)“, bei den so genannten prozeduralen bzw. traditionellen Themen fehlen zu den fundamentalen Datentypen (in Java sind dies 8 Datentypen) noch weitere Datentypen. Um die Schülerinnen und Schüler nicht zu überfordern, wird bei dem Thema „komplexe(re) Datentypen (Thema E)“ sich auf den Datentyp Feld beschränkt, inklusive des „Zeichen-Feldes“ (Char-Array), was in Java eher als Klasse String realisiert wird:

*Thema T7: Komplexere Datentypen (Arrays und Strings)*

Hier gibt es einen kleinen Widerspruch, wenn dieses Thema bei der OOP-Later-Klasse noch vor der Einführung in das objektorientierte Paradigma unterrichtet wird. Der Datentyp String ist in Java eine Klasse. Klassen wurden noch nicht behandelt. Die beste Möglichkeit ist m. E., dass man einfach von einem speziellen Datentypen spricht, der Methodenaufrufe bereit hält (siehe Informations- und Arbeitsblatt in [Ehl07a, S. 4]).

Verbleiben nur noch das Thema „Vererbung (Thema F)“ und das Thema „Assoziation (Thema G)“ inklusive kleiner Unterthemen (in Klammern):

*Thema T8: Vererbung (inkl. abstrakte Klassen und abstrakte Methoden)*

*Thema T9: Assoziationen (inkl. Klassenattribute und Klassenmethoden)*

Aus den sieben Themen A bis G wurden jetzt neun Unterrichtsthemen generiert. Die Lernziele bzw. Fachkompetenzen für die einzelnen Themen müssen noch bestimmt werden. Dies geschieht im Kapitel 4.1.

Aus der Analyse der zeitlichen Abfolge des Unterrichts aus Kapitel 1.2.2 (speziell auch der Abbildung 4) und den o. g. Erläuterungen ergibt sich die in Abbildung 29 aufgeführte Reihenfolge für das OOP-First-Vorgehen.

Man sieht, dass das Thema Methoden an verschiedener Stelle bei der Vorstudie und bei der Hauptstudie unterrichtet wurde. Der unterrichtende Kollege hat sich aufgrund des realen Unterrichtsgeschehens während der Hauptstudie entschieden, die Reihenfolge der Themen T4 und T5 zu vertauschen. Dies ist aus fachdidaktischer Sicht in Ordnung. Es geht ja nicht um eine exakte Festschreibung der Themenfolge, sondern um das Modell, dass beide Vorgehensweisen in der Summe die gleichen Themen unterrichten und dass die Themen in beiden Klassen sequenziell unterrichtet werden. Z. B. kann auch die Reihenfolge der Steuerstrukturen Selektion und Iteration umgedreht werden.

Reihenfolge	Thema
1	T1: Einführung in die OOP (mit BlueJ)
2	T2: Einführung in den Java-Editor, Attribute (Variablen), fundamentale Datentypen, Steuerstruktur Sequenz
3	T3: Klassenimplementierung (Fachkonzept) und Objekterzeugung im Anwendungsprogramm (Benutzungsoberfläche)
4/5*	T4: Methoden
5/4*	T5: Steuerstruktur: Selektion (inkl. Struktogramm)
6	T6: Steuerstruktur: Iteration (inkl. Struktogramm)
7	T7: Komplexere Datentypen: Arrays und Strings
8	T8: Vererbung (inkl. abstrakte Klassen und abstrakte Methoden)
9	T9: Assoziationen (inkl. Klassenattribute und Klassenmethoden)

Abbildung 29: Reihenfolge der OOP-First-Themen (vgl. z. B. [Ehl07b, S. 19] und [ES09a, S. 18])

(\* linke Zahl: Reihenfolge Vorstudie, rechte Zahl: Reihenfolge Hauptstudie)

Die OOP-Later-Klasse fokussiert sich erst einmal (ca. ½ Schuljahr) auf so genannte prozedurale Themen inkl. Daten- und Steuerstrukturen. Aus der Analyse der zeitlichen Abfolge des Unterrichts aus Kapitel 1.2.2 (speziell auch der Abbildung 4) und den o. g. Erläuterungen ergibt sich die in Abbildung 30 aufgeführte Reihenfolge für das OOP-Later-Vorgehen.

Reihenfolge	Thema
1	T2: Einführung in den Java-Editor, Variablen (Attribute), fundamentale Datentypen, Steuerstruktur Sequenz
2	T5: Steuerstruktur: Selektion (inkl. Struktogramm)
3	T6: Steuerstruktur: Iteration (inkl. Struktogramm)
4	T7: Komplexere Datentypen: Arrays und Strings
6/5*	T1: Einführung in die OOP (mit BlueJ)
7/6*	T3: Klassenimplementierung (Fachkonzept) und Objekterzeugung im Anwendungsprogramm (Benutzungsoberfläche)
5/7*	T4: Methoden
8	T8: Vererbung (inkl. abstrakte Klassen und abstrakte Methoden)
9	T9: Assoziationen (inkl. Klassenattribute und Klassenmethoden)

Abbildung 30: Reihenfolge der OOP-Later-Themen (vgl. z. B. [Ehl07b, S. 19] und [ES09a, S. 18])

(\* linke Zahl: Reihenfolge Vorstudie, rechte Zahl: Reihenfolge Hauptstudie)

In der Hauptstudie wurde das Thema Methoden, welches in der Vorstudie noch vor dem Einstieg in die OOP unterrichtet wurde, jetzt nach hinten gezogen (siehe Kapitel 3.3.14). Letztlich sollen die Themen in der Hauptstudie in der Reihenfolge, die in Abbildung 31 dargestellt ist, unterrichtet und am Ende mit einem Fragebogen abgefragt werden.

OOP-First	T1	T2	T3	T4*	T5*	T6	T7	T8	T9
OOP-Later	T2	T5	T6	T7	T1	T3	T4	T8	T9
Geplantes Ende des Themas (Befragungs-Datum)	Sept. 2007	Okt. 2007	Nov. 2007	Dez. 2007	Febr. 2008	März 2008	April 2008	Mai 2008	Juni 2008

Abbildung 31: Geplante Reihenfolge der Themen bzw. geplante Befragungstermine in der Hauptstudie

(vgl. [ES10a, Figure 2, S. 109] und Abbildung 144: reale Befragungstermine, \* Themen wurden real vertauscht unterrichtet)

Die Länge der Unterrichtszeit für ein bestimmtes Thema soll dabei in beiden Klassen gleich sein.



### 3.3.9 Festlegung der Unterrichtsmethoden, der Arbeitsmittel und des Medieneinsatzes

Das Ziel ist es, möglichst viele Einflussvariablen gleichartig zu gestalten [vgl. ES09a, S. 16 und Kapitel 3.3.5]. Bisher wurde für die Gleichartigkeit der Klassen (in Bezug auf Geschlecht, Alter, letzte Schule, Mathematiknote etc.) und der Themen (in Bezug auf Anzahl, Inhalte, Unterrichtszeit etc.) gesorgt. Jetzt muss dafür gesorgt werden, dass der Unterrichtsablauf, die Unterrichtsmethoden, die Arbeitsmittel und der Medieneinsatz gleichartig gestaltet werden (Im folgenden Abschnitt kursiv hervorgehoben).

Mein Kollege und ich haben uns darauf geeinigt, in ca. 1/3 der Unterrichtszeit „Theorie“ zu unterrichten. Dabei setzen wir als Unterrichtsmethode größtenteils das „*fragend-entwickelnde Unterrichten*“ ein. Als Medien im Theorie-Unterricht stehen *Tafel und Beamer* zur Verfügung. Bei den praktischen Übungen (ca. 2/3 des Unterrichts) soll als Sozialform eine Mischung aus *Einzel- und Partnerarbeit* vorherrschend sein. Dabei dienen *Informations- und Arbeitsblätter* dem Ziel, dass die Schülerinnen und Schüler selbstständig kleine Programme in Java schreiben. Die Klassen haben abwechselnd jede Woche entweder 1 Block (2 Schulstunden) oder 2 Blöcke (4 Schulstunden) Unterricht, der Durchschnitt beträgt also 3h pro Woche. Eine Schulstunde hat 45 Minuten. Bei Bedarf steht immer ein Computerraum zur Verfügung, jeder der 22 Schülerinnen und Schüler hat dort einen *eigenen PC-Arbeitsplatz*. Der Computerraum hat eine *Tafel und einen Beamer*.

Der Unterrichts-Ablauf der einzelnen Themen ist wie folgt vereinbart:

- Allgemeine Information zum Thema, z. B. Erläuterungen zu einer Vererbungsstruktur. In der Regel erfolgt eine Ausgabe eines Informationsblattes (siehe Anhang T). Teil der Erläuterung könnte ein OOA-Klassendiagramm sein, welches eine Vererbungsstruktur (mit einer abstrakten Klasse) zeigt (siehe Abbildung 32).

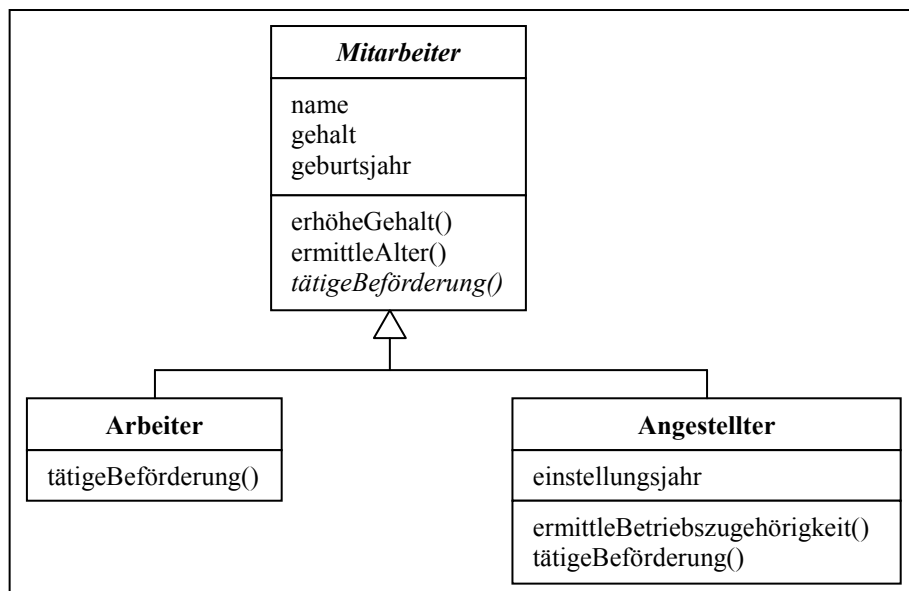


Abbildung 32: Teil eines allgemeinen Informationsblatts (hier Thema 8: Vererbung)

- Spezielle Information über die Umsetzung in Java, z. B. Erläuterungen zur Implementierung einer Vererbung, einer abstrakten Klassen und einer abstrakten Methode. In der Regel erfolgt eine Ausgabe eines

Informationsblattes (siehe Anhang T). Teil der Erläuterung könnte z. B. ein Quellcode-Fragment sein, welches Teile der Implementierung zeigt (siehe Abbildung 33).

```
// Basisklasse bzw. Oberklasse
public abstract class Mitarbeiter {
    private String name;
    private int gehalt;
    private int geburtsjahr;
    public Mitarbeiter()           {...}
    public Mitarbeiter(String,int,int) {...}
    public void setName(String)   {...}
    public String getName()       {...}
    public void setGehalt(int)     {...}
    public int getGehalt()        {...}
    public void erhoeheGehalt(int) {...}
    public void erniedrigeGehalt(int) {...}
    public void setGeburtsjahr(int) {...}
    public int getGeburtsjahr()   {...}
    public int ermittleAlter(int) {...}
    public abstract void taetigeBefoerderung();
} // Basisklasse Mitarbeiter
```

Abbildung 33: Teil eines Informationsblatts zur Java-Implementierung (hier Thema 8: Vererbung)  
(siehe auch Anhang T und [Ehl05, S. 20])

- Praktische Übung für die Schülerinnen und Schüler, z. B. sollen sie die Vererbungsstruktur aus Abbildung 32 implementieren und erweitern. Hierfür gibt es in der Regel ein Arbeitsblatt (siehe Anhang T). Dieses enthält eine Reihe von Aufgaben (siehe Abbildung 34).

Die OOP-Later-Klasse wird mit Hilfe zweier Skripte unterrichtet:

- 1. Skript: Einführung in die Programmierungstechnik [Ehl07a]
- 2. Skript: Objektorientierte Programmierung (OOP) und Unified Modeling Language (UML) mit Übungen in Java [Ehl05]

- Implementieren Sie die Klasse Mitarbeiter und die Methode „**taetigeBefoerderung()**“ als abstrakt (**Mitarbeiter.java**).
  - Führen Sie die 3 neuen abgeleiteten Klassen Angestellter (**Angestellter.java**), Arbeiter (**Arbeiter.java**) und Manager (**Manager.java**) ein.
  - Implementieren Sie die abgeleitete Klasse Auszubildender (**Auszubildender.java**). Die zusätzlichen Attribute sind „**ausbildungsbeginn**“ und „**ausbildungsdauer**“. Die zusätzlichen Methoden sind „**ermittleAusbildungsjahr(int AktJahr)**“, „**setAusbildungsdauer(int Dauer)**“, „**getAusbildungsdauer()**“, „**setAusbildungsbeginn(int )**“ und „**getAusbildungsbeginn()**“. Da ein Azubi nicht befördert werden kann, darf die ursprüngliche Methode „**taetigeBefoerderung()**“ keine Auswirkung auf das Gehalt haben. Jedes Ausbildungsjahr bewirkt aber 200 EURO mehr Ausbildungsbeihilfe.
  - Schreiben Sie ein Hauptprogramm (**OOP4.java**), in welchem Sie **4 Objekte** erzeugen, denn die Firma hat 4 Mitarbeiter.
- ...

Abbildung 34: Auszug aus einem typischen Arbeitsblatt (hier Thema 8: Vererbung)  
(siehe auch Anhang T und [Ehl05, S. 23])

Die OOP-First-Klasse bekommt ähnlich gestaltete Informations- und Arbeitsblätter. Sie unterscheiden sich aber bei den ersten sieben Themen von denen der OOP-Later-Klasse, da die inhaltlichen Vorbedingungen in beiden Klassen bei den einzelnen Themen unterschiedlich sind. Nur bei den letzten beiden Themen (Vererbung und

Assoziation) können dieselben Informations- und Arbeitsblätter (siehe Anhang T) verwendet werden, da dann die beiden Klassen auf dem selben Stand sind. Auch die OOP-First-Klasse bearbeitet die OOP-Übungen 2 bis 7 (OOP2 bis OOP7) aus dem 2. Skript [Ehl05] (siehe Übungen bei den einzelnen Themen in Abbildung 37).

Zusammenfassend kann Folgendes festgestellt werden:

- Unterrichtsablauf, Unterrichtsmethoden und Medieneinsatz können gleichartig gestaltet werden.
- Die Arbeitsblätter, Programmierungsbeispiele und Informationsblätter können dagegen nicht identisch sein, da die Schüler bei den meisten Themen jeweils unterschiedliches Vorwissen haben. Es wird aber die gleiche Struktur der 3-Gliedrigkeit von allgemeiner Information, Java-Implementierungs-Information und Programmier-Aufgabe verwendet.
- Die Arbeitsmittel gleichen sich im Laufe des Schuljahres immer mehr an und sind bei den letzten beiden Themen identisch.

Einen Einblick in den Unterrichtsablauf geben auch die Kurzarbeitspläne für die beiden Klassen (siehe Abbildung 35 und Abbildung 36).

### 3.3.10 Randbedingungen zur Auswahl der unterrichtenden Lehrer

Wenn zwei fachdidaktische Vorgehensweisen im Unterricht „gegeneinander“ antreten, müssen alle Unterrichtselemente kontrolliert werden, die man im Sinne einer personenbezogenen Einflussnahme (durch den Lehrer) interpretieren könnte.

Es gibt bei der „Lehrer-Implementierung“ dieser Studie die folgenden vier Möglichkeiten:

- A. Zwei Kollegen unterrichten beide Klassen.
- B. Ein Kollege unterrichtet beide Klassen.
- C. Ein Kollege und ich unterrichten beide Klassen.
- D. Ich unterrichte beide Klassen.

Szenario B und D scheiden leider aus formalen stundenplantechnischen Gründen aus. Beide Klassen haben manchmal den Technikunterricht zur selben Zeit.

Das Szenario A hat die folgenden Schwierigkeiten: Es müssen gleich zwei Lehrer gefunden werden, die bereit sind, an der Studie mitzuwirken (dies sind insgesamt mit Vor- und Hauptstudie zwei Jahre) und bereit sind, die starken Vorgaben zu akzeptieren. Die beiden Kollegen müssten (von mir) ständig kontrolliert und synchronisiert werden.

Also verbleibt realistisch nur noch Szenario C, welches m. E. aber auch Vorteile hat:

- Nur ein Kollege (statt zwei Kollegen) muss gefunden werden.
- Ich bin als beteiligter Lehrer ständig an der realen Ausführung der Studie beteiligt und kann den Prozess besser beobachten und kontrollieren.

Natürlich stellt sich durch eine derart von Randbedingungen geprägte Implementierungs-Entscheidung sofort die Frage: Sind zwei unterschiedliche Lehrer nicht ein Design-Fehler der Studie?

Einige Studien blenden den Lehrereinfluss einfach mit einem Ausdruck des Bedauerns aus. So schreiben Seifried und Klüber [SK06]:

„Die Vergleichsgruppen wurden von unterschiedlichen Lehrpersonen unterrichtet, die sich jeweils freiwillig für die Teilnahme am Forschungsprojekt entschieden hatten. Eine Kontrolle des Lehrereffektes war aus forschungspragmatischen Gründen bedauerlicherweise nicht möglich.“ [SK06, S. 6].

Diese Studie will es sich nicht so einfach machen. Es gibt zwar auch gute Gründe, warum ich als Versuchsleiter nicht beide Klassen unterrichten sollte, z. B. weil aus meinen Thesen eine gewisse Affinität zu einem OOP-Later-Vorgehen herauszulesen ist. Somit könnte später der Vorwurf gemacht werden, ich habe (bewusst oder unterbewusst) in der OOP-First-Klasse „schlecht“ unterrichtet.

Trotzdem ist diese Implementierung mit zwei unterschiedlichen Lehrern natürlich nicht optimal: Der Einfluss des Lehrers auf den Unterricht ist sehr groß. Es muss somit alles versucht werden, um den Einfluss der unterschiedlichen Lehrer auf das Untersuchungsergebnis so gering wie möglich zu halten bzw. für eine gleichartige Einflussnahme zu sorgen. Die „Variable Lehrer“ muss daher stark kontrolliert werden. Im nächsten Kapitel werden eine Vielzahl von Kontroll- und Synchronisations-Mechanismen genannt, die für die Gleichartigkeit der Lehrer sorgen sollen. Zusätzlich wird über Schülerbefragungen die „Güte der Gleichartigkeit“ der eingesetzten Lehrer ermittelt (vgl. Kapitel 7.2.1.: „Waren die Lehrer wirklich gleichartig?“).

### 3.3.11 Synchronisation der beiden Lehrer bzw. des Unterrichts

Wie kann das Problem der Sicherstellung der Gleichartigkeit der Lehrer bzw. des Unterrichts gelöst bzw. reduziert werden? Dies kann m. E. nur mit sehr konkreten Absprachen erfolgen, an die sich die beiden Lehrer halten:

- Absprachen zu den Unterrichts-Methoden (siehe Kapitel 3.3.9),
- Absprachen zu den Arbeitsmitteln und zum Medieneinsatz (siehe Kapitel 3.3.9),
- Absprachen zum Verhältnis von Theorie und Praxis (siehe Kapitel 3.3.9),
- Absprachen zu den Übungen, Hausaufgaben, Tests und Klassenarbeiten (s. u.),
- Absprachen zu den verwendeten Tools (siehe Randbedingungen im Kapitel 3.3.1),
- Absprachen zu den Themen (siehe Kapitel 3.3.8 und Abbildung 37) inkl. Lernziele (siehe Kapitel 4.1),
- Absprachen zur Unterrichtszeit bezogen auf die einzelnen Themen (siehe Abbildung 35 und Abbildung 36).

Dies bedeutet, dass mein Kollege und ich im Vorfeld des Schuljahres sehr viele Dinge besprechen und festlegen und während des Schuljahres regelmäßige Treffen haben, um unseren Unterrichtsstil, die Unterrichtsmethoden und das Unterrichtsmaterial möglichst gleichartig zu gestalten [ES09a, S. 17].

Woche	OG71 (OOP-First)	OG72 (OOP-Later)
27.08. – 31.08.	Einführung in die Informatik	Einführung in die Informatik
03.09. – 07.09.	T1: BlueJ	T2: IDE, Datentypen, Sequenz.
10.09. – 15.09.	T1	T2: Variablen, Konstanten
17.09. – 21.09.	PSE-	FAHRT
24.09. – 28.09.	Stoffausgleich	Stoffausgleich
01.10. – 05.10.	T2: IDE, Datentypen, Sequenz	T5: Selektionen:
08.10. – 12.10.	T2: Variablen, Konstanten	T5:
15.10. – 19.10.	HERBST-	FERIEN
22.10. – 26.10.	Stoffausgleich	Stoffausgleich
29.10. – 02.11.	erste	Klassenarbeit
05.11. – 09.11.	T3: IDE, Klasse, Objekt	T6: Iterationen
12.11. – 16.11.	T3: setter / getter/ Konstrukt.	T6
19.11. – 23.11.	Stoffausgleich	Stoffausgleich
26.11. – 30.11.	zweite	Klassenarbeit
03.12. – 07.12.	T4: Methoden	T7: Array / Strings
10.12. – 14.12.	T4	T7:
17.12. – 21.12.	Zeugnis-	ausgabe
24.12. – 13.01.	WEIHNACHTS-	FERIEN

Abbildung 35: Kurzarbeitsplan (1. Halbjahr 2007/2008)

Woche	OG71 (OOP-First)	OG72 (OOP-Later)
24.12. – 13.01.	WEIHNACHTS-	FERIEN
14.01. – 18.01.	Wiederholung	Wiederholung
21.01. – 25.01.	T5: Selektionen	T1: BlueJ
28.01. – 01.02.	T5	T1
05.02. – 08.02.	Stoffausgleich	Stoffausgleich
11.02. – 15.02.	erste	Klassenarbeit
18.02. – 22.02.	T6: Iterationen	T3: Klasse, Objekt
25.02. – 29.02.	T6	T3
03.03. – 07.03.	Stoffausgleich	Stoffausgleich
10.03. – 14.03.	Stoffausgleich	Stoffausgleich
15.03. – 30.03.	OSTER-	FERIEN
31.03. – 04.04.	T7: Array & Strings	T4: Methoden
07.04. – 11.04.	T7	T4
14.04. – 18.04.	Stoffausgleich	Stoffausgleich
21.04. – 25.04.	Stoffausgleich	Stoffausgleich
28.04. – 30.04.	Kurz-	Woche
01.05. – 18.05.	FERIEN bzw.	STOFFAUSGLEICH
19.05. – 23.05.	T8: Vererbung	T8: Vererbung
26.05. – 30.05.	T8	T8
02.06. – 06.06.	Stoffausgleich	Stoffausgleich
09.06. – 13.06.	zweite	Klassenarbeit
16.06. – 20.06.	T9: Assoziationen	T9: Assoziationen
23.06. – 27.06.	T9: Klassenattribute/-methoden	T9: Klassenattribute/-methoden
30.06. – 04.07.	VERGLEICHS-	TEST
07.07. – 11.07.	Stoffausgleich	Stoffausgleich
14.07. + 15.07.	ZEUGNIS-	AUSGABE

Abbildung 36: Kurzarbeitsplan (2. Halbjahr 2007/2008)

Bestärkt wird diese Sichtweise später durch die Ergebnisse der Vorstudie. Die (signifikanten) Unterschiede in den Ergebnissen lassen sich durch eine teilweise mangelnde Synchronisation der Lehrer bzw. des Unterrichts erklären. Daher werden die Kontroll- und Synchronisationsmechanismen in der Hauptstudie wesentlich verstärkt.

Als Beispiel soll der Kurzarbeitsplan für das erste Schulhalbjahr (Hauptstudie) angeführt werden (siehe Abbildung 35). Man sieht, dass die Themen, die Themenreihenfolge und die Themenlänge genau festgelegt sind. Dies setzt sich auch im zweiten Schulhalbjahr fort (siehe Abbildung 36).

Zusätzlich müssen die Themen näher spezifiziert werden, z. B. die Themenschwerpunkte genannt werden (siehe Abbildung 37) und die Lernziele formuliert werden (siehe Kapitel 4.1). Der Lernzielkatalog liegt den unterrichtenden Lehrern am Beginn des Schuljahres vor.

Thema-Nr.	OOP-First Abfolge	OOP-Later Abfolge	THEMA	Übung
T1	1	5	<b>Einführung in die OOP mit BlueJ:</b> Klasse, Objekt, Attribut, Methode	BlueJ: Figuren, Zeichnung
T2	2	1	Einführung in den <b>Java-Editor</b> , main-Methode, <b>Ein- und Ausgabe</b> <b>Datentypen:</b> byte, short, int, long, char, float, double <b>Variablen, Konstanten</b> , Steuerstruktur <b>Sequenz</b> inkl. Struktogramm	Berechnung des Body-Maß-Index
T3	3	6	IDE: <b>Klassenimplementierung und Objekterzeugung:</b> Setter, getter, Konstruktoren, UML: Klassen- und Objektdiagramm Trennung von Klasse und Anwendungsprogramm	OOP2 und OOP3
T4	5	7	<b>Methoden:</b> Deklaration contra Aufruf, Rückgabewerte, aktuelle und formale Parameter, statische Methoden	
T5	4	2	<b>Steuerstruktur Selektion:</b> Fallunterscheidung, ein- und zweiseitige Auswahl, verschachtelte Auswahl inkl. Struktogramme, Boolescher Datentyp und log. Operatoren	
T6	6	3	<b>Steuerstruktur Iteration:</b> kopfgesteuerte Schleife, fußgesteuerte Schleife, Zählschleife und Endlosschleife inkl. Struktogramme	
T7	7	4	<b>Arrays und Strings:</b> Deklaration von Feldern, Arbeiten mit Feldern und Schleifen Methodenaufrufe: length(), equals(), toUpperCase()	Initialisierung von Feldern, Suche nach der kleinsten Zahl
T8	8	8	<b>Vererbung:</b> Generalisierung, Spezialisierung, abstrakte Klassen und Methoden Überschreiben, Konstruktoren, Polymorphie	OOP4 und OOP5
T9	9	9	<b>Assoziationen</b> Realisierung einer 1:1- und 1:n-Beziehung Klassenattribute und Klassenmethoden	OOP6 und OOP7

Abbildung 37: Themen und Themenschwerpunkte im Schuljahr 2007/2008) für die beiden Lehrer  
(mit leichten redaktionellen Veränderungen am Original, z. B. im Layout)

Im Kapitel 3.3.9 wurden für den Unterricht bzw. die Lehrer die Unterrichts-Methoden, die Arbeitsmittel, der Medieneinsatz und die Theorie- und Praxisaufteilung festgelegt. Zusätzlich müssen mein Kollege und ich dafür sorgen, dass ähnliche Übungen (siehe auch Abbildung 37) durchgeführt werden und die Tests und

Klassenarbeiten in Qualität und die Quantität vergleichbar sind. Dies gilt auch für die Art, die Anzahl und den Umfang der Hausaufgaben.

Letztlich muss ein Kollege gefunden werden, der von seiner Art mir ähnelt und genau so viel Unterrichtserfahrung aufweist wie ich. Weil aus meinen Thesen eine gewisse Affinität zu einem OOP-Later-Vorgehen herauszulesen ist, werde ich die OOP-Later-Klasse unterrichten, damit nicht der Vorwurf gemacht werden kann, ich habe (bewusst oder unterbewusst) in der OOP-First-Klasse „schlecht“ unterrichtet. Umgekehrt gilt dies natürlich auch. Deshalb soll der eingesetzte Kollege in der OOP-First-Klasse von dem objects-first-Vorgehen überzeugt sein. Dies ist ein weiteres Kriterium bei der Auswahl eines entsprechenden Kollegen. Die Wahl fällt auf einen Kollegen (in dieser Arbeit als OOP-First-Lehrer bezeichnet), der sowohl genug Kompetenz und Erfahrung im Informatikunterricht (z. B. als Fachleiter Informatik und Fachseminarleiter Informatik) hat als auch das OOP-First-Vorgehen befürwortet. Er erklärt sich freundlicherweise dazu bereit, an dieser Studie teilzunehmen.

Um zu kontrollieren, ob wir beiden Lehrer vergleichbar sind, werden in den subjektiven Fragebogen folgende Fragen eingebaut (siehe auch Anhang K und Kapitel 5):

- Frage 7: Den unterrichtenden Lehrer finde ich ... („sehr gut“ bis „sehr schlecht“ bzw. „keine Aussage“)
- Frage 17: Die Darstellung und Veranschaulichung der Lerninhalte des letzten Themas waren ... („sehr verständlich“ bis „sehr unverständlich“)
- Frage 19: Die Unterstützung des Lehrers bei meinen Lernversuchen beim letzten Thema war ... („sehr gut“ bis „sehr schlecht“ bzw. „keine Aussage“)

Da die Schülerinnen und Schüler nur pseudonym sind (vgl. Kapitel 3.3.7), müssen sie auch die Gelegenheit erhalten, auf die Fragen (mit Hilfe der Ankreuzmöglichkeit „keine Aussage“) nicht zu antworten.

Auch die nächste Frage hat u. a. etwas mit der Person des Lehrers zu tun:

- Frage 18: Die Arbeitsatmosphäre in der Klasse beim letzten Thema war eher ... („sehr gut“ bis „sehr schlecht“)

Das OSZ Informations- und Medizintechnik hat zur internen Evaluation der Unterrichts- und Lehrer-Qualität einen eigenen Online-Fragebogen entwickelt. Er enthält 17 Fragen (siehe Kapitel 7.2.1), die sich in vier Bereiche aufteilen:

1. Rahmenbedingungen des Unterrichts (Fragen 1 bis 7)
2. Zufriedenheit mit dem Lehrer (Fragen 8 bis 13)
3. Individuelle Komponenten (Fragen 14 bis 16)
4. Allgemeine Einschätzung (Frage 17)

Die Fragen z. B. aus dem Bereich 2 lauten wie folgt:

- Frage 8: Die fachliche Kompetenz und die Kenntnisse des Lehrers sind ... („sehr hoch“ bis „sehr gering“)
- Frage 9: Der Unterrichtsstoff wurde klar und übersichtlich vermittelt. („trifft voll zu“ bis „trifft überhaupt nicht zu“)
- Frage 10: Mit der sprachlichen Ausdrucksweise und der Verständlichkeit der Erklärungen des Lehrers war ich ... („sehr zufrieden“ bis „sehr unzufrieden“)

- Frage 11: Mit dem persönlichen Auftreten (Engagement, Freude, Spaß am Lehren) des Lehrers war ich ... („sehr zufrieden“ bis „sehr unzufrieden“)
- Frage 12: Die soziale Kompetenz des Lehrers (Hilfsbereitschaft, Verhalten gegenüber Kritik, etc.) ist ... („sehr hoch“ bis „sehr gering“)
- Frage 13: Das Eingehen des Lehrers auf Beiträge und Fragen der Schülerinnen/Schüler war für mich ... („völlig ausreichend“ bis „völlig unzureichend“)

Die Frage aus dem Bereich 4 soll eine allgemeine Einschätzung über den Unterricht geben:

- Frage 17: Insgesamt erhält der Unterricht von mir die Note ... (1 bis 6)

Man sieht, dass hier in der Regel das gleiche Bewertungsraster (Kreuz an einer von fünf Stellen) vorhanden ist wie später beim Fragebogen zum subjektiven Erleben. Allerdings wird diese Online-Befragung nicht nach jedem Thema durchgeführt, sondern nur einmal gegen Ende des Schuljahres (bei der Hauptstudie: Juni/Juli 2008) und die Befragung ist anonym.

Die ganzen in diesem Kapitel aufgeführten Kontroll- und Synchronisationsmechanismen dienen dazu, den Einfluss der „Störvariablen Lehrer“ zu kontrollieren bzw. zu minimieren. Zwei weitere Kapitel beschäftigen sich mit Implementierungs-Problemen:

- Kapitel 3.3.14: „Vorstudie und deren Lehren für die Hauptstudie“
- Kapitel 3.3.15: „Problemfelder: Lehrer, Vortest und fehlende Schüler“

Ob die Synchronisation des Unterrichts und der Lehrer am Ende gelingt wird in der späteren Auswertung diskutiert, u. a. in:

- Kapitel 7.1.1: „Konnte die Themenreihenfolge wie geplant unterrichtet werden?“
- Kapitel 7.1.2: „Konnten die Befragungen zu den Themen zeitlich wie geplant erfolgen?“
- Kapitel 7.2.1: „Waren die Lehrer wirklich gleichartig?“

### 3.3.12 Vortest, Vergleichstest und Nachhaltigkeitstest

Im Kapitel 3.3.2 wurde erläutert, dass aus der ersten Frage des Entscheidungsexperiments („Erzielt das OOP-First-Vorgehen oder das OOP-Later-Vorgehen bessere Lernergebnisse (unterteilt in einzelne Themengebiete)?“) ersichtlich wird, dass es *verschiedene Tests* (Hinweis: Implementierungsfaktoren und -entscheidungen sind in diesem Kapitel kursiv gekennzeichnet.) geben muss, welche das Erreichen von Lernzielen (siehe Kapitel 4.1) abprüfen. Und die Tests wurden im Vorfeld schon benannt und in einen zeitlichen Kontext gestellt (siehe Abbildung 24: Design der Hauptstudie). Der *zeitliche Ablauf der Tests* ergibt sich somit aus dem Studiendesign.

Im Folgenden sollen zusätzlich die *Anzahl der Tests*, die *Art der Tests*, der *Umfang der Tests* und die *Art der Fragen* begründet und festgelegt werden.

Im Kapitel 2.2 wurden u. a. internationale Studienergebnisse von vergleichenden Untersuchungen vorgestellt, die manchmal Unterschiede bei den Lernergebnissen in Abhängigkeit von einer bestimmten fachdidaktischen



Vorgehensweise festgestellt haben. Tew, McCracken und Guzdial [TMG05] bzw. Vilner, Zur und Gal-Ezer [VZG07] berichten, dass sich aber die Unterschiede nach einiger Zeit verflüchtigen. Dies ist ein Grund dafür, dass bei dieser Studie ein *Nachtest* (Post-Test) nicht nach jedem der neun Themen erfolgt, sondern am Ende des Schuljahres, nachdem alle Themen unterrichtet wurden (siehe Kapitel 4.3.1). Dies befindet sich auch in Übereinstimmung mit den bisherigen Gepflogenheiten am Beruflichen Gymnasium des OSZ IMT. Die Informatik-Lehrer der Einführungsphase haben schon seit dem Jahr 2004 am Ende des Schuljahres, also *kurz vor den Sommerferien*, einen so genannten *Vergleichstest* geschrieben (siehe Anhang C).

Um die Ergebnisse eines Vergleichstest (als Nachtest) objektiver bewerten zu können, wird bei diesem Studiendesign ein *Vortest-Nachtest-Design* implementiert, da ein *Vortest* (Pre-Test) das Ursprungswissen der Schülerinnen und Schüler abfragt und somit der Lernzuwachs besser zu ermitteln ist. Dieser Test muss gleich *zu Beginn des Schuljahres* geschrieben und sollte *kurz gehalten* werden und darf keinen Einfluss auf das spätere Lernverhalten der Schülerinnen und Schüler haben (siehe Kapitel 4.3.3).

In der bildungspolitischen Debatte wird immer mehr der Begriff der Nachhaltigkeit bzw. des nachhaltigen Lernens genannt. Ingeborg Schüssler [Sch01] versteht darunter:

„Im lerntheoretischen Verwendungskontext bezieht sich Nachhaltigkeit auf den Prozess des dauerhaften Erwerbs und die Festigung von Kenntnissen, Fähigkeiten und Fertigkeiten [...]“ [Sch01, S. 2].

Sie spricht von einer Mehrdimensionalität des Qualitätsmerkmals „Nachhaltigkeit“ und bezeichnet die Monate nach dem Unterricht bzw. Training als Vergessens- bzw. Routinierungsphase (siehe Abbildung 38). Wie viel die Schülerinnen und Schüler von dem vermittelten Stoff dauerhaft behalten wird in dieser Studie mit Hilfe eines *Nachhaltigkeitstests* (Follow-Up-Tests) ermittelt (siehe Kapitel 4.3.4). Allerdings kann dieser Test nicht beliebig viele Monate nach Ende des Schuljahres geschrieben werden, sondern muss maximal zwei Monate später (*nach den Sommerferien*) geschrieben werden (siehe auch Abbildung 24), weil danach die Schüler in der Kursphase in den verschiedenen Leistungskursen unterschiedlich in informatischen Themen weitergeführt werden.

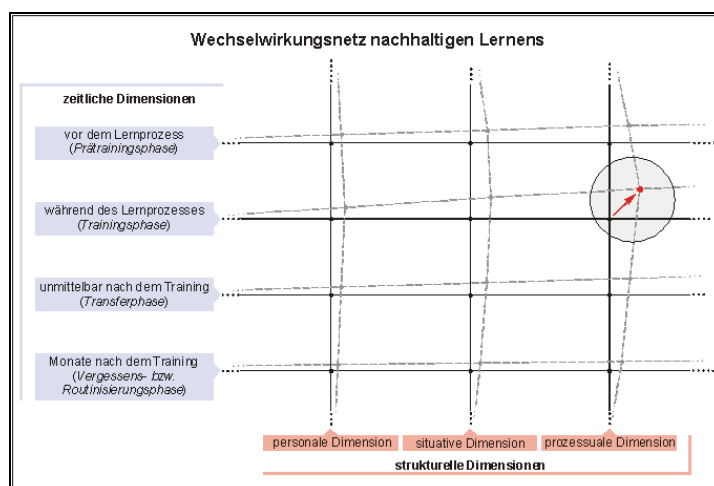


Abbildung 38: Mehrdimensionalität des Qualitätsmerkmals „Nachhaltigkeit“ [Sch01, S. 18, Abb. 7]

Der Vergleichstest und der Nachhaltigkeitstest müssen sehr ähnlich sein (sowohl von den Fragen als auch von der Zeit), um die Ergebnisse vergleichen zu können und sie müssen von beiden beteiligten Lehrern gemeinsam erarbeitet werden, um keine der beiden Klassen zu bevorzugen oder zu benachteiligen. Die *Fragen werden in die*

*einzelnen Themengebiete aufgeteilt*, um auch einen Zusammenhang auch mit dem subjektiven Erleben herstellen zu können. Objektorientierte Themen (bzw. Fragen) sollen sich im Gleichgewicht befinden zu so genannten prozeduralen Themen (bzw. Fragen), damit beide Elemente eines Programmier-Anfangsunterrichts gleichstark im Test vertreten sind. Sollte es themenübergreifende Inhalte geben (siehe Kapitel 4.2), dann können diese auch geprüft werden, damit eventuelle Nebeneffekte der beiden fachdidaktischen Vorgehensweisen erkannt werden können. Angesichts der Tatsache, dass also mindestens neun Themen abgefragt werden, kann der *zeitliche Rahmen des Vergleichs- und Nachhaltigkeitstests mehrere Schulstunden* betragen, damit einerseits genug Fragen pro Themengebiet gestellt werden können, andererseits die Schüler nicht unter Zeitdruck geraten. Die Punkte, die es für Antworten auf *Multiple-Choice-Fragen* gibt, sollen *unter 35%* der erreichbaren Gesamtpunkte liegen, da es nach der Notenskala für die gymnasiale Oberstufe in Berlin (siehe Abbildung 153) schon ab 35% die Note 5+ gibt.

Der zeitliche Ablauf der Tests wird in Abbildung 24 dargestellt, die genaue Implementierung der Tests erfolgt im Kapitel 4.3.

### 3.3.13 Fragebögen zum subjektiven Erleben

Im Kapitel 3.3.2 wurde erläutert, dass aus der zweiten Frage des Entscheidungsexperiments („Ist das subjektive Erleben der Schüler positiver beim OOP-First-Vorgehen oder beim OOP-Later-Vorgehen (unterteilt in einzelne Items)?“) ersichtlich wird, dass *ein Fragebogen* (Hinweis: Implementierungsfaktoren und -entscheidungen sind in diesem Kapitel kursiv gekennzeichnet) entwickelt werden muss, der die subjektiven Aspekte des Unterrichtsgeschehens abfragt.

In früheren Studien (siehe Kapitel 2.2) wurde verschiedene Einflussfaktoren (innere Einstellungen und Charakteren der Schüler bzw. Studenten) benannt, z. B. das eigene Selbstvertrauen, das Abstraktionsvermögen, die Einstellung zur Informatik und bevorzugte Lösungsstrategien, und untersucht, inwiefern sie Auswirkungen auf den Lernerfolg haben [ES09a, S. 17]. Wenn in dieser Studie das subjektive Erleben (aufgeteilt in drei Erlebnis-Dimensionen) untersucht wird, also u. a. der „Wohlfühlfaktor“ und das „Schwere-Empfinden“, dann befindet sie sich z. B. im Einklang mit der Untersuchung von Wilson und Shrock [WS01]. Dort wurden zwölf Faktoren untersucht, die evtl. Einfluss auf den Erfolg eines Informatik-Anfangskurses nehmen können. Der Faktor „Comfort Level“ hat nach dieser Untersuchung den größten Einfluss.

Im Folgenden werden die formalen Randbedingungen für die Entwicklung der Fragebögen für die Hauptstudie festgelegt:

- Die einzelnen Fragen sollen sich auf die drei Erlebnis-Dimensionen „emotionaler Aspekt“, „motivationaler Aspekt“ und „kognitiver Aspekt“ beziehen, die im Kapitel 2.4.3 herausgearbeitet wurden. Damit lässt sich der Faktor „subjektives Erleben“ besser spezifizieren.
- Die einzelnen Fragen sollen sich auf verschiedene Kategorien beziehen, wie z. B. auf das „letzte Thema“ und das „Fach Informatik“. Damit kann der Faktor „subjektives Erleben“ auf verschiedene Bereiche des schulischen Erlebens bezogen werden.

- Der Fragebogen soll durch die Schülerinnen und Schüler in kurzer Zeit (ca. fünf Minuten) zu beantworten sein, z. B. können die Antwortmöglichkeiten zum Ankreuzen sein. Damit wird für die Umfrage von der Unterrichtszeit nicht zu viel in Anspruch genommen und die Schülerinnen und Schüler verlieren durch die Kürze der Umfrage nicht ihre Bereitschaft an der Studie teilzunehmen.
- Nach jedem der einzelnen Themen (also neunmal im Schuljahr) soll das Stimmungsbild erhoben werden. Damit wird der Unterricht als zeitlicher Prozess in den Mittelpunkt gestellt.

Der zeitliche Ablauf der Umfrage wird in Abbildung 24 dargestellt, die genaue Implementierung des Fragebogens erfolgt im Kapitel 5.

### 3.3.14 Vorstudie und deren Lehren für die Hauptstudie

Die Vorstudie diente dazu, das gesamte Studiendesign auszuprobieren und Erfahrungen zu sammeln [ES09a, S. 18]. Auch sie ging über ein gesamtes Schuljahr. Es stellte sich heraus, dass die Vorstudie tatsächlich einige Design- bzw. Umsetzungs-Schwächen hatte. Diese sollen hier schon einmal genannt werden, da das gesamte Kapitel 3.3 den Schwerpunkt bei der Implementierung der Hauptstudie hat und somit die Erfahrungen der Vorstudie mit berücksichtigt werden müssen.

Die unterrichteten Themen setzten in der Vorstudie noch auf dem bis dahin in der Einführungsphase des Faches Technik (Informatik) am OSZ IMT geltenden Kernarbeitsplan (siehe Anhang B) auf, der Grundlage für den Arbeitsplan (siehe Exposé in Anhang A) war. Die dort genannten ca. 20 Themen wurden dann in größere Abschnitte zusammengefasst, die zu diesem Zeitpunkt noch Module hießen.

Die Module wurden willkürlich nummeriert. Die zeitliche Reihenfolge des Unterrichtens der Module und der Befragung der Schülerinnen und Schüler im Hinblick auf die Module ist aus der Abbildung 39 und der Abbildung 40 ersichtlich.

Folgendes fällt bei den Befragungen auf:

- Sechs Module wurden in beiden Klassen gleich abgefragt (Module 1, 2, 3, 7, 10 und 20).
- Zwei Module wurden unterschiedlich abgefragt und die OOP-First-Klasse hat die Abfrage der Module 11 und 12 zusammengefasst.
- In der OOP-Later-Klasse wurde nicht das Modul „Array und Strings“ (Modul 5) abgefragt.
- In der OOP-First-Klasse wurde ein Zusatzmodul „Datenpersistenz: Dateien, Serialisierung“ abgefragt (Modul 6).
- Es gab in beiden Klassen keine Abfrage zum Thema „Assoziationen“.

Auffällig ist auch, dass der Unterricht in der OOP-First-Klasse viel schneller als in der OOP-Later-Klasse erfolgte. Dies bedeutet aber auch, dass für die einzelnen Themen weniger Zeit verwendet wurde [ES09a, S. 18]. Die „eingesparte“ Zeit in der OOP-First-Klasse wurde dann am Ende des Schuljahres (ab 1. April 2007) für ein kurzes Projekt genutzt. In Kapitel 6.2.1 wird dies als ein möglicher Grund genannt, warum die OOP-First-Klasse bei zwei Themen („dynamische OOP“ und „UML“) signifikant schlechter abgeschnitten hat als die OOP-Later-Klasse.

All diese Fakten zeugen von einer mangelhaften Synchronisation der beteiligten Lehrer. Für die Hauptstudie werden die Themen und die Zeit für die Themen im Voraus genau festgelegt [ES09a, S. 18]. Bei der Abfolge der Themen wird in der OOP-Later-Klasse (weil das Thema „Methoden“ so schlecht abgeschnitten hat, siehe Kapitel 6.2.1) das Thema „Methoden“ hinter den OO-Einstieg verlagert (vgl. Abbildung 30). Das Thema „Prozeduren“ wird jetzt nicht mehr vor dem OOP-Einstieg als „statische Methoden“, sondern erst nach dem OOP-Einstieg als „Methoden“ unterrichtet [ES09c, S. 127].

OG 61 (OOP-First)			
Lfd. Nr.	Datum	Modul	Thema
1	29.09.06	1	Einführung in die OOP mit BlueJ
2	03.11.06	10	Einführung in die IDE, Variablen, Konstanten, Datentypen, Wertzuweisungen
3	10.11.06	2	Setter / Getter / Konstruktoren, Trennung: Fachkonzept/Nutzerebene
4	01.12.06	11/12	Sequenz, alle Selektionen
5	15.12.06	3	Methoden
6	22.12.06	20	Kopf- und fußgesteuerte Schleife, Zählschleife inkl. Struktogramme
7	02.02.07	5	Arrays und Strings
(oB)	16.02.07	'6'	Datenpersistenz: Dateien, Serialisierung
8	30.03.07	7	Vererbung

Abbildung 39: Befragungstermine zu den einzelnen Modulen in der OOP-First-Klasse

(Vorstudie, vgl. [ES07a, S. 3, Figure 4])

OG 62 (OOP-Later)			
Lfd. Nr.	Datum	Modul	Thema
1	28.09.06	10	Einführung in Java, Variablen, Konstanten, Datentypen, Wertzuweisungen
2	02.11.06	11	Sequenz und Fallunterscheidung inkl. Struktogramm
3	20.11.06	12	Ein- und zweiseitige Auswahl, verschachtelte Auswahl, Struktogramm
4	14.12.06	20	Schleifen
5	18.01.07	3	Methoden
6	22.02.07	1	Einführung in die OOP mit BlueJ
7	29.03.07	2	Setter / Getter/ Konstruktoren, Trennung: Klasse / Anwendungsprogramm
8	24.05.07	7	Vererbung

Abbildung 40: Befragungstermine zu den einzelnen Modulen in der OOP-Later-Klasse

(Vorstudie, vgl. [ES07a, S. 3, Figure 5])

Im Kapitel 3.3.11 wurden schon alle Kontroll- und Synchronisations-Mechanismen (resultierend auch aus den Erfahrungen der Vorstudie) genannt, die helfen sollen, den Unterricht in den beiden Klassen trotz verschiedener Lehrer gleichartig zu gestalten.

Hierfür müssen auch die Lehrer gleichartiger werden. Die beiden Lehrer wurden in der Vorstudie von den beiden Klassen unterschiedlich wahrgenommen bzw. beurteilt (siehe Abbildung 96 in Kapitel 6.2.2). Der Durchschnittswert-Unterschied (OOP-First:  $\bar{O} = 1,8$  / OOP-Later:  $\bar{O} = 2,2$ ) beträgt immerhin 0,4 Punkte. Der OOP-Later-Lehrer erhält für die Hauptstudie den Auftrag, seinen Wert zu verbessern.

Der Vergleichstest wird neu strukturiert (siehe Kapitel 4.3.1). Der Vergleichstest der Vorstudie nahm noch Bezug auf einen Test mit 4 Themen, der am OSZ IMT schon vor der Studie am Ende des Schuljahres geschrieben wurde und bestand aus fünf Themen und einem Zusatzthema. Der Vergleichstest der Hauptstudie nimmt jetzt mehr Bezug zu den einzelnen unterrichteten Themen und besteht aus neun Themen und zwei Zusatzthemen (siehe Abbildung 56).

Auch der Fragebogen zum subjektiven Erleben wurde überarbeitet: Es gibt jetzt generell mehr Fragen (vgl. die Fragebögen in Anhang D und Anhang K), u. a. zum privaten und schulischen Stress und zur Motivation im Allgemeinen bzw. bezogen auf das Fach und mehr Fragen zum Lehrer bzw. zur Lernatmosphäre.

Ob die Klassen (nach statistischen Gesichtspunkten) wirklich gleich zusammengesetzt sind (im Hinblick auf ihre informatorische Leistungsfähigkeit) soll durch die erzielte Mathematik-Note am Ende des Schuljahres überprüft werden. Dies erfordert aber denselben Mathematiklehrer in den beiden Klassen und kann für die Hauptstudie auch tatsächlich realisiert werden [ES09a, S. 17].

Die Änderung des Begriffs Moduls ist zwingend erforderlich. Module sind gleich. Die Themen bei OOP-First und OOP-Later sind aber durch die unterschiedliche sequenzielle Vorgehensweise nicht 100% deckungsgleich. Z. B. sind beim Thema T1 „Einführung in die OOP“ in der OOP-First-Klasse die Begriffe Datentyp und Variable noch nicht bekannt, hingegen sind beim gleichen Thema T1 in der OOP-Later-Klasse diese Begriffe schon lange vorher bei dem Thema T2 „Einführung in den Java-Editor, Variablen, fundamentale Datentypen und Steuerstruktur Sequenz“ unterrichtet worden (vgl. Abbildung 37).

Die Änderungen im Design der Studie von Vorstudie zu Hauptstudie sind also wie folgt:

- Die Synchronisation der Lehrer bzw. des Unterrichts wird deutlich erhöht.
- Es gibt deutlich verbesserte Absprachen zwischen den Lehrern.
- Der OOP-Later-Lehrer soll in seiner Klasse (wenn möglich) höhere Sympathiewerte erzielen.
- Bei den Fragebögen gibt es mehr Fragen zum subjektiven Erleben (z. B. zum Stress und zur Motivation, aber auch zu den Lehrern und zur Lernatmosphäre) der Schülerinnen und Schüler.
- Die Fragebögen werden genau neunmal in die Klassen gegeben.
- Die Arbeitsblätter und die Übungen ähneln sich und gleichen sich zum Schuljahresende immer mehr an.
- Statt Module gibt es jetzt Themen, die Themen werden von 1 bis 9 durchnummeriert.
- Die Themen werden exakt definiert, sind gleichartig und werden gleich lang unterrichtet.
- Der Vergleichstest (Post-Test) wird völlig neu konzipiert, alle Themen werden beim Vergleichstest abgefragt.
- Es wird zusätzlich ein Nachhaltigkeitstest (Follow-Up-Test) geschrieben.
- Es soll ein Vortest (Pre-Test) realisiert werden (vgl. nächstes Kapitel).
- In beiden Klassen wird derselbe Mathematiklehrer eingesetzt.

### 3.3.15 Problemfelder: Lehrer, Vortest und fehlende Schüler

Es gibt beim Studiendesign 3 Problemfelder. Das erste Problemfeld ist wie folgt:

1. Die beiden Klassen werden von zwei unterschiedlichen Lehrern unterrichtet. Wie kann das Problem der Sicherstellung der Gleichartigkeit der Lehrer bzw. des Unterrichts gelöst bzw. reduziert werden?

Lösung: Es werden Maßnahmen (z. B. Synchronisation der Lehrer bzw. des Unterrichts) ergriffen, um das Problem deutlich zu verringern. Sie wurden schon ausführlich in den Kapiteln 3.3.10 und 3.3.11 vorgestellt und erläutert. Hier eine Zusammenfassung [ES09a]:

„Regarding the concrete teaching environment, several measures were taken. Teaching time, exams, programming language and tools were the same. The individual assignments, homework tasks and examples used had to differ, though. In order to measure effects of different ‘teaching-sequences’ we wanted to be able to make use of prior knowledge gained in each group. Therefore, e.g. an example used in topic X should of course make use of and integrate knowledge learned in prior topics, and these were different for the two groups. However, the teaching style itself should be the same. That is for example the role of examples, the complexity of assignments and so on. Therefore, both teachers in each course had regular meetings to coordinate their teaching style and exchange used materials.

Of course, maybe the different teaching persons have affected the study – in order to control this, one measure taken were the regular meetings, another was to include questions regarding the teacher into the questionnaires the students answered after each module.” [ES09a, S. 17-18].

Es wird also zweigleisig verfahren. Zum einen wird durch verstärkte Vorgaben für die Lehrer und durch verstärkte Absprachen zwischen den Lehrern die Gleichartigkeit des Unterrichts gewährleistet bzw. verbessert. Zum anderen wird die Verschiedenheit bzw. Gleichartigkeit der Lehrer bzw. ihres Unterrichts über Fragen an die Schülerinnen und Schüler ermittelt.

Im Kapitel 7.2.1 wird nachgefragt, inwieweit die Lehrer gleichartig waren bzw. als gleichartig erlebt wurden.

Das zweite Problemfeld beim Studiendesign ist wie folgt:

2. Wie kann ein Vortest aussehen, ohne die Schüler zu frustrieren, da diese so gut wie keine Programmier-Vorkenntnisse mitbringen (siehe Kapitel 3.3.6)? Im Studiendesign (vgl. Abbildung 24) sind ein Vortest, ein Vergleichstest und ein Nachhaltigkeitstest vorgesehen. Wie kann aber ein Test am Anfang des Schuljahres geschrieben werden, der ähnlich geartet ist wie der Vergleichstest (um den Lernzuwachs zu protokollieren), ohne dass durch die zu erwartende Überforderung der Schüler der Test (negativen) Einfluss auf das weitere Schuljahr und damit auf die Untersuchungsergebnisse hat?

Lösung: Im Kapitel 4.3.3 wird begründet, warum der stark reduzierte Vergleichstest als Vortest in einer anderen Lerngruppe, nämlich in den Schulklassen ein Jahrgang später geschrieben wird [ES09a]:

„The pre-test was a small sub-sample of these questions. However, as the question of OOP-first versus OOP-later is focused on introductory programming, we did not expect any prior knowledge. Instead, giving students a big test at the beginning in which they are likely to fail completely could lead to negative attitudes. Therefore, we decided to not give the pre-test to the students involved in the study; instead, we provided this test to another

group of students not involved in the study. In addition, we thought it is reliable enough to assume equality of the students regarding prior knowledge.“  
[ES09a, S. 17].

Im Kapitel 7.2.2 wird nachgefragt, ob der Vortest die Eingangsvoraussetzung bestätigt, dass die Schüler ohne bzw. mit ganz wenig Programmier-Vorkenntnissen an das OSZ IMT kommen.

Das dritte Problemfeld beim Studiendesign ist wie folgt:

3. Wie verfährt man mit dem Problem von fehlenden Schülern bei den Umfragen oder beim Vergleichstest bzw. Nachhaltigkeitstest?

Das Problemfeld gliedert sich wie folgt auf:

- Ein Schüler ist in den Unterrichtsstunden zu einem bestimmten Thema anwesend, fehlt aber bei der abschließenden Umfrage.
- Ein Schüler fehlt oft beim Unterricht eines Themas, ist aber bei der Umfrage anwesend.
- Ein Schüler ist häufiger nicht anwesend bzw. verlässt die Schule.
- Ein Schüler fehlt beim Vergleichstest oder Nachhaltigkeitstest.
- Ein Schüler verlässt nach dem Vergleichstest die Schule.

Lösung: Es müssen Vorgaben gemacht werden. Diese sind wie folgt:

- Bei den neun Umfragen wird nicht nachgeschrieben. Die Anwesenheit beim Unterricht spielt keine Rolle. Begründung: Der bürokratische Aufwand wäre zu hoch.
- Es werden Cut-off-Kriterien für die Umfragen festgelegt. Schüler, die weniger als an 2/3 (aber an mehr als 1/3) aller Umfragen teilgenommen haben, werden beim Vergleichstest nicht berücksichtigt. Schüler, die nur maximal an 1/3 aller Umfragen teilgenommen haben, werden aus den Umfragen herausgenommen (siehe Abbildung 41).

Anzahl der Umfragen	Werden die Daten der Umfragen gewertet?	Zugelassen zum Vergleichstest?
6-9	ja	ja
4-5	ja	nein
1-3	nein	nein

Abbildung 41: Cut-off-Kriterien für die Umfragen

Begründung: Geringe Anwesenheit hat in der Regel Auswirkungen auf den Lernerfolg. Gemessen werden soll aber nicht der Lernerfolg in Abhängigkeit von der Anwesenheit sondern der Lernerfolg in Abhängigkeit von der fachdidaktischen Vorgehensweise (vgl. [Sei04, S. 144]). Dies gilt auch für die Wiederholer in den beiden Klassen. Zwar werden ihre Umfragen gewertet, nicht aber ihre Tests, da sie Vorwissen aus einem früheren Schuljahr mit einem OOP-First-Vorgehen mitbringen.

- Es werden Cut-off-Kriterien für die Tests festgelegt. Fehlende Schüler beim Vergleichstest bzw. Nachhaltigkeitstest müssen nachschreiben. Ist dies zeitnah nicht möglich, werden die Daten nicht berücksichtigt. Schüler, die die Schule nach dem Vergleichstest verlassen, werden ebenfalls nicht mehr

berücksichtigt. Es werden somit nur die Daten der Schüler gewertet, die an beiden Tests teilnehmen (siehe Abbildung 42).

Zum Vergleichstests zugelassen?	Vergleichstest mitgeschrieben?	Nachhaltigkeitstest mitgeschrieben?	Daten der Tests werden gewertet?
ja	ja	ja	ja
ja	ja	nein	nein
ja	nein	ja	nein
ja	nein	nein	nein

Abbildung 42: Cut-off-Kriterien für die Tests

Begründung: Die Daten müssen vollständig sein. Die „Schülermenge“ bei Nachhaltigkeitstest und Vergleichstest muss dieselbe sein. Nur dann können die Ergebnisse beider Tests seriös miteinander verglichen werden.

Im Kapitel 7.2.3 wird nachgefragt, ob es Probleme in den beiden Klassen mit fehlenden Schülern gab. Und ob die Vorgaben Auswirkungen auf die Datenanalyse hatten, speziell auf die „Schülermenge“ bei dem Vergleichs- und dem Nachhaltigkeitstest.

### 3.4 Zusammenfassung

Das Ziel der empirischen Studie (siehe Kapitel 3.1) ist es, die beiden Forschungsfragen zu beantworten:

- Erzielt das OOP-First-Vorgehen oder das OOP-Later-Vorgehen bessere Lernergebnisse (unterteilt in einzelne Themengebiete)?
- Ist das subjektive Erleben der Schüler positiver beim OOP-First-Vorgehen oder beim OOP-Later-Vorgehen (unterteilt in einzelne Items)?

Hierfür wurde im Kapitel 3.2 als Forschungsdesign ein (Entscheidungs-) Experiment gewählt, welches sich in die fünf Phasen gliedert: Definitions-, Design-, Feld-, Analyse- und Kommunikationsphase.

Das Studiendesign (vgl. Abbildung 24) wurde dann im Kontext zum Forschungsdesign konkret implementiert. Hierfür wurden im Kapitel 3.3 die folgenden Sachverhalte und Randbedingungen berücksichtigt bzw. diskutiert:

- Die konkreten Randbedingungen am OSZ IMT (siehe Kapitel 3.3.1 und 3.3.2).
- Die konkreten zeitlichen Dimensionen im Hinblick auf die Schuljahre bzw. die Vor- und Hauptstudie (siehe Kapitel 3.3.2, 3.3.3, 3.3.4 und 3.3.14).
- Die konkreten Schüler und Lehrer (siehe Kapitel 3.3.6, 3.3.7, 3.3.10, 3.3.11 und 3.3.15).
- Die konkreten Themen, Unterrichts-Methoden, Arbeitsmittel und Medien (siehe Kapitel 3.3.8 und 3.3.9).
- Die konkreten Tests (siehe Kapitel 3.3.2, 3.3.12 und 3.3.15).
- Die konkreten Fragebögen (siehe Kapitel 3.3.2 und 3.3.13).

In den nächsten beiden Kapiteln 4 und 5 werden nun die Tests und die Fragebögen ausführlich entwickelt.



## **4 Lernerfolg: Lernziele, Indikatoren und Tests**

Im gesamten Kapitel 4 wird die Datensammlung zum objektiven Aspekt dieser Studie vorbereitet.

Hierfür werden in Kapitel 4.1 für alle neun Themen die Lernziele bzw. Kompetenzen formuliert. Am Beginn der Studie waren die Rahmenlehrpläne noch lernzielorientiert. Daher finden sich in den nächsten Kapiteln eher Lernziel- als Kompetenzformulierungen. Dabei wird Bezug genommen auf den damals am OSZ IMT gültigen Arbeitsplan für den Programmierunterricht in der Einführungsphase des Beruflichen Gymnasiums (siehe Anhang B).

Zusätzlich zu den neun Themen werden zwei weitere Kategorien mit themenübergreifenden Fachkompetenzen abgeleitet (siehe Kapitel 4.2).

Die Überprüfung, inwieweit die Lernziele erreicht wurden, geschieht mit Hilfe einzelner Fragen (Indikatoren), die aufsummiert den Vergleichstest (Post-Test) ergeben (siehe Kapitel 4.3.1 und Anhang M). Im Kapitel 3.3.12 wurde begründet, warum es drei verschiedene Tests geben soll (inkl. der zeitlichen Festlegungen im Kontext zum Studiendesign) und welche Vorgaben für die Tests bzw. die Testfragen gelten sollen. Z. B. sollen sich objektorientierte Fragestellungen im Gleichgewicht zu prozeduralen Fragestellungen befinden und soll der Multiple-Choice-Anteil bei den Fragen nicht über 30% liegen. Für den Test müssen den Fragen Wertungspunkte und Schwierigkeitsgrade (also Anforderungsbereiche) zugeordnet werden. Jedes Thema soll die gleiche Summe von Wertungspunkten haben und etwa das gleiche durchschnittliche Schwierigkeitsniveau.

Dieses Kapitel (wie fast alle Kapitel) legt den Schwerpunkt auf die Betrachtung der Hauptstudie. Natürlich gab es aber schon bei den Themen der Vorstudie Lernzielformulierungen und abgeleitete Indikatoren. Somit sind einige nachfolgende Lernzielformulierungen inkl. der Indikatoren nicht völlig neu. Genauere Informationen finden sich im Kapitel 4.3.1 („Vergleichstest (Post-Test) in der Vorstudie und in der Hauptstudie“) wieder.

Nach dem Vergleichstest werden dann der Nachhaltigkeitstest (Follow-Up-Test) und der Vortest (Pre-Test) entwickelt (siehe Kapitel 4.3.3 und 4.3.4). Aus diesen Tests sollen Daten gewonnen werden, deren Auswertung (siehe Kapitel 6) die erste Forschungsfrage (siehe Kapitel 2.4.1) beantworten soll.

Die Testentwicklung wird am Ende kritisch reflektiert (siehe Kapitel 4.3.5).

### **4.1 Beschreibung der Lernziele und Ableitung der Indikatoren**

Im Folgenden werden zu jedem der neun Themen die einzelnen Lernziele bzw. Fachkompetenzen entwickelt werden. Darauf aufbauend werden die dazugehörigen Indikatoren genannt, die sich dann als Fragen im Vergleichstest wiederfinden. Zu jeder Fachkompetenz könnte eine Vielzahl von möglichen Indikatoren genannt werden (was auch im Sinne der allgemeinen Testtheorie ist). Damit der Vergleichstest sich aber in einer schulisch vertretbaren Länge befindet, wird im Folgenden oft nur ein Indikator pro Fachkompetenz angeführt. Manchmal deckt dieser aber dann nicht den gesamten Bereich der angeführten Fachkompetenz ab. Die Beschreibung der Lernziele und die Ableitung der Indikatoren erfolgte in Absprache mit dem beteiligten Lehrer, um keine der beiden Klassen zu bevorzugen oder zu benachteiligen.

Exemplarisch sollen nun die Lernziele für das Thema T2 entwickelt werden und die Indikatoren genannt werden. Alle weiteren Themen werden mit Hilfe von Abbildungen verkürzt dargestellt. Das Thema T2 lautet: „Einführung in den Java-Editor, Variablen (Attribute), fundamentale Datentypen, Steuerstruktur Sequenz (inkl. Struktogramm)“. In diesem Thema sind schon die einzelnen Fachkompetenzen bzw. Feinlernziele (im Folgenden mit einem ‚F‘ benannt und durchnummeriert) impliziert:

- Einführung in den Java-Editor: Die Schüler starten den Java-Editor, editieren kleine Programme und speichern diese ab (F1).
- Variablen: Die Schüler erläutern den Unterschied zwischen Variablen und Konstanten (F3). Sie deklarieren Variable, ordnen den Variablen sinnvolle Werte zu und benutzen die Java-Syntax (F5).
- Fundamentale Datentypen: Die Schüler erläutern den Begriff Datentyp (F2). Sie nennen einzelne fundamentale Datentypen, ordnen ihnen die Anzahl der Bytes bzw. die möglichen Werte zu und benutzen die Datentypen in speziellen Anwendungsfällen (F4).
- Steuerstruktur Sequenz (inkl. Struktogramm): Die Schüler schreiben kleine sequenzielle Programme (EVA-Prinzip) und zeichnen die dazugehörigen Struktogramme (F6).

Zu jeder Fachkompetenz bzw. zu jedem Feinlernziel muss nun ein Indikator gefunden werden:

- Indikator zu F1: Hier gibt es ein Problem, da die Aufgaben auf Papier gelöst werden sollen. Daher lässt sich hier kein Indikator I1 definieren.
- Indikator I3 zu F3: Die Schüler bekommen zwei Quelltextzeilen

```
final double PI = 3; PI = 3.14;
```

und werden per Multiple-Choice gefragt, welche Aussage(n) zutrifft:

- Compilerfehler in Zeile 1
  - Compilerfehler in Zeile 2
  - PI ist eine Variable
  - PI muss klein geschrieben werden
  - 3,14 statt 3.14
- Indikator I5 zu F5: Die Schüler bekommen drei lückenhafte Quelltextzeilen

```
char a = _____; _____ b = 7.0f; _____ c = (5 == 7);
```

und sollen den Quelltext ergänzen.
  - Indikator I2 zu F2: Die Schüler sollen bestimmen, dass durch den Datentyp einer Variable festgelegt wird, welche(n) ...
    - Werte die Variable annehmen kann
    - Namen die Variable hat
    - Adresse die Variable im Speicher hat
    - Wert die Variable hat
    - Anzahl von Bits die Variable im Speicher belegt

- Indikatoren I4.1 und I4.2 zu F4:

Die Schüler werden gefragt, welche Werte eine Variable vom Datentyp `byte` annehmen kann:

- 0 bis 255
- 0 bis 65535
- 128 bis 127
- 127 bis 128
- Der Wert ist hardwareabhängig

Die Schüler bekommen zwei Quelltextzeilen

```
int a = 7; short b = a;
```

und werden per Multiple-Choice gefragt, welche Aussage(n) zutrifft:

- Es gibt einen Compilerfehler
- Beide Variablen benötigen den gleichen Speicherplatz
- a belegt 16 Bit
- b belegt 8 Bit

- Indikator I6 zu F6: Die Schüler bekommen drei Quelltextzeilen

```
int a, b; a = 7; b = 8;
```

und sollen das dazugehörige Struktogramm zeichnen.

Jeder Aufgabe müssen nun Punkte (**P**) zugeordnet werden. Die Summe der Punkte pro Thema soll den Wert 10 ergeben, so dass die Summe der Punkte des Vergleichstests ca. 100 ist, was z. B. im Berliner Informatik-Abitur die Regel ist. Auch muss ein Anforderungsbereich (**AF**) angegeben werden, damit später kontrolliert werden kann, ob alle Themen mit dem gleichen Schwierigkeitsgrad abgefragt werden.

Die Zuordnung der Aufgaben zu so genannten Anforderungsbereichen orientiert sich dabei an den „Einheitlichen Prüfungsanforderungen Informatik“ der Kultusministerkonferenz [Kmk04], bei der jede (Abitur-) Aufgabe mindestens einem von drei so genannten Anforderungsbereichen zugeordnet werden muss:

„Der Anforderungsbereich I umfasst

- die Wiedergabe von bekannten Sachverhalten aus einem abgegrenzten Gebiet im gelernten Zusammenhang [...]

Der Anforderungsbereich II umfasst

- die selbstständige Verwendung (Auswählen, Anordnen, Verarbeiten und Darstellen) bekannter Sachverhalte zur Bearbeitung neuer Frage- oder Problemstellungen [...]

Der Anforderungsbereich III umfasst

- das [sic!] planmäßige Verarbeitung komplexer Gegebenheiten mit dem Ziel, zu selbstständigen Gestaltungen bzw. Deutungen, Folgerungen, Begründungen, Wertungen zu gelangen [...]“ [Kmk04, S. 10-11].

Weiter heißt es:

„Die Prüfungsaufgabe sowohl für das Grundkursfach als auch für das Leistungskursfach erreicht dann ein angemessenes Niveau, wenn das Schwergewicht der zu erbringenden Prüfungsleistungen im Anforderungsbereich II liegt und daneben die Anforderungsbereiche I und III berücksichtigt werden, und zwar Anforderungsbereich I in höherem Maße als Anforderungsbereich III.“ [Kmk04, S. 12].

Die Anzahl der Punkte orientiert sich in der Regel nach dem Anforderungsbereich und danach, ob es eine Multiple-Choice-Aufgabe ist oder nicht. Aufgaben im Anforderungsbereich I bzw. Multiple-Choice-Aufgaben erhalten in der Regel nur einen Punkt. Aufgaben in den Anforderungsbereichen II und III bzw. Aufgaben, die nicht zum Ankreuzen sind, erhalten in der Regel mehr als einen Punkt. Somit könnten die Punkte für die Aufgaben zum Thema T2 so vergeben werden, wie sie die Abbildung 43 zeigt.

Indikator	Multiple-Choice	Anforderungsbereich	Punkte
I2	ja	I	1
I3	ja	I	1
I4.1	ja	I	1
I4.2	ja	II	1
I5	nein	II	3
I6	nein	II	3

Abbildung 43: Anforderungsbereiche und Punkteverteilung für die Indikatoren des Themas T2

Zur Korrektur bedarf es zusätzlich noch einer Lösung (evtl. mit der Vergabe von Teilpunkten):

- Lösung zu I3 (1P):

**X** Compilerfehler in Zeile 2

- Lösung zu I5 (3 \* 1P):

```
char a = 'a'; float b = 7.0f; boolean c = (5 == 7);
```

- Lösung zu I2 (2 \* 0,5P):

**X** Werte die Variable annehmen kann

**X** Anzahl von Bits die Variable im Speicher belegt

- Lösung zu I4.1 (1P):

**X** -128 bis 127

- Lösung zu I4.2 (1P):
  - ✗ Es gibt einen Compilerfehler
- Lösung zu I6 (2 \* 1,5P):

a ← 7
b ← 8

Alle Informationen können verkürzt in einer Abbildung zusammengefasst werden, für das Thema T2 ist dies die Abbildung 44. Der Kürze wegen werden die Antworten gleich mit der Lösung dargestellt. **T** steht für Thema, **F** für Fachkompetenz bzw. Feinlernziel, **VT** für Vergleichstest-Aufgabennummer, **I** für Indikator bzw. Aufgabe und **L** für Lösung. Die Hintergrundfarbe der Abbildung unterstützt die Einordnung der Information, z. B. weist die Farbe grau auf Feinlernziele bzw. Fachkompetenzen hin, die Farbe gelb auf Indikatoren und die Farbe grün auf die Lösung.

Für das Thema T2 wurden die Definition der Fachkompetenzen bzw. Feinlernziele, die Angabe der Indikatoren, die Zuordnung der Aufgaben zu Anforderungsbereichen, die Nennung der Lösung und die Zuteilung der Punkte zu den (Teil-) Aufgaben exemplarisch vorgenommen.

Alle weiteren Themen werden im Folgenden nur mit Hilfe der verkürzten Abbildungen veranschaulicht:

- Thema T5 (Selektion): Abbildung 45
- Thema T6 (Iteration): Abbildung 46
- Thema T7 (Arrays und Strings): Abbildung 47
- Thema T4 (Methoden): Abbildung 48
- Themen T1 und T3 (Einführung in die OOP / Klasse und Objekt): Abbildung 49
- Thema T8 (Vererbung): Abbildung 50
- Thema T9 (Assoziation): Abbildung 51

Die Themen 1 und 3 werden zwar getrennt unterrichtet, aber für den Vergleichstest (siehe nächstes Kapitel) zusammengezogen, da bei einem schriftlichen Test (auf Papier, ohne BlueJ-Tool) die unterrichteten Teilthemen doch stark miteinander verwoben sind.

<b>T2</b>	<b>Einführung in den Java-Editor, Variablen (Attribute), fundamentale Datentypen, Steuerstruktur Sequenz (inkl. Struktogramm)</b>	<b>VT1</b>
F1	<b>Die Schüler starten den Java-Editor, editieren kleine Programme und speichern diese ab.</b>	
I1/L1	Wird nicht abgefragt, da Vergleichstest auf Papier geschrieben wird	
F2	<b>Die Schüler erläutern den Begriff Datentyp.</b>	
I2	Durch den Datentyp einer Variablen wird festgelegt, welche(n) ...	AFI
L2	<input checked="" type="checkbox"/> Werte die Variable annehmen kann <input type="checkbox"/> Namen die Variable hat <input type="checkbox"/> Adresse die Variable im Speicher hat <input type="checkbox"/> Wert die Variable hat <input checked="" type="checkbox"/> Anzahl von Bits die Variable im Speicher belegt	1P
F3	<b>Die Schüler erläutern den Unterschied zwischen Variablen und Konstanten.</b>	
I3	Gegeben sind die beiden Quelltextzeilen: <pre>final double PI = 3; PI = 3.14;</pre> Welche Aussage(n) trifft/treffen zu?	AFI
L3	<input type="checkbox"/> Compilerfehler in Zeile 1 <input checked="" type="checkbox"/> Compilerfehler in Zeile 2 <input type="checkbox"/> PI ist eine Variable <input type="checkbox"/> PI muss klein geschrieben werden <input type="checkbox"/> 3,14 statt 3.14	1P
F4	<b>Die Schüler nennen einzelne fundamentale Datentypen, ordnen ihnen die Anzahl der Bytes bzw. möglichen Werte zu und benutzen die Datentypen in speziellen Anwendungsfällen.</b>	
I4.1	Welche Werte kann eine Variable vom Datentyp <code>byte</code> annehmen?	AFI
L4.1	<input type="checkbox"/> 0 bis 255 <input type="checkbox"/> 0 bis 65535 <input checked="" type="checkbox"/> -128 bis 127 <input type="checkbox"/> -127 bis 128 <input type="checkbox"/> Der Wert ist hardwareabhängig	1P
I4.2	Gegeben sind die beiden Quelltextzeilen: <pre>int a = 7; short b = a;</pre> Welche Aussage(n) trifft/treffen zu?	AFII
L4.2	<input checked="" type="checkbox"/> Es gibt einen Compilerfehler <input type="checkbox"/> Beide Variablen benötigen den gleichen Speicherplatz <input type="checkbox"/> a belegt 16 Bit <input type="checkbox"/> b belegt 8 Bit	1P
F5	<b>Die Schüler deklarieren Variablen, ordnen den Variablen sinnvolle Werte zu und benutzen die Java-Syntax.</b>	
I5	Gegeben ist die folgende lückenhafte Variablendeklaration: <pre>char a = ____; ____ b = 7.0f; ____ c = (5 == 7);</pre> Ergänzen Sie den Quelltext in den 3 Lücken!	AFII
L5	<code>char a = 'a'; float b = 7.0f; boolean c = (5 == 7);</code>	3P
F6	<b>Die Schüler schreiben kleine sequenzielle Programme (EVA-Prinzip) und zeichnen die dazugehörigen Struktogramme.</b>	
I6	Gegeben sind die drei Quelltextzeilen: <pre>int a, b; a = 7; b = 8;</pre> Zeichnen Sie das dazugehörige Struktogramm!	AFII
L6	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <pre>a ← 7 b ← 8</pre> </div> Hinweis: Variablendeklarationen werden in einem Struktogramm nicht angegeben	3P

Abbildung 44: Lernziele und Indikatoren für das Thema 2 (Java-Editor, Attribute, Datentypen, Sequenz)

T5 Steuerstruktur Selektion (inkl. Struktogramm)		VT2								
F1	<b>Die Schüler implementieren kleine Algorithmen mit Selektionen. Sie schreiben hierfür Quelltext im Java-Editor, überprüfen die Syntax durch Kompilation und testen das Programm zur Laufzeit.</b>									
I1/L1	Wird nicht abgefragt, da Vergleichstest auf Papier geschrieben wird									
F2	<b>Die Schüler erläutern den Begriff einseitige und zweiseitige Auswahl und realisieren diese inkl. der Bedingung im spez. Java-Quellcode.</b>									
I2	Gegeben ist der folgende Vergleich: <pre>if (wahrheit == false) {}</pre> Die verkürzte(n) Schreibweise(n) ist/sind wie folgt:	AFI								
L2	<input type="radio"/> if (wahrheit) {} <input type="radio"/> if (wahrheit <> true) {} <input checked="" type="radio"/> if (!wahrheit) {} <input type="radio"/> if (wahrheit!) {}	1P								
F3	<b>Die Schüler schreiben und analysieren (Schreibtischtest) Quelltext mit verschachtelter Auswahl.</b>									
I3	Gegeben ist der folgende Quelltextauschnitt: <pre>int b = 0, c = 7; if (c &lt; 7)     if (c &gt;= 7) b = 1; else b = 2; else     if (c &gt; 7) b = 3; else b = 4;</pre> Was steht am Ende in der Variablen b?	AFII								
L3	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4	1P								
F4	<b>Die Schüler schreiben und analysieren (Schreibtischtest) Quelltext mit Fallunterscheidung.</b>									
I4/I5	Gegeben ist der folgende Quelltextauschnitt: <pre>int a = 5;                      // oder 7 oder 9 switch (a) {     case 5:    a = 3 * a;    break;     case 9:    a = a / 2;    break;     default:   a = a % 3;    break; } // switch if (a &gt;= 15) a--; else a++; System.out.println("a: " + a);</pre>									
I4	Welcher Wert wird ausgegeben, bei den folgenden ursprünglichen Werten für die Variable a?	AFII								
L4	<table border="1"> <thead> <tr> <th>a</th> <th>Ausgabe</th> </tr> </thead> <tbody> <tr> <td>5</td> <td><b>14</b></td> </tr> <tr> <td>7</td> <td><b>2</b></td> </tr> <tr> <td>9</td> <td><b>5</b></td> </tr> </tbody> </table>	a	Ausgabe	5	<b>14</b>	7	<b>2</b>	9	<b>5</b>	4P
a	Ausgabe									
5	<b>14</b>									
7	<b>2</b>									
9	<b>5</b>									
F5	<b>Die Schüler schreiben Algorithmen mit Selektionen und zeichnen die dazugehörigen Struktogramme.</b>									
I5	Zeichnen Sie das dazugehörige Struktogramm!	AFII								
L5		4P								

Abbildung 45: Lernziele und Indikatoren für das Thema 5 (Selektion)

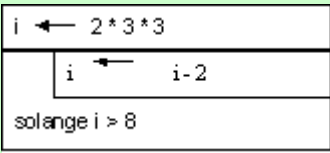
T6	Steuerstruktur Iteration (inkl. Struktogramm)	VT3
F1	<b>Die Schüler implementieren kleine Algorithmen mit Iterationen. Sie schreiben hierfür Quelltext im Java-Editor, überprüfen die Syntax durch Kompilation und testen das Programm zur Laufzeit.</b>	
I1/L1	Wird nicht abgefragt, da Vergleichstest auf Papier geschrieben wird	
F2	<b>Die Schüler erläutern den Unterschied von kopfgesteuerten, fußgesteuerten, Zähl- und Endlosschleifen (auch im Hinblick auf Struktogramme und spez. Java-Syntax).</b>	
I2.1	Gegeben ist die folgende Quelltextzeile: <pre>for (int i = 1; i &lt; 7; i++) {}</pre> Diese Schleife wird im Struktogramm mit Hilfe welchen Strukturelements nach DIN 66261 dargestellt?	AFI
L2.1	<input checked="" type="checkbox"/> kopfgesteuerte Schleife <input type="checkbox"/> fußgesteuerte Schleife <input type="checkbox"/> Endlosschleife <input type="checkbox"/> keiner der genannten Schleifen	1P
I2.2	Welche Antwort(en) enthält/enthalten eine Endlosschleife?	AFI
L2.2	<input checked="" type="checkbox"/> for ( ; ; ) { } <input checked="" type="checkbox"/> do { } while (!false); <input type="checkbox"/> while (!true) { } <input checked="" type="checkbox"/> int y = 7; while (y == 7) { }	1P
F3	<b>Die Schüler schreiben und analysieren (Schreibtischtest) Quelltext mit einer Zählschleife.</b>	
I3	Ergänzen Sie den folgenden Quelltext, so dass als Ausgabe: 10,8,6,4, erscheint!	AFII
L3	<pre>for (int z = 10; z &gt;= 4; z = z - 2)     System.out.print(z + ',');</pre>	2P
F4	<b>Die Schüler schreiben und analysieren (Schreibtischtest) Quelltext mit einer kopfgesteuerten Schleife.</b>	
I4	Gegeben ist der folgende Quelltextauschnitt: <pre>a = 3 * 8; b = a / 4; while (b &gt; 1) {     a = a - 2; b = b - 1; }</pre> Welcher Wert steht am Ende des Algorithmus in der Variable a?	AFII
L4	a = 14	2P
F5	<b>Die Schüler schreiben und analysieren (Schreibtischtest) Quelltext mit einer fußgesteuerten Schleife.</b>	
I5 / I6	Gegeben ist der folgende Quelltextauschnitt: <pre>i = (2 * 3) * 3; do {     i = i - 2; } while (i &gt; 8);</pre>	
I5	Wie oft wird der Schleifenkörper durchlaufen?	AFII
L5	Durchläufe = 5	2P
F6	<b>Die Schüler schreiben Algorithmen mit Iterationen und zeichnen die dazugehörigen Struktogramme inkl. Endlosschleife.</b>	
I6	Zeichnen Sie das dazugehörige Struktogramm!	AFII
L6		2P

Abbildung 46: Lernziele und Indikatoren für das Thema 6 (Iteration)



T7	Arrays und Strings	VT4
F1	<b>Die Schüler implementieren eindimensionale Zahlen- und Zeichenfelder. Sie schreiben hierfür Quelltext im Java-Editor, überprüfen die Syntax durch Kompilation und testen das Array zur Laufzeit.</b>	
I1/L1	Wird nicht abgefragt, da der Vergleichstest auf Papier geschrieben wird	
F2	<b>Die Schüler erläutern das Prinzip von Feldern (auch im Kontext zur Java-VM).</b>	
I2	<code>int[] x = new int[25];</code> Welche Aussage(n) trifft/treffen nach der Ausführung der Programmzeile zu?	AFI
L2	<input type="radio"/> <code>x.length</code> hat den Wert 24 <input type="radio"/> <code>x[0]</code> hat den Wert null <input type="radio"/> <code>x[25]</code> hat den Wert 0 <input type="radio"/> <code>x[24]</code> ist nicht definiert <input checked="" type="radio"/> keine Aussage trifft zu	1P
F3	<b>Die Schüler durchlaufen ein Array mit Hilfe einer Zählschleife.</b>	
I3	Initialisieren Sie das folgende Feld mit Hilfe einer Zählschleife mit den Werten 15, 35 und 55! <code>short[] zahlenfeld = new short[3];</code>	AFII
L3	<code>for (int i = 0; i &lt;= 2; i++)</code> <code>zahlenfeld[i] = 15 + i*20;</code>	3P
F4	<b>Die Schüler analysieren vorhandene Algorithmen, die mit einem Feld arbeiten auf Laufzeitverhalten (inkl. Ausnahmen).</b>	
I4	Gegeben ist der folgende Quelltextausschnitt: <code>int[] exampleArray = {10, 20, 30, 40};</code> <code>System.out.println(exampleArray[2*2]*3);</code> Wie lautet die Bildschirmausgabe?	AFII
L4	(ArrayIndexOutOfBoundsException-) Exception	2P
F5	<b>Die Schüler erzeugen Objekte der Klasse String. Sie testen das Verhalten dieser Objekte zur Laufzeit mit Hilfe kurzer Testprogramme.</b>	
I5/L5	Wird nicht abgefragt, da der Vergleichstest auf Papier geschrieben wird	
F6	<b>Die Schüler erläutern den Unterschied zwischen Literal-String- und String-Objekten und unterscheiden zwischen Referenzvergleich (==) und Inhaltsvergleich (equals).</b>	
I6	Gegeben ist die folgende Variablendeklaration: <code>String s = "Test"; String t = new String("Test");</code> Welche Bedingungen liefern true zurück?	AFII
L6	<input checked="" type="checkbox"/> <code>if (s == "Test") {}</code> <input type="checkbox"/> <code>if (t == s) {}</code> <input checked="" type="checkbox"/> <code>if (s.equals("Test")) {}</code> <input checked="" type="checkbox"/> <code>if (t.equals(s)) {}</code>	1P
F7	<b>Die Schüler wenden einzelne Methoden auf String-Objekten an und unterscheiden die konstanten Stringwerte von den Rückgabewerten der Methoden.</b>	
I7	Gegeben ist der folgende Quelltext: <code>String s = new String("HALLO"); s.toLowerCase();</code> <code>System.out.println(s);</code> <code>s = "kleine"; s += " Zwerge";</code> <code>System.out.println(s.toUpperCase());</code> <code>System.out.println(5*3 + 7 + " 5*3 Tore " + 3*5);</code> Wie lautet die Bildschirmausgabe?	AFII
L7	HALLO KLEINE ZWERGE 22 5*3 Tore 15	3P

Abbildung 47: Lernziele und Indikatoren für das Thema 7 (Arrays und Strings)

T4	Methoden	VT5
F1	<b>Die Schüler implementieren einfache Methoden. Sie schreiben hierfür Quelltext im Java-Editor insbesondere die Methodenköpfe, überprüfen die Syntax durch Kompilation und testen die Methoden inkl. der Parameterübergabe zur Laufzeit.</b>	
I1/L1	Wird nicht abgefragt, da der Vergleichstest auf Papier geschrieben wird	
F2	<b>Die Schüler erläutern das Prinzip von Methoden und unterscheiden zwischen aktuellem und formalen Parameter.</b>	
I2	Beim Aufruf einer Methode findet man nie ...	AFI
L2	<input type="radio"/> einen aktuellen Parameter <input checked="" type="radio"/> einen formalen Parameter <input type="radio"/> einen Rückgabewert	1P
F3	<b>Die Schüler unterscheiden zwischen statischen und nichtstatischen Methoden, Methoden mit und ohne Parametern und Methoden mit und ohne Rückgabewerten.</b>	
I3.1	Welche Aussage(n) trifft/treffen für die folgende Quelltextzeile zu: <pre>if (isGeldAlle()) {}</pre>	AFI
L3.1	<input type="radio"/> Die Methode hat keinen Rückgabewert <input type="radio"/> Der Methode fehlt ein Parameter <input checked="" type="radio"/> Die Methode hat einen Boolean-Rückgabewert	1P
I3.2	Gegeben ist der folgende Quelltextauszug: <pre>String s; s = liefereEtwas(4.7);</pre> Welche Aussage(n) ist/sind richtig?	AFI
L3.2	<input checked="" type="radio"/> Die Methode kann statisch sein <input checked="" type="radio"/> Die Methode hat einen Rückgabewert <input type="radio"/> s muss vom Datentyp double sein <input type="radio"/> Vor dem Wert 4.7 fehlt der Datentyp double	1P
F4	<b>Die Schüler schreiben (insbesondere Methodenköpfe) und analysieren Quelltext mit Methoden.</b>	
I4	Gegeben sind die folgenden Quelltextzeilen des Anwendungsprogramms, die in der main-Methode stehen: <pre>int[] zahlenfeld = new int[10]; initialisiereFeld(zahlenfeld); kleinsteZahl = ermittleKleinsteZahl(zahlenfeld);</pre> Wie lautet der Methodenkopf der statischen Methode „ermittleKleinsteZahl (...)“?	AFII
L4	<pre>public static int ermittleKleinsteZahl(int[] feld)</pre>	3P
F5	<b>Die Schüler entwickeln nach einer vorgegebenen Aufgabenstellung algorithmische Strukturen innerhalb einer Methode und implementieren diese.</b>	
I5	In der Klasse „Konto“ soll eine Methode „zahleEin“ implementiert werden, die als formalen Parameter einen Double-Geldbetrag hat und zurückliefert, ob das Konto nach der Einzahlung gedeckt ist (Attribut kontostand > 0 bedeutet: Rückgabewert ist „true“). Geben Sie den vollständigen Quelltext der Methode „zahleEin“ an!	AFII
L5	<pre>public boolean zahleEin(double geldbetrag) {     kontostand += geldbetrag;     if (kontostand &gt; 0)         return true;     return false; }</pre>	4P

Abbildung 48: Lernziele und Indikatoren für das Thema 4 (Methoden)

T1/T3	Einführung in die OOP / Klassenimplementierung und Objekterzeugung	VT6/7			
F1	<b>Die Schüler modellieren Klassen und implementieren diese inkl. Attribute, Methoden und Konstruktoren. Sie erzeugen Objekte einer Klasse und testen das Verhalten der Objekte zur Laufzeit.</b>				
I1/L1	Wird nicht abgefragt, da der Vergleichstest auf Papier geschrieben wird				
F2	<b>Die Schüler unterscheiden zwischen Klasse und Objekt.</b>				
I2.1	Welche Aussage(n) über Klassen sind falsch?	AFI			
L2.1	<input checked="" type="checkbox"/> Eine Klasse ist in einem bestimmten Zustand <input checked="" type="checkbox"/> Eine Klasse ist die Summe der Objekte <input checked="" type="checkbox"/> Eine Klasse hat immer Attribute	1P			
I2.2	Welche Aussage(n) über Objekte ist/sind richtig?	AFI			
L2.2	<input type="checkbox"/> Von jeder Klasse gibt es mindestens ein Objekt <input checked="" type="checkbox"/> Objekte können mit anderen Botschaften austauschen <input checked="" type="checkbox"/> Objekte befinden sich in einem Zustand <input type="checkbox"/> Objekte haben keinen Zugriff auf sich selbst	1P			
F3	<b>Die Schüler erläutern den Attribut-Begriff und implementieren so genannte „Setter und Getter“.</b>				
I3.1	Welche Aussage(n) zu Attributen ist/sind richtig?	AFI			
L3.1	<input checked="" type="checkbox"/> Attribute sind vergleichbar mit Variablen <input type="checkbox"/> In einer Klasse haben die Attribute Werte <input checked="" type="checkbox"/> Attribute werden in der Regel gekapselt <input type="checkbox"/> Attribute haben keinen Datentyp <input type="checkbox"/> Ein Objekt deklariert Attribute	1P			
I3.2	Wozu braucht man die Verwaltungsmethoden („Setter“ und „Getter“)?	AFI			
L3.2	<input type="checkbox"/> Weil diese immer deklariert werden müssen <input type="checkbox"/> Weil die Attribute in der Regel öffentlich sind <input checked="" type="checkbox"/> Weil die Attribute in der Regel privat sind	1P			
I3.3	Die Klasse „Reisebuero“ hat u. a. das Attribut „anzahlKunden“ vom Datentyp „int“. Wie lautet der vollständige Quelltext der „Setter“-Verwaltungsmethode?	AFII			
L3.3	<pre>public void setAnzahlKunden(int anzahlKunden) {this.anzahlKunden = anzahlKunden;}</pre>	3P			
I3.4	Die Klasse „Mensch“ hat das Attribut „vorname“ vom Datentyp „String“. Wie lautet der vollständige Quelltext der „Getter“-Verwaltungsmethode?	AFII			
L3.4	<pre>public String getVorname() { return this.vorname; }</pre>	2P			
F4	<b>Die Schüler erläutern und implementieren Konstruktoren.</b>				
I4.1	Wozu dienen Konstruktoren?	AFI			
L4.1	<input type="checkbox"/> Um eine Vererbungshierarchie zu ermöglichen <input type="checkbox"/> Um eine Klasse zu initialisieren <input checked="" type="checkbox"/> Um ein Objekt zu initialisieren <input type="checkbox"/> Um ein Klasse deklarieren zu können	1P			
I4.2	Die Klasse „Fahrrad“ hat die Attribute „groesse“ (Datentyp int) und „hersteller“ (Datentyp String). Wie lautet der Methodenkopf des vollparametrisierten Konstruktors?	AFII			
L4.2	<pre>public Fahrrad(int groesse, String hersteller)</pre>	2P			
F5	<b>Die Schüler erläutern und zeichnen UML-Klassendiagramme und UML-Objektdiagramme.</b>				
I5.1	Methoden werden im UML-Objektdiagramm ...	AFI			
L5.1	<input type="checkbox"/> angegeben <input checked="" type="checkbox"/> nicht angegeben <input type="checkbox"/> je nach Bedarf angegeben	1P			
I5.2	Die Klasse „Surfbrett“ hat das Attribut „laenge“ (Datentyp double). Zeichnen Sie das vollständige UML-OOO-Klassendiagramm inklusive Konstruktoren und allen Verwaltungsoperationen („Setter“ und „Getter“)!	AFII			
L5.2	<table border="1" style="display: inline-table; vertical-align: top;"> <tr> <td style="text-align: center;"><b>Surfbrett</b></td> </tr> <tr> <td>- laenge: double</td> </tr> <tr> <td>+ Surfbrett() + Surfbrett(double) + setLaenge(double): void + getLaenge(): double</td> </tr> </table>	<b>Surfbrett</b>	- laenge: double	+ Surfbrett() + Surfbrett(double) + setLaenge(double): void + getLaenge(): double	4,5P
<b>Surfbrett</b>					
- laenge: double					
+ Surfbrett() + Surfbrett(double) + setLaenge(double): void + getLaenge(): double					
I5.3	Jemand sagt: „Diese Katze ist 3 Jahre alt und schwarz.“ Zeichnen Sie das UML-Objektdiagramm der Katze!	AFII			
L5.3	<table border="1" style="display: inline-table; vertical-align: top;"> <tr> <td style="text-align: center;"><b>dieKatze: Katze</b></td> </tr> <tr> <td>alter = 3 farbe = schwarz</td> </tr> </table>	<b>dieKatze: Katze</b>	alter = 3 farbe = schwarz	2,5P	
<b>dieKatze: Katze</b>					
alter = 3 farbe = schwarz					

Abbildung 49: Lernziele und Indikatoren für die Themen 1 und 3 (OOP-Einführung / Klasse und Objekt)


T8	Vererbung	VT8
F1	<b>Die Schüler entwickeln Klassenmodellierungen mit Vererbung. Sie implementieren Ober- und Unterklassen und erzeugen in Anwendungsprogrammen entsprechende Objekte. Sie überprüfen die Syntax durch Kompilation und testen die Vererbungshierarchie zur Laufzeit.</b>	
I1/L1	Wird nicht abgefragt, da der Vergleichstest auf Papier geschrieben wird	
F2	<b>Die Schüler erläutern das Prinzip der Vererbung.</b>	
I2	Eine Oberklasse hat 2 Attribute. Wie viele Attributswerte hat ein Objekt der Unterklasse?	AFI
L2	O maximal 2                      O genau 2 X mindestens 2                    O kann man so nicht sagen	1P
F3	<b>Die Schüler modellieren Klassenhierarchien, erläutern abstrakte Methoden und abstrakte Klassen, überschreiben Methoden und setzen alles in Java-Quelltext um.</b>	
I3.1	Welche Aussage(n) ist/sind richtig? Eine abstrakte Klasse ...	AFI
L3.1	O enthält immer eine abstrakte Methode    X kann nicht instanziiert werden    O hat keine Attribute	1P
I3.2	Gegeben ist ein Quelltextausschnitt der folgenden Oberklasse: <pre>public abstract class Oberklasse {     private int attribut1;     // Im folgenden sind noch der parametrisierte Konstruktor und     // die Set- und Get-Methode für das Attribut deklariert!!!     public abstract void setzeAttribut1AufNull(); }</pre> Schreiben Sie einen Teil des Quelltextes der Klasse „Unterklasse“: Sehen Sie ein gekapseltes attribut2 (Datentyp: String), einen vollparametrisierten Konstruktor und die Implementierung der abstrakten Methode der Oberklasse vor!	AFII
L3.2	<pre>public class Unterklasse extends Oberklasse {     private String attribut2;     public Unterklasse(int attribut1, String attribut2) {         super(attribut1); this.attribut2 = attribut2;     }     public void setzeAttribut1AufNull() { setAttribut1(0); } }</pre>	7P
F4	<b>Die Schüler zeichnen UML-Klassendiagramme mit Vererbungshierarchien*.</b> * Hinweis: Wird verstärkt bei dem Zusatzthema „Objektorientierte Modellierung“ abgefragt!	
I4	Welcher „Vererbungs Pfeil“ ist im UML-Klassendiagramm anstelle der gestrichelten Linie vorzusehen? 	AFI
L4		1P

Abbildung 50: Lernziele und Indikatoren für das Thema 8 (Vererbung)

T9	Assoziation	VT9
F1	<b>Die Schüler entwickeln Klassenmodellierungen mit Assoziationen. Sie implementieren Beziehungen zwischen Klassen und erzeugen in Anwendungsprogrammen entsprechende Objekte inkl. der Beziehungen zueinander. Sie überprüfen die Syntax durch Kompilation und testen die Assoziationen zur Laufzeit.</b>	
I1/L1	Wird nicht abgefragt, da der Vergleichstest auf Papier geschrieben wird	
F2	<b>Die Schüler erläutern das Prinzip einer Assoziation und wenden Klassenattribute und Klassenmethoden an.</b>	
I2.1	Klassenattribute ...	AFI
L2.1	<input checked="" type="checkbox"/> sind immer Teil einer Klasse <input type="checkbox"/> sind den einzelnen Objekten zugeordnet <input checked="" type="checkbox"/> besitzen einen Wert	1P
I2.2	Welche Aussage(n) über Klassenmethoden (statische Methoden) trifft/treffen zu?	AFI
L2.2	<input type="checkbox"/> Sie besitzen immer ein Klassenattribut <input type="checkbox"/> Sie haben nie einen Rückgabewert <input checked="" type="checkbox"/> Aufruf erfolgt in der Regel mit: <code>Klassenname.methodenname()</code> <input type="checkbox"/> Keine Aussage trifft zu	1P
F3	<b>Die Schüler modellieren Assoziationen (auch mit Hilfe von UML-Klassendiagrammen) und setzen diese in Java-Quelltext um.</b>	
I3.1 / I3.2	Gegeben ist das folgende Szenario: Eine Firma besteht aus mehreren Abteilungen. Jede Abteilung hat genau einen Mitarbeiter als Abteilungsleiter und viele andere Mitarbeiter.	
I3.1	Zeichnen Sie das UML-Klassendiagramm!	AFII
L3.1	<pre> classDiagram     class Firma     class Abteilung     class Mitarbeiter     Firma &lt; -- Abteilung     Abteilung --&gt; Firma : bes: khl: sur &gt;&gt;     Abteilung --&gt; Mitarbeiter : has: Abteilungsleiter &gt;&gt;     Abteilung --&gt; Mitarbeiter : has &gt;&gt; </pre>	4P
I3.2	Ergänzen Sie den folgenden Quelltext (4 Zeilen), so dass die Assoziationen zwischen einer Abteilung und den Mitarbeitern implementiert sind! Hinweis: Durch den Konstruktor erhalten die privaten Attribute die entsprechenden Referenzen. Die Klasse Mitarbeiter liegt vor.	AFII
L3.2	<pre> public class Abteilung {     /*1*/     /*2*/     public Abteilung(Mitarbeiter abteilungsleiter,                      Mitarbeiter[] mitarbeiterListe) {          /*3*/         /*4*/     } // Konstruktor } // Abteilung </pre>	4P

Abbildung 51: Lernziele und Indikatoren für das Thema 9 (Assoziation)

## 4.2 Beschreibung themenübergreifender Fachkompetenzen

Auf den ersten öffentlichen Veranstaltungen, bei denen ich u. a. das Exposé (vgl. Anhang A) vorstellte, wurde von Teilnehmern nachgefragt, ob der OOP-First-Einstieg nicht auch andere Kompetenzen fördert als die Fachkompetenzen zu den einzelnen Themen: gerade eine „Modellierungs-Kompetenz“ würde doch mehr ausgebildet als beim OOP-Later-Einstieg. Zusätzlich wurde die ausschließlich statische Betrachtungsweise der OOP kritisiert und angeregt, nachzuprüfen, inwiefern die Schülerinnen und Schüler auch die dynamische Sichtweise der OOP verinnerlicht haben, also die Interaktionen zwischen den Objekten durch Botschaften (Methodenaufrufe).

Dies führte zu zwei weiteren Fachkompetenzformulierungen inkl. den dazu passenden Indikatoren um herauszufinden, inwiefern die beiden fachdidaktischen Vorgehensweisen Kompetenzen im Hinblick auf „dynamische OOP“ und „Objektorientierte Modellierung“ fördern [ES09a]:

„In the test at the end of the course, two topics were included: dynamic OOP and object-oriented modeling [...]. One might argue that introduction to programming only, and even a focus on language constructs is not sufficient. Therefore, dynamic OOP asks for understanding the dynamics of an object-oriented program. Modeling asks for the process of analyzing a problem and designing an object-oriented solution for a given problem. In the operationalization of the two different approaches for the study it might be that this kind of knowledge is missing.“ [ES09a, S. 18].

Für das erste Zusatzthema Z1 wird somit die dynamische OOP betrachtet. Zuerst werden Kompetenzen formuliert, z. B. die Kompetenz bzw. das Feinlernziel F2 (vgl. Abbildung 53):

„Die Schüler analysieren eine vorgegebene Klassenmodellierung und den Zustand der erzeugten Objekte und ihre Beziehungen zueinander. Sie erfassen den Initialisierungs-Zustand eines Objektes und die Zustandsänderung durch Methodenaufrufe auf den Objekten. Mit Hilfe eines Schreibtischtests verfolgen sie die dynamische Entwicklung der Objektzustände und die sich ändernde Beziehungsstruktur zwischen den Objekten nach.“

Als Indikator werden den Schülerinnen und Schülern zwei Java-Klassen (Klasse Mensch und Klasse Fahrrad) vorgegeben (siehe Abbildung 52). Danach sollen sie ein Anwendungsprogramm (Klasse mit der main-Methode) analysieren, indem Mensch- und Fahrrad-Objekte erzeugt werden. Den konkreten Menschen werden bestimmte Namen als Attributswerte und den konkreten Fahrrädern bestimmte Farben zugeordnet. Menschen können heiraten und Fahrräder kaufen. Nach der Objekterzeugung und der Objektinteraktion sollen die Schülerinnen und Schüler herausfinden, wie viele Objekte erzeugt wurden, wie viele Methodenaufrufe erfolgten und in welchem Zustand sich die Objekte am Programmende befinden (siehe Abbildung 53).

Zur Kompetenz bzw. zum Feinlernziel F1 (siehe Abbildung 53) gibt es keine Aufgabe, weil eine Implementierungskompetenz schwer durch einen schriftlichen Test (auf Papier) abgefragt werden kann.

Als zweites Zusatzthema wird die objektorientierte Modellierung (OOM) betrachtet. Zuerst werden Kompetenzen formuliert, z. B. die Kompetenz bzw. das Feinlernziel F2 (vgl. Abbildung 54):

„Die Schüler modellieren (z. B. nach einer vorgegebenen Problembeschreibung) Klassen (inkl. Vererbung) und ihre Beziehungen zueinander. Sie

zeichnen hierfür UML-Klassendiagramme und UML-Objektdiagramme.“

Als Indikator wird den Schülerinnen und Schülern ein Szenario vorgegeben (siehe Abbildung 54):

„Menschen können Wertgegenstände besitzen, z. B. besitzt Herr Müller ein konkretes Auto mit einer maximalen Geschwindigkeit und einem gewissen Wert. Frau Meyer besitzt ein konkretes Haus mit einer konkreten Quadratmeterzahl an Wohnfläche und einem gewissen Wert.“

Die Schülerinnen und Schüler sollen dann dieses Szenario mit Hilfe eines UML-Klassendiagramms modellieren.

Zur Kompetenz bzw. zum Feinlernziel F1 (siehe Abbildung 54) gibt es keine Aufgabe, weil das Testen einer Modellierung schwer durch einen schriftlichen Test (auf Papier) abgefragt werden kann.

```
public class Mensch {  
  
    private String name;  
    private Mensch Ehepartner;  
    private int anzahlFahrraeder;  
    private Fahrrad[] fahrraeder;  
  
    public Mensch() {  
        this("Janine"); }  
    public Mensch(String name) {  
        this.name = name;  
        this.anzahlFahrraeder = 0;  
        fahrraeder = new Fahrrad[100]; }  
    public void setName(String name) {  
        this.name = name; }  
    public String getName() {  
        return this.name; }  
    public int getAnzahlFahrraeder() {  
        return this.anzahlFahrraeder; }  
    public void heirate(Mensch einMensch) {  
        Ehepartner = einMensch; }  
    public void kaufe(Fahrrad einFahrrad) {  
        this.fahrraeder[anzahlFahrraeder] = einFahrrad;  
        this.anzahlFahrraeder++; }  
    public Mensch getEhepartner() {  
        return this.Ehepartner; }  
    public Fahrrad getFahrrad(int i) {  
        return fahrraeder[i]; }  
  
} // Klasse Mensch  
  
public class Fahrrad {  
  
    private String farbe;  
  
    public Fahrrad() {  
        this.farbe = ("unbekannt"); }  
    public Fahrrad(String farbe) {  
        this.farbe = farbe; }  
    public void setFarbe(String farbe) {  
        this.farbe = farbe; }  
    public String getFarbe() {  
        return this.farbe; }  
  
} // Klasse Fahrrad
```

Abbildung 52: Vorgegebene Java-Klassen für das Zusatzthema Z1 (dynamische OOP)

(aus Platzgründen ist der Quelltext in dieser Abbildung sehr komprimiert dargestellt)

Z1	Dynamische OOP	VT10
F1	Die Schüler implementieren Klassen. Sie erzeugen in einem Anwendungsprogramm Objekte mit Hilfe von parameterlosen und parametrisierten Konstruktoren und definieren ihre Beziehung zueinander. Durch algorithmische Strukturen sorgen sie für eine Zustandsänderung der einzelnen Objekte und für Änderungen in der Beziehungsstruktur. Sie testen ihre Software durch Kompilation und durch das Ausführen des Anwendungsprogramms. Sie überprüfen die Korrektheit ihres Programms und ihrer Klassenmodellierung durch entsprechende Bildschirmausgaben zur Laufzeit.	
II/L1	Wird nicht abgefragt, da der Vergleichstest auf Papier geschrieben wird	
F2	Die Schüler analysieren eine vorgegebene Klassenmodellierung und den Zustand der erzeugten Objekte und ihre Beziehungen zueinander. Sie erfassen den Initialisierungs-Zustand eines Objektes und die Zustandsänderung durch Methodenaufrufe auf den Objekten. Mit Hilfe eines Schreibtischtests verfolgen sie die dynamische Entwicklung der Objektzustände und die sich ändernde Beziehungsstruktur zwischen den Objekten nach.	
I2	Gegeben ist der folgende Quelltext (siehe auch Abbildung 52): <pre> public class DynamischeOOP {     public static void main(String[] args) {         Mensch ersterMensch = new Mensch("Klaus"), zweiterMensch = new Mensch();         Mensch dritterMensch = new Mensch(), vierterMensch = new Mensch();         zweiterMensch.setName("Harald"); dritterMensch.setName("Susi");         Fahrrad erstesFahrrad = new Fahrrad("rot"), zweitesFahrrad = new Fahrrad();         Fahrrad drittesFahrrad = new Fahrrad("blau");         ersterMensch.kaufe(zweitesFahrrad);         dritterMensch.kaufe(erstesFahrrad); dritterMensch.kaufe(drittesFahrrad);         zweiterMensch.heirate(dritterMensch); dritterMensch.heirate(zweiterMensch);         zweitesFahrrad.setFarbe("gruen"); drittesFahrrad.setFarbe("gelb");         System.out.println(ersterMensch.getName());         System.out.println(zweiterMensch.getEhepartner().getName());         int anzahl = dritterMensch.getAnzahlFahrraeder();         for (int i = 0; i &lt; anzahl; i++)             System.out.println(i+1 + ". Fahrrad: " +                 dritterMensch.getFahrrad(i).getFarbe());         System.out.println(vierterMensch.getName());     } // main } // Klasse DynamischeOOP </pre>	
I2.1	Wie viele Objekte werden zur Laufzeit durch die main-Methode erzeugt?	AFI
L2.1	7	2P
I2.2	Wie viele Methodenaufrufe auf Objekten erfolgen zur Laufzeit in der main-Methode?	AFII
L2.2	18	3P
I2.3	Wie lautet genau die Bildschirmausgabe?	AFII
L2.3	Klaus Susi 1. Fahrrad: rot 2. Fahrrad: gelb Janine	5P

Abbildung 53: Lernziele und Indikatoren für das Zusatzthema Z1 (dynamische OOP)



Z2	Objektorientierte Modellierung (OOM)	VT11
F1	Die Schüler entwickeln Klassenmodellierungen mit Vererbung und Assoziationen. Sie implementieren die Klassenstrukturen. Sie testen ihre Software durch Kompilation und durch das Ausführen des Anwendungsprogramms. Sie überprüfen die Korrektheit ihrer Modellierung durch entsprechende Testszenarien in Anwendungsprogrammen.	
I1/L1	Wird nicht abgefragt, da der Vergleichstest auf Papier geschrieben wird	
F2	Die Schüler modellieren (z. B. nach einer vorgegebenen Problembeschreibung) Klassen (inkl. Vererbung) und ihre Beziehungen zueinander. Sie zeichnen hierfür UML-Klassendiagramme und UML-Objektdiagramme.	
I2	Gegeben ist das folgende Szenario: Menschen können Wertgegenstände besitzen, z. B.: Herr Müller besitzt ein konkretes Auto mit einer maximalen Geschwindigkeit und einem gewissen Wert. Frau Meyer besitzt ein konkretes Haus mit einer konkreten Quadratmeterzahl an Wohnfläche und einem gewissen Wert. Modellieren Sie mit Hilfe eines UML-Klassendiagramms dieses Szenario!	AFII
L2	<pre> classDiagram     class Mensch {         arnede         name     }     class Wertgegenstand {         wert     }     class Auto {         maxGeschwindigkeit     }     class Haus {         wohnfläche     }     Mensch --&gt; Wertgegenstand : besitzt &gt;     Wertgegenstand &lt; -- Auto     Wertgegenstand &lt; -- Haus </pre>	10P

Abbildung 54: Lernziele und Indikatoren für das Zusatzthema Z2 (OOM)

Nachdem in den letzten beiden Kapiteln die Definition der Fachkompetenzen bzw. Feinlernziele, die Angabe der Indikatoren, die Zuordnung der Aufgaben zu den Anforderungsbereichen, die Nennung der Lösung und die Zuteilung der Punkte zu den Aufgaben für die neun Themen und die zwei Zusatzthemen für den Vergleichstest der Hauptstudie vorgenommen wurden, sollen im nächsten Kapitel der Vergleichstest analysiert und der Nachhaltigkeitstest und der Vortest entwickelt werden (vgl. Design der Hauptstudie in Abbildung 24).

### 4.3 Tests: Vergleichstest, Vortest und Nachhaltigkeitstest

In der Vorstudie wurde nur ein Vergleichstest (Post-Test) durchgeführt, also kein Vortest (Pre-Test) und auch kein Nachhaltigkeitstest (Follow-Up-Test). Im Folgenden sollen die Unterschiede des Vergleichstests in der Vorstudie und in der Hauptstudie genannt werden (siehe Kapitel 4.3.1), die Anforderungsbereiche und Punkte der einzelnen Fragen in Abhängigkeit von den Themen betrachtet und der zeitliche Rahmen bestimmt werden (siehe Kapitel 4.3.2). Danach wird aus dem Vergleichstest der Vortest (siehe Kapitel 4.3.3) und der Nachhaltigkeitstest der Hauptstudie (siehe Kapitel 4.3.4) generiert. Alle drei Tests werden am Ende (siehe Kapitel 4.3.5) einer kritischen Reflexion unterzogen.

### 4.3.1 Vergleichstest (Post-Test) in der Vorstudie und in der Hauptstudie in Bezug zu den abgefragten Themen

Der Vergleichstest in der Vor- und Hauptstudie unterscheidet sich von der Einteilung der abgefragten Themen sehr (siehe auch Abbildung 56 und Kapitel 3.3.14). Der Vorstudientest war noch geprägt von einem Vorgehen am OSZ IMT, bei dem die Einführungsphasen-Klassen (schon vor Beginn dieser Studie) nach einem Schuljahr mit einem Test verglichen wurden. Die vier Aufgaben-Blätter aus dem Jahre 2004, von vier verschiedenen beteiligten Lehrern erstellt, hatten die vier Themen Datentypen, Steuerstrukturen, Struktogramme und UML (siehe Anhang C und Abbildung 55).

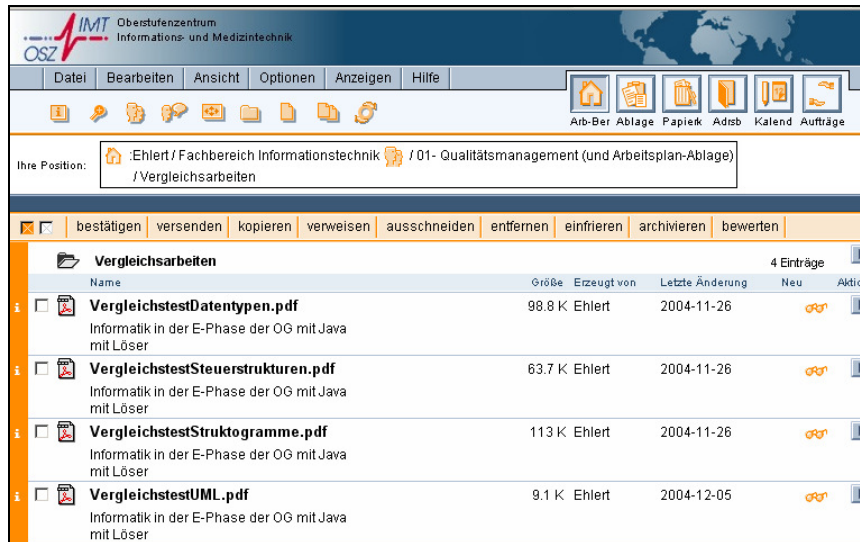


Abbildung 55: Vergleichstests vor der Studie (aus dem Jahr 2004)

Der Vergleichstest der Vorstudie orientierte sich noch an diesem Vorgehen, ordnete aber die Fragen der vier Themen teilweise in neue Kategorien ein oder fasste Themen zusammen. Zusätzlich wurden Fragen zum neuen Thema „dynamische OOP“ entwickelt. Letztlich gab es somit die folgenden fünf Themen und ein Zusatzthema (siehe Anhang F und Abbildung 56): „Steuer- und Datenstrukturen“, „Struktogramme“, „Methoden“, „statische OOP“, „UML“ und „dynamische OOP“.

Vergleichstest vor der Studie		Vergleichstest in der Vorstudie		Vergleichstest in der Hauptstudie	
1	Datentypen	1	Steuer- und Datenstrukturen	1	Datentypen und Steuerstruktur Sequenz
2	Steuerstrukturen	2	Struktogramme	2	Steuerstruktur: Selektion
3	Struktogramme	3	Methoden	3	Steuerstruktur: Iteration
4	UML	4	Statische OOP	4	Arrays und Strings
		5	UML	5	Methoden
		6	Dynamische OOP	6	Einführung in die OOP /
				7	Klasse und Objekt
				8	Vererbung
				9	Assoziation
				10	Dynamische OOP
				11	OOM

Abbildung 56: Dynamische Entwicklung des Vergleichstests im Laufe der Studie

Der Vergleichstest der Hauptstudie (siehe Anhang M) wird dagegen völlig neu gestaltet. Im Mittelpunkt steht die Idee, dass alle neun unterrichteten Themen sich in den Aufgaben wiederfinden sollen. Das Zusatzthema „dynamische OPP“ wird ergänzt durch das weitere Zusatzthema „OO-Modellierung“. Der Vergleichstest der Hauptstudie wird generiert durch die Aufsummierung der Indikatoren bzw. der Aufgaben, die zu den elf Themen entwickelt wurden (siehe Kapitel 4.1 und Kapitel 4.2).

Trotzdem können einige Aufgaben (teilweise in abgeänderter Form) des Vergleichstests der Vorstudie wieder verwendet werden (siehe Abbildung 57, Abbildung 58 und Abbildung 59), da auch sie aus den Lernzielen der einzelnen Themen (mit Hilfe von Indikatoren) entwickelt wurden und sie durch eine Itemschwierigkeits-Analyse als geeignet eingestuft wurden. Diese Lernziele finden sich jetzt aber oft unter anderen Themenüberschriften wieder (siehe Kapitel 4.1 und Kapitel 4.2).

Vorstudie Aufgabe	Punkte	P <sub>i</sub> OOP-F.	P <sub>i</sub> OOP-L.	Thema des Vergleichstests der Vorstudie	Hauptstudie Aufgabe	Punkte
1.1.				Thema: Kontroll- und Datenstrukturen (Multiple-Choice)		
1	1	1,00	0,88	leicht verändert übernommen	1.1	1
2	1	0,5	0,58	leicht verändert übernommen	1.2	1
3	1	0,96	0,65	leicht verändert übernommen	1.3	1
4	1	0,96	0,62	übernommen	1.4	1
5	1	1,00	0,96	übernommen	2.1	1
6	1	0,79	0,54	verändert übernommen	4.1	1
7	1	0,88	0,46	leicht verändert übernommen	4.2	1
8	1	0,67	0,92	gestrichen: passt nicht mehr ins Aufgaben-Schema		
9	1	0,58	0,27	leicht verändert übernommen	3.2	1
10	1	0,38	0,85	gestrichen: passt nicht mehr ins Aufgaben-Schema		
1.2				Thema: Kontroll- und Datenstrukturen		
1	4	0,72	0,75	leicht verändert übernommen	1.5	3
2	6	0,25	0,35	verändert übernommen	2.3.a	4
2.1				Thema: Struktogramme (Multiple-Choice)		
1	1	0,29	0,92	gestrichen, findet sich aber im neuen Aufgabentyp wieder	2.3.b	
2	1	1,00	0,92	gestrichen, findet sich aber im neuen Aufgabentyp wieder	3.4	
3	1	0,96	0,92	gestrichen, wird aber durch andere Aufgabe aufgenommen	1.6	
4	1	0,79	0,85	leicht verändert übernommen	3.1	1
5	1	0,79	0,69	gestrichen, wird aber durch andere Aufgabe aufgenommen	1.6	3
6-7	2	0,63	0,54	leicht verändert übernommen	3.3	2
8-10	3	0,63	0,46	verändert, findet sich aber im neuen Aufgabentyp wieder	3.4	4
2.2				Thema: Struktogramme		
1	10	0,64	0,60	verändert übernommen	2.3.b	4
3.1				Thema: Methoden (Multiple-Choice)		
1	1	0,38	0,65	leicht verändert übernommen	5.1	1
2	1	0,88	0,92	gestrichen		
3	1	0,46	0,54	leicht verändert übernommen	5.2	1
4	1	0,50	0,65	gestrichen		
5	1	0,54	0,77	verändert	5.3	1
6	1	0,46	0,42	gestrichen		
7	1	0,79	0,73	gestrichen		
8	1	0,79	0,96	gestrichen		
9	1	0,96	0,85	gestrichen		
10	1	0,67	0,69	gestrichen		
3.2				Thema: Methoden		
1	3	0,53	0,58	übernommen	5.4	3
2	4	0,45	0,53	gestrichen		
3	3	0,70	0,53	verändert übernommen	5.5	4

Abbildung 57: Vergleichstest-Fragen in der Vor- und Hauptstudie (Teil 1)

Vorstudie Aufgabe	Punkte	Pi OOP-F.	Pi OOP-L.	Thema des Vergleichstests der Vorstudie	Hauptstudie Aufgabe	Punkte
4.1				Thema: statische OOP (Multiple-Choice)		
1	1	0,92	0,85	leicht verändert übernommen	67.1	1
2	1	0,88	0,81	leicht verändert übernommen	67.5	1
3	1	0,67	0,81	leicht verändert übernommen	67.6	1
4	1	0,25	0,58	leicht verändert übernommen	67.2	1
5	1	0,79	0,7	gestrichen		
6	1	0,88	0,85	gestrichen		
7	1	0,46	0,58	leicht verändert übernommen	67.3	1
8	1	0,67	0,38	leicht verändert übernommen	8.2	1
9	1	0,79	0,77	übernommen	8.1	1
10	1	0,42	0,88	gestrichen		
4.2				Thema: statische OOP		
1	3	0,74	0,82	übernommen	67.8	3
2	3	0,84	0,83	übernommen	67.9	2
3	2	0,80	0,87	übernommen	67.7	2
4	2	0,66	0,86	gestrichen		
5.1				Thema: UML (Multiple-Choice)		
1	1	0,67	1,00	gestrichen, findet sich aber im neuen Aufgabentyp	67.11	
2	1	0,79	1,00	gestrichen, findet sich aber im neuen Aufgabentyp	67.10	
3	1	0,63	0,88	gestrichen, findet sich aber im neuen Aufgabentyp	67.10	
4	1	0,92	0,85	gestrichen, findet sich aber im neuen Aufgabentyp	67.10	
5	1	0,88	0,96	gestrichen, findet sich aber im neuen Aufgabentyp	67.10	
6	1	0,50	0,69	gestrichen, findet sich aber im neuen Aufgabentyp	67.11	
7	1	0,88	0,92	gestrichen, findet sich aber im neuen Aufgabentyp	67.10	
8	1	0,88	0,85	gestrichen, findet sich aber im neuen Aufgabentyp	67.10	
9	1	0,71	0,73	gestrichen, findet sich aber im neuen Aufgabentyp	67.11	
10	1	0,88	0,96	übernommen	8.3	1
5.2				Thema: UML		
1	7	0,73	0,85	leicht verändert übernommen	67.10	4,5
2	3	0,51	0,74	übernommen	67.11	2,5
6				Thema: dynamische OOP		
1	2	0,88	1,00	übernommen	10.1	2
2	3	0,42	0,47	übernommen	10.2	3
3	5	0,40	0,66	übernommen	10.3	5
	<b>Summe</b>	<b>Ø</b>	<b>Ø</b>			<b>Summe</b>
	<b>110</b>	<b>0,69</b>	<b>0,74</b>			<b>71</b>

Abbildung 58: Vergleichstest-Fragen in der Vor- und Hauptstudie (Teil 2)

Es lassen sich die folgenden Aussagen treffen (siehe Abbildung 56, Abbildung 57, Abbildung 58 und Abbildung 59):

- Der Vergleichstest der Hauptstudie enthält elf Themen statt der sechs Themen des Vergleichstests der Vorstudie. Nur drei Themen („Methoden“, „statische OOP“ und „dynamische OOP“) sind gleich und finden sich in beiden Tests.
- Statt 62 Fragen gibt es jetzt 49 Fragen. Von den 62 Fragen wurden 27 gestrichen, sechs verändert übernommen, 17 leicht verändert übernommen und zwölf übernommen.
- Da auch im Vergleichstest der Vorstudie das Lösen einer Aufgabe als Indikator für das Erreichen eines Feinlernziels angesehen wurde, können Items, die sich auf das gleiche Feinlernziel beziehen, wiederverwendet werden. Durch die neue Gestaltung (bzw. das Redesign) des Vergleichstests wurde aber auf eine ausführliche Itemanalyse des Vorstudien-Vergleichstests verzichtet, da das entscheidende Kriterium für ein Item war, ob es als Indikator für ein Feinlernziel eines neuen Themas geeignet ist. Trotzdem dürfen Items aus dem

Vorstudien-Vergleichstest wiederverwendet werden, wenn sie sich auf das gleiche Feinlernziel beziehen und eine passende Itemschwierigkeit besitzen.

- Die Itemschwierigkeit sollte in der Regel zwischen 0,2 und 0,8 liegen (vertiefende Informationen zur Itemanalyse finden sich im Kapitel 4.3.5). Von den 17 leichten Items (über 0,8 für beide Gruppen) wurden neun gestrichen, u. a. deshalb, um nicht zu stark ausdifferenzieren. Drei Items wurden leicht verändert übernommen und fünf vollständig übernommen, z. B. um den Einstieg in ein neues Thema mit einem leichten Item zu beginnen (z. B. Aufgaben 1.1, 2.1, 67.1 und 10.1). Bei den neuen Aufgaben für den Vergleichstest der Hauptstudie müssen auch Itemschwierigkeiten unter dem Wert 0,2 liegen, damit auch leistungsstarke Schülerinnen und Schüler ausdifferenziert werden können. Bei der späteren Auswertung des Vergleichstests im Hinblick auf die Itemschwierigkeit (siehe Abbildung 64 und Abbildung 65) wird erkennbar, dass dieses Vorhaben auch gelingt: Drei Items liegen unter dem Wert 0,2.

Hauptstudie Aufgabe	Thema Nr.	Thema des Vergleichstests der Hauptstudie	Max. Punkte	Punkte aus gleichen Aufgaben der Vorstudie	Punkte aus ähnlichen Aufgaben der Vorstudie
1	T2	Datentypen und Sequenz	10	1	9
2	T5	Steuerstruktur: Selektion	10	1	8
3	T6	Steuerstruktur: Iteration	10	0	8
4	T7	Arrays und Strings	10	0	2
5	T4	Methoden	10	3	7
6 / 7	T1 / T3	Einführung in die OOP / Klasse und Objekt	20	9,5	9,5
8	T8	Vererbung	10	2	1
9	T9	Assoziation	10	0	0
10	Z1	Dynamische OOP	10	10	0
11	Z2	OOM	10	0	0
<b>Summe</b>			<b>110</b>	<b>26,5</b>	<b>44,5</b>

Abbildung 59: Punkte aus Aufgaben des Vergleichstests der Vorstudie in Aufgaben der Hauptstudie

- 26,5 Punkte von 110 Punkten des Vergleichstests der Hauptstudie stammen aus Aufgaben des Vergleichstests der Vorstudie, 44,5 Punkte von 110 Punkten stammen aus Aufgaben, die ähnlich sind wie Aufgaben des Vergleichstests der Vorstudie. Dabei wurden die Indikatoren wie folgt neu geordnet:
  - Aufgaben zum alten Thema „Kontroll- und Datenstrukturen“ finden sich teilweise in den neuen Themen „Datentypen und Steuerstruktur Sequenz“, „Steuerstruktur Iteration“ und „Steuerstruktur Selektion“ wieder.
  - Aufgaben zum alten Thema „Struktogramme“ finden sich teilweise in den neuen Themen „Datentypen und Steuerstruktur Sequenz“, „Steuerstruktur Iteration“ und „Steuerstruktur Selektion“ wieder.
  - Das alte Thema „Methoden“ ist gleichzeitig auch das neue Thema „Methoden“. Die Aufgaben wurden (zur Hälfte) komplett übernommen. Die Punkte für dieses Thema sind von 20 auf 10 reduziert worden.
  - Die alten Themen „statische OOP“ und „UML“ finden sich teilweise in den Themen „statische OOP“ und „Vererbung“ (als neues Thema) wieder.

- Das Thema „dynamische OOP“ konnte komplett übernommen werden, sowohl von den Aufgaben als auch von den Punkten.
- Die neuen Themen „Arrays und Strings“ und „Vererbung“ des Vergleichstests der Hauptstudie enthalten relativ wenig vergleichbare Aufgaben aus der Vorstudie.
- Die Aufgaben für die neuen Themen „Assoziation“ und „OOM“ mussten komplett neu generiert werden.
- Der Begriff „leicht verändert“ bei den Aufgaben bedeutet in der Regel, dass eine Antwortmöglichkeit mehr oder weniger vorhanden ist oder die Frage- bzw. Antwortsätze leicht verändert wurden, z. B. von „Welche Antworten sind falsch?“ auf „Welche Antworten sind richtig?“.

### 4.3.2 Vergleichstest (Post-Test) in der Hauptstudie

In den beiden Kapiteln 4.1 und 4.2 wurden in den Abbildungen, welche die Informationen zu den einzelnen Themen in einer verkürzten Form darstellen, schon die Punkte und die Anforderungsbereiche mit aufgeführt, zwei für einen Test relevante Informationen. Jetzt fehlen nur noch die Analyse und der Vergleich der Schwere der Aufgaben zu den einzelnen Themen und die Bestimmung der zeitlichen Länge des Tests.

Im Folgenden sollen zu den einzelnen Themen des Vergleichstests (Post-Test) der Hauptstudie die Anforderungsbereiche analysiert werden (siehe Abbildung 60), indem die ausgewiesenen Anforderungsbereiche der Indikatoren bzw. Aufgaben aus Kapitel 4.1 und Kapitel 4.2 aufsummiert werden. Zusätzlich werden in Abbildung 60 die Anzahl der Aufgaben und die maximal erreichbaren Punkte (davon in Klammern die maximal erzielbaren Punkte durch Multiple-Choice-Aufgaben) pro Thema angegeben.

Aufgabe	Thema Nr.	Thema	Anzahl Aufgaben	Max. Punkte (MC-Punkte)	Punkte AFI	Punkte AFII	Punkte AFIII
1	T2	Datentypen und Steuerstruktur Sequenz	6	10 (4)	3	7	0
2	T5	Steuerstruktur: Selektion	4	10 (2)	1	9	0
3	T6	Steuerstruktur: Iteration	6	10 (2)	2	8	0
4	T7	Arrays und Strings	5	10 (2)	1	9	0
5	T4	Methoden	5	10 (3)	3	7	0
6 / 7	T1 / T3	Einführung in die OOP / Klasse und Objekt	11	20 (6)	6	14	0
8	T8	Vererbung	4	10 (3)	3	7	0
9	T9	Assoziation	4	10 (2)	2	8	0
10	Z1	Dynamische OOP	3	10 (0)	2	8	0
11	Z2	OOM	1	10 (0)	0	10	0
<b>Summe</b>			<b>49</b>	<b>110 (24)</b>	<b>23</b>	<b>87</b>	<b>0</b>

Abbildung 60: Themen im Vergleichstest der Hauptstudie im Kontext zu den Anforderungsbereichen

Aus dieser Übersicht (siehe Abbildung 60) lassen sich die folgenden Aussagen ableiten:

- Die maximal erzielbaren Punkte durch Multiple-Choice-Antworten liegen pro Thema zwischen 0% (Themen Z1 und Z2) und 40% (Thema T2) der insgesamt erzielbaren Punkte. Im Mittel wird aber die Vorgabe aus Kapitel 3.3.12, dass der Anteil nicht über 35% liegen sollte, mit 22% (24/110) deutlich unterschritten.
- Durch den Multiple-Choice-Anteil kann auch manche Frage zufällig richtig gelöst werden. Dies spielt später bei der Bewertung der Ergebnisse (vgl. Kapitel 6.3.1) eine Rolle.
- Jede Aufgabe hat im Durchschnitt 2,2 (110:49) Bewertungspunkte.
- Der in der „Einheitlichen Prüfungsanforderung Informatik (vgl. [Kmk04] und Kapitel 4.1) gewünschte Schwerpunkt der Aufgaben auf den Anforderungsbereich II wird durch den Vergleichstest (mit AFI/AFII/AFIII = 21/79/0) realisiert, genauso wie die Vorgabe, dass der Anteil von Aufgaben des Anforderungsbereichs I höher liegen soll als der des Anforderungsbereichs III. Aufgaben zum Anforderungsbereich III fehlen aber gänzlich bei dem Test, weil es einen Widerspruch gegeben hätte, wenn jedem der elf Themen eine schwere Teilaufgabe (z. B. eine Transferleistung) zugeordnet und diese nur mit einem Wertungspunkt (1P) versehen worden wäre, da im Informatik-Abitur in der Regel maximal 10% der Aufgaben dem AFIII zugeordnet sind bzw. zugeordnet sein dürfen.
- Der Vortest (Pre-Test) und der Nachhaltigkeitstest (Follow-Up-Test) sollen von der Verteilung der Anforderungsbereiche ähnlich sein, damit die Testergebnisse vergleichbar sind.

Interessant ist eine Aufteilung der Fragen zu den Themen nach der Schwierigkeit. Hierfür wird der prozentuale Anteil des AFII-Bereichs an den Gesamtpunkten betrachtet (siehe Abbildung 61: Die Farben veranschaulichen die unterschiedlichen Schwierigkeits-Bereiche).

Aufgabe	Thema Nr.	Thema	AFII in Prozent
1	T2	Datentypen und Steuerstruktur Sequenz	70
5	T4	Methoden	70
6 / 7	T1 / T3	Einführung in die OOP / Klasse und Objekt	70
8	T8	Vererbung	70
3	T6	Steuerstruktur: Iteration	80
9	T9	Assoziation	80
10	Z1	Dynamische OOP	80
2	T5	Steuerstruktur: Selektion	90
4	T7	Arrays und Strings	90
11	Z2	Objektorientierte Modellierung (OOM)	100
		<b>Durchschnitt</b>	79

Abbildung 61: Themen des Vergleichstests der Hauptstudie nach Schwierigkeit (AFII) geordnet

Ideal wäre es gewesen, wenn allen Themen denselben Wert (z. B. 80%) gehabt hätten. Da dies nicht der Fall ist, müssen die Werte dieser Tabelle später bei der Beurteilung der Ergebnisse des Vergleichstests (siehe Kapitel 6.3.2 und Kapitel 7.3.1) mit berücksichtigt werden.

Für jedes der elf Themen können die Schülerinnen und Schüler zehn Punkte erhalten. Somit ergibt sich als maximale erreichbare Punktzahl des Tests der Wert 110.

Jetzt muss noch die Bearbeitungszeit bestimmt werden. Dazu werden folgende Entscheidungen getroffen (siehe Abbildung 62):

- Die Schülerinnen und Schüler erhalten für die leichtesten Themen pro Bewertungspunkt eine Minute Zeit, dies ergibt als Summe zehn Minuten.
- Pro 10% mehr AFII-Schwierigkeitsgrad gibt es zusätzlich zwei Minuten Zeit.
- Die Zeit ist so großzügig bemessen, dass der Test als Niveautest kategorisiert werden kann (vgl. [LR98, S. 164] und Itemanalyse in Kapitel 4.3.5).

Somit ist die Zeit für die Schülerinnen und Schüler zur Bearbeitung des Vergleichstests insgesamt 130 Minuten.

Soweit die Theorie. In der Schulpraxis soll der Test dann in zwei Etappen geschrieben werden, zum einen, um die Schüler nicht zu sehr zu beanspruchen, zum anderen, weil im vorgegebenen Quelltext des Zusatzthemas Z1 („dynamische OOP“) Lösungsansätze für andere Aufgaben enthalten sind:

- Die Themen T1 bis T9 werden zuerst in 100 Minuten (statt 102 Minuten) geschrieben.
- Die Zusatzthemen Z1 und Z2 werden ein paar Tage später in 30 Minuten (statt 28 Minuten) geschrieben.

Aufgabe	Thema Nr.	AFII in Prozent	Zeit in Minuten
1	T2	70	10
5	T4	70	10
6 / 7	T1 / T3	70	20
8	T8	70	10
3	T6	80	12
9	T9	80	12
10	Z1	80	12
2	T5	90	14
4	T7	90	14
11	Z2	100	16
		<b>Summe</b>	130

Abbildung 62: Zeitlicher Rahmen des Vergleichstests der Hauptstudie

Insgesamt sind dies fast drei Schulstunden (3 \* 45 Minuten = 135 Minuten). Der Vergleichstest hat somit eine angemessene Länge, überfordert durch die Aufteilung nicht die Schüler und entspricht auch der Vorgabe aus Kapitel 3.3.12, die besagt, dass der zeitliche Rahmen des Vergleichstests mehrere Schulstunden betragen kann.

### 4.3.3 Vortest (Pre-Test) in der Hauptstudie

Ein Vortest (Pre-Test) wurde in der Vorstudie als nicht erforderlich gehalten, da die Schüler (fast) ohne Programmier-Vorkenntnisse an das Oberstufenzentrum Informations- und Medizintechnik kommen. Da dies eine



unbewiesene Behauptung ist, wird für die Hauptstudie ein Vortest vorgesehen, damit die Behauptung belegt oder widerlegt werden kann. Um die Schülerinnen und Schüler aber nicht zu stark mit ihrem „Nichtwissen“ zu konfrontieren und ihre Motivation für eine Studienteilnahme aufrechtzuerhalten, werden nicht die Schülerinnen und Schüler der Hauptuntersuchung, sondern die des darauf folgenden Jahrgangs untersucht (unter der Annahme, dass sich dieser nicht signifikant von dem Jahrgang davor unterscheidet) und es wird dabei der Vergleichstest deutlich reduziert als Vortest verwendet [ES09a]:

„The pre-test was a small sub-sample of these questions. However, as the question of OOP first versus OOP later is focused on introductory programming, we did not expect any prior knowledge. Instead, giving students a big test at the beginning, in which they are likely to fail completely, could lead to negative attitudes. Therefore we decided to not give the pre-test to the students involved in the study; instead we provided this test to another group of students not involved in the study. In addition, we thought it is reliable enough to assume equality of the students regarding prior knowledge.”  
[ES09a, S. 17].

Die Reduktion ist wie folgt:

- Statt der elf Themen werden nur die ersten acht Themen abgefragt, da die drei Themen „Assoziation“, „dynamische OOP“ und „OOM“ zeitlich am Ende des Schuljahres unterrichtet werden und damit als zu anspruchsvoll (selbst für fortgeschrittene Programmier-Anfänger) erachtet werden.
- Die acht Themen werden in zwei Hälften geteilt, jeder Schüler muss also nur Fragen zu vier Themen beantworten, damit sich die Zeit eventueller Frustration in Grenzen hält.

Somit erhält ein Teil jeder Klasse Fragen zu den folgenden Themen (siehe Anhang Q):

- VT1: Datentypen und Steuerstruktur Sequenz
- VT3: Steuerstruktur Iteration
- VT6/T7: Einführung in die OOP / Klasse und Objekt

Der andere Teil jeder Klasse erhält Fragen zu den folgenden Themen (siehe Anhang R):

- VT2: Steuerstruktur Selektion
- VT4: Arrays und Strings
- VT5: Methoden
- VT8: Vererbung

Beide Gruppen können somit maximal 40 Punkte erringen. Der Vortest ist also ein um die Hälfte verkürzter Vergleichstest im Hinblick auf die Themen eins bis acht. Das neunte Thema (Assoziation) und die beiden Zusatzthemen werden nicht abgefragt.

Die Ergebnisse (siehe Kapitel 6.3.1) bestätigen die Vermutung, dass die Schülerinnen und Schüler nahezu ohne Programmier-Vorkenntnisse an das OSZ IMT kommen: Sie erzielten im Durchschnitt nur 3,9 von 40 Punkten,

obwohl der Vortest (mit AFI/AFII/AFIII = 24/76/0) geringfügig leichter ist als der Vergleichstest (mit AFI/AFII/AFIII = 21/79/0), da drei Themen gar nicht abgefragt werden.

#### 4.3.4 Nachhaltigkeitstest (Follow-Up-Test) in der Hauptstudie

Der Nachhaltigkeitstest (Follow-Up-Test) in der Hauptstudie (siehe Anhang O) muss sich einerseits vom Vergleichstest unterscheiden (so dass sich die Schüler nicht an die Fragen bzw. Lösungen erinnern können), andererseits muss er aber dem Vergleichstest sehr ähnlich sein, sowohl vom Umfang als auch von der Art und der Schwere der Fragen.

Es gibt somit wieder Fragen zu den folgenden neun Themen und zwei Zusatzthemen:

- NT1: Thema T2 (Datentypen und Steuerstruktur Sequenz)
- NT2: Thema T5 (Selektion)
- NT3: Thema T6 (Iteration)
- NT4: Thema T7 (Arrays und Strings)
- NT5: Thema T4 (Methoden)
- NT6 / NT7: Themen T1 und T3 (Einführung in die OOP / Klasse und Objekt)
- NT8: Thema T8 (Vererbung)
- NT9: Thema T9 (Assoziation)
- NT10: Zusatzthema Z1 (dynamische OOP)
- NT11: Zusatzthema Z2 (OOM)

Es lassen sich die folgenden Aussagen treffen (siehe auch Abbildung 63, Abbildung 64 und Abbildung 65):

- Von den ursprünglich 49 Fragen des Vergleichstests werden 40 Fragen für den Nachhaltigkeitstest nur leicht verändert. Beispielhaft wird dies in Abbildung 63 für jeweils eine Aufgabe der neun Themen demonstriert. Ziel ist es, die Fragen bzw. Antworten vergleichbar zu halten. Nur acht Aufgaben werden mehr als leicht verändert, z. B. um die unerwünschte Itemschwierigkeit 1,00 (Frage 10.1: alle Schüler haben die Aufgabe gelöst) zu verringern. Nur eine Frage (Frage 10.2) entfällt komplett. Es sind somit nur noch 48 Fragen.
- Der Nachhaltigkeitstest ist von der Verteilung der Anforderungsbereiche (mit AFI/AFII/AFIII = 20/80/0) mit dem Vergleichstest (mit AFI/AFII/AFIII = 21/79/0) fast identisch.
- Das Zusatzthema „Dynamische OOP“ (10. Aufgabe) wird von einem „Mensch-Ehepartner-Fahrrad-Szenario“ (Vergleichstest) auf ein „Mensch-Pferd-Sattel-Szenario“ (Nachhaltigkeitstest) umgestellt.
- Das Zusatzthema „OO-Modellierung“ (11. Aufgabe) wird von einem „Mensch-Wertgegenstand-Auto-Haus-Szenario“ (Vergleichstest) auf ein „Mensch-Fahrzeug-PKW-LKW-Szenario“ (Nachhaltigkeitstest) umgestellt.

Thema	Frage	Test
T2	Welche Werte kann eine Variable vom Datentyp <code>byte</code> annehmen?	VT1
Neu	Welche Werte kann eine Variable vom Datentyp <code>short</code> annehmen?	NT1
T5	Gegeben ist der folgende Quelltextauschnitt: <pre>int a = 5;                // oder 7 oder 9 switch (a) {     case 5:    a = 3 * a;    break;     case 9:    a = a / 2;    break;     default:   a = a % 3;    break; } // switch if (a &gt;= 15) a--; else a++; System.out.println("a: " + a);</pre> Welcher Wert wird für die Variable <code>a</code> (3 Tabelleneinträge) ausgegeben?	VT2
Neu	Gegeben ist der folgende Quelltextauschnitt: <pre>int a = 4;                // oder 6 oder 8 switch (a) {     case 4:    a = 3 * a;    break;     case 6:    a = a / 5;    break;     default:   a = a % 3;    break; } // switch if (a &lt; 10) a--; else a++; System.out.println("a: " + a);</pre> Welcher Wert wird für die Variable <code>a</code> (3 Tabelleneinträge) ausgegeben?	NT2
T6	Welche Antwort(en) enthält/enthalten eine Endlosschleife?	VT3
Neu	Welche Antwort(en) enthält/enthalten <u>keine</u> Endlosschleife? (Mit leicht geänderten Antworten)	NT3
T7	Gegeben ist der folgende Quelltextauschnitt: <pre>int[] exampleArray = {10, 20, 30, 40}; System.out.println(exampleArray[2*2]*3);</pre> Wie lautet die Bildschirmausgabe?	VT4
Neu	Gegeben ist der folgende Quelltextauschnitt: <pre>int[] exampleArray = {10, 20, 30, 40}; for (int i = 1; i &lt;= exampleArray.length; i++)     System.out.println(exampleArray[i]);</pre> Wie lautet die Bildschirmausgabe?	NT4
T4	Beim Aufruf einer Methode findet man nie ...	VT5
Neu	Beim Aufruf einer Methode findet man <u>immer</u> ...	NT5
T1 / T3	Die Klasse „Reisebuero“ hat u. a. das Attribut „anzahlKunden“ vom Datentyp „int“. Wie lautet der vollständige Quelltext der „Setter“-Verwaltungsmethode?	VT6 / VT7
Neu	Die Klasse „Bruecke“ hat u. a. das Attribut „breite“ vom Datentyp „float“. Wie lautet der vollständige Quelltext der „Setter“-Verwaltungsmethode?	NT6 / NT7
T8	Eine Oberklasse hat 2 Attribute. Wie viele Attributswerte hat ein Objekt der Unterklasse?	VT8
Neu	Eine Unterklasse hat 3 Attribute. Wie viele Attributswerte hat ein Objekt der Oberklasse? (Mit geänderten Antworten)	NT8
T9	Welche Aussage(n) über Klassenmethoden (statische Methoden) trifft/treffen zu?	VT9
Neu	Welche Aussage(n) über Klassenmethoden (statische Methoden) trifft/treffen zu? (Mit geänderter Antwort)	NT9

Abbildung 63: Exemplarische Beispiele der Unterschiede zwischen Nachhaltigkeits- und Vergleichstest

VT Aufgabe	Punkte	P <sub>i</sub> OOP-Later	Thema des Vergleichstests (VT) bzw. des Nachhaltigkeitstests (NT) in der Hauptstudie	NT Aufgabe	Punkte	AF
1.			Thema: Datentypen und Steuerstruktur Sequenz (MC)			
1.1	1	0,89	Aufgabe leicht verändert	1.1	1	I
1.2	1	0,53	Aufgabe leicht verändert	1.2	1	I
1.3	1	0,53	Aufgabe leicht verändert	1.3	1	I
1.4	1	0,79	Aufgabe leicht verändert	1.4	1	II
			Ohne Multiple-Choice			
1.5	3	0,60	Aufgabe leicht verändert	1.5	3	II
1.6	3	0,82	Aufgabe leicht verändert	1.6	3	II
2.			Thema: Steuerstruktur: Selektion (Multiple-Choice)			
2.1	1	0,66	Aufgabe leicht verändert	2.1	1	I
2.2	1	0,68	Aufgabe leicht verändert	2.2	1	II
			Ohne Multiple-Choice			
2.3a	4	0,55	Aufgabe leicht verändert	2.3a	4	II
2.3b	4	0,89	Aufgabe leicht verändert	2.3b	4	II
3.			Thema: Steuerstruktur: Iteration (Multiple-Choice)			
3.1	1	0,58	Aufgabe leicht verändert	3.1	1	I
3.2	1	0,26	Aufgabe leicht verändert	3.2	1	I
			Ohne Multiple-Choice			
3.3	2	0,47	Aufgabe leicht verändert	3.3	2	II
3.4a	2	0,63	Aufgabe leicht verändert	3.4a	2	II
3.4b	2	0,91	Aufgabe leicht verändert	3.4b	2	II
3.5	2	0,55	Aufgabe leicht verändert	3.5	2	II
4.			Thema: Arrays und Strings (Multiple-Choice)			
4.1	1	0,13	Aufgabe leicht verändert	4.1	1	II
4.2	1	0,11	Aufgabe leicht verändert	4.2	1	I
			Ohne Multiple-Choice			
4.3	2	0,25	Aufgabe erweitert	4.3	3	II
4.4	3	0,39	Aufgabe leicht verändert	4.4	3	II
4.5	3	0,61	Aufgabe reduziert	4.5	2	II
5.			Thema: Methoden (Multiple-Choice)			
5.1	1	0,47	Aufgabe leicht verändert	5.1	1	I
5.2	1	0,63	Aufgabe leicht verändert	5.2	1	I
5.3	1	0,39	Aufgabe leicht verändert	5.3	1	I
			Ohne Multiple-Choice			
5.4	3	0,46	Aufgabe leicht verändert	5.4	3	II
5.5	4	0,48	Aufgabe leicht verändert	5.5	4	II
67.			Thema: Einführung in die OOP / Klasse und Objekt			
67.1	1	0,45	Aufgabe leicht verändert	67.1	1	I
67.2	1	0,61	Aufgabe leicht verändert	67.2	1	I
67.3	1	0,55	Aufgabe leicht verändert	67.3	1	I
67.4	1	0,63	Aufgabe leicht verändert	67.4	1	I
67.5	1	0,79	Aufgabe leicht verändert	67.5	1	I
67.6	1	0,76	Aufgabe leicht verändert	67.6	1	I
			Ohne Multiple-Choice			
67.7	2	0,82	Aufgabe leicht verändert	67.7	2	II
67.8	3	0,76	Aufgabe leicht verändert	67.8	3	II
67.9	2	0,76	Aufgabe leicht verändert	67.9	2	II
67.10	4,5	0,81	Aufgabe leicht verändert	67.10	4,5	II
67.11	2,5	0,81	Aufgabe leicht verändert	67.11	2,5	II
8.			Thema: Vererbung (Multiple-Choice)			
8.1	1	0,92	Aufgabe leicht verändert	8.1	1	I
8.2	1	0,58	Aufgabe leicht verändert	8.2	1	I
8.3	1	0,89	Sehr verändert, z. B. keine Multiple-Choice-Antworten	8.3a	4	II
			Ohne Multiple-Choice			
8.4	7	0,64	Aufgabe stark verkürzt	8.3b	4	II

Abbildung 64: Nachhaltigkeitstest im Kontext zum Vergleichstest (Teil 1)

VT Aufgabe	Punkte	P <sub>i</sub> OOP-Later	Thema des Vergleichstests (VT) bzw. des Nachhaltigkeitstests (NT) in der Hauptstudie	NT Aufgabe	Punkte	AF
9.			Thema: Assoziation (Multiple-Choice)			
9.1	1	0,55	Aufgabe leicht verändert	9.1	1	I
9.2	1	0,68	Aufgabe leicht verändert	9.2	1	I
			Ohne Multiple-Choice			
9.3a	4	0,73	Aufgabe vollständig geändert	9.3	3	II
9.3b	4	0,18	Aufgabe verändert	9.4	5	II
10.			Thema: Dynamische OOP			
10.1	2	1,00	Aufgabe erschwert	10.1	2	I
10.2	3	0,60	entfällt			
10.3	5	0,65	Aufgabe fast gleich, aber Lösung aufwändiger	10.2	8	II
11.			Thema: Objektorientierte Modellierung			
11.1	10	0,59	Aufgabe leicht verändert	11.1	10	II
	<b>Summe 110</b>	<b>Ø 0,61</b>			<b>Summe 110</b>	

Abbildung 65: Nachhaltigkeitstest im Kontext zum Vergleichstest (Teil 2)

Somit sind alle drei Tests, die im Studiendesign für die Hauptstudie (vgl. Abbildung 24) vorgesehen waren, entwickelt worden. Die angewendete Vorgehensweise zur Frage- bzw. Test-Erstellung soll nun im nächsten Kapitel kritisch reflektiert werden.

#### 4.3.5 Kritische Reflexion der Tests, speziell des Vergleichstests der Hauptstudie

Bevor die Tests kritisch reflektiert werden, soll zuerst die Vorgehensweise bei der Entwicklung des Vergleichstests (für die Hauptstudie) noch einmal kurz dargestellt werden:

Aus einer vorhandenen Themenstellung wurden einzelne Feinlernziele bzw. Fachkompetenzen formuliert. Dies ist im Sinne einer zielgerichteten und systematischen Konstruktion von Testaufgaben: Vor der Entwicklung der Aufgabe muss die zu überprüfende Kompetenz festliegen. Der Indikator für das Erreichen eines Feinlernziels ist, dass ein Schüler ein entsprechendes Item bzw. eine entsprechende Frage vollständig richtig beantworten kann.

Ein Beispiel zum Thema T2, welches wie folgt lautet:

„Einführung in den Java-Editor, Variablen (Attribute), fundamentale Datentypen, Steuerstruktur Sequenz (inkl. Struktogramm)“.

Aus diesem allgemeinen Thema wurden Feinlernziele entwickelt, u. a.:

„Die Schüler sollen Variablen deklarieren können, Variablen sinnvolle Werte zuordnen können und die entsprechende Java-Syntax benutzen können.“.

Diese können auch als entsprechende Fachkompetenzen formuliert werden, die man auch als Indikatoren ansehen kann, z. B.:

„Die Schüler deklarieren Variablen, ordnen den Variablen sinnvolle Werte zu und benutzen die entsprechende Java-Syntax.“.

Danach erfolgte die Überprüfung der Indikatoren mit Hilfe von Items bzw. Fragen, z. B.:

„Gegeben ist die folgende lückenhafte Variablendeklaration:  
`char a = ____; ____ b = 7.0f; ____ c = (5 == 7);`  
Ergänzen Sie den Quelltext in den 3 Lücken!“.

Das erste Item überprüft die Fähigkeit, einen Char-Wert syntaktisch richtig einer Char-Variablen zuzuordnen (`char a = 'a'`). Das zweite Item überprüft die Fähigkeit, einen Float-Wert in Java zu identifizieren (`float b = 7.0f`). Beim dritten Item müssen die Schüler wissen, dass ein Vergleich zweier Integer-Werte einen booleschen Wert zurückliefert und dass Boolesche-Werte in Java nur Boolean-Variablen zugewiesen werden dürfen (`boolean c = (5 == 7)`).

Diese Vorgehensweise ist didaktisch richtig, das „Endprodukt Vergleichstest“ muss aber kritisch reflektiert und untersucht werden, z. B. auch im Hinblick auf 3 wichtige Gütekriterien der Testtheorie (vgl. die zehn Kriterien nach [KJ03, S. 408-415]).

#### 1. Gütekriterium Validität:

„Ein Test gilt dann als valide [...], wenn er das Merkmal, das er messen soll, auch wirklich misst und nicht irgendein anderes.“ [MK08, S. 13].

Das Ziel erreicht man u. a., indem eine Item-Menge pro Indikator gebildet wird, die ein möglichst valides Ergebnis im Hinblick auf das Erreichen des vorgesehenen Feinlernziels erzielt. Man sieht aber, dass die drei Items nur ein ganz kleiner Ausschnitt aus einer sehr großen möglichen Item-Menge sind. Das erste Item z. B. misst sehr genau die Fähigkeit, ob ein Char-Wert syntaktisch richtig einer Char-Variablen zugeordnet werden kann und ist damit sehr valide im Kontext des Merkmals „Kann einer Char-Variablen ein sinnvoller Wert zugewiesen werden?“. Es wird also genau dieses Merkmal gemessen und nicht irgendein anderes. Aber das Merkmal heißt in Wirklichkeit „Können Variablen (unterschiedlicher Datentypen) sinnvolle Werte zugewiesen werden?“. Das erste Item fragt somit nur einen von acht möglichen fundamentalen Datentypen in Java ab. Der Grund hierfür ist, dass zusätzlich andere Gütekriterien der Testtheorie wie „Testökonomie“ (vgl. [MK08, S. 21]) und „Zumutbarkeit“ (vgl. [MK08, S. 22]) zum Tragen kommen: Die zeitlichen Ressourcen des Tests sind begrenzt und die zu testenden Schüler sollten nicht über Gebühr belastet werden. Dies ist auch der Grund, warum es relativ oft sogar nur ein Item pro Indikator gibt.

#### 2. Gütekriterium Objektivität:

„Ein Test ist dann objektiv, wenn er dasjenige Merkmal, das er misst, unabhängig von Testleiter, Testauswerter und von der Ergebnisinterpretation misst.“ [MK08, S. 8].

Die Objektivität des Vergleichstests ist relativ hoch, da die Aufgabenstellungen eindeutig (und klein) sind. Der Test kann von jedem beliebigen Lehrer (Testleiter) beaufsichtigt werden, da Fragen während des Tests nicht beantwortet werden. Die Testauswertung ist durch die klar formulierte Lösung (inkl. der Hinweise auf Teilpunkte) und die Kleinschrittigkeit der Lösung auch objektiv. Sie könnte von jedem Informatik-Lehrer vorgenommen

werden und würde zu den (fast) gleichen Ergebnispunkten führen, da es nur ganz geringe Interpretations-Spielräume bei der Bewertung gibt.

### 3. Gütekriterium Reliabilität:

„Ein Test ist dann reliabel (zuverlässig), wenn er das Merkmal, das er misst, exakt, d. h. ohne Messfehler, misst.“ [MK08, S. 11].

Wenn das schon o. g. Item für das Merkmal „Kann einer Char-Variablen ein sinnvoller Wert zugewiesen werden?“ betrachtet wird, dann kann man davon ausgehen, dass die Testaufgabe exakt das Merkmal (ohne Messfehler) misst. Die Wiederholung dieser Teiltistung würde an derselben Testperson unter gleichen Bedingungen zum gleichen Ergebnis führen. Der Reliabilitätskoeffizient wäre dann idealtypisch der Wert 1,0 [MK08, S. 11]. Nun wird dies nicht für jedes Item hundertprozentig gelingen, aber der Gesamt-Reliabilitätskoeffizient wird sicherlich über dem Wert 0,7 liegen, der einen guten Test auszeichnet [MK08, S. 11]. Einen eindeutigen Beweis gibt es aber nicht, da das Ausmaß der Reliabilität über bestimmte Verfahren der klassischen Testtheorie (z. B. Retest-Reliabilität, Paralleltest-Reliabilität, Testhalbierungs-Reliabilität, innere Konsistenz) in dieser Studie nicht explizit bestimmt wurde. Der Nachhaltigkeitstest kann in diesem Kontext als eine Art Paralleltest angesehen werden, da er aus inhaltlich möglichst ähnlichen Items wie der Vergleichstest besteht. Allerdings ist er zeitlich (mit Absicht) weit auseinander vom Vergleichstest, da er ja nicht zur Messung des Reliabilitätskoeffizienten konstruiert wurde, sondern zur Messung der Nachhaltigkeit. Immerhin eliminiert bzw. kontrolliert er durch die „Zwillingsitems“ etliche reliabilitätsverändernde Einflüsse (z. B. Übungs- und Erinnerungseffekte) [MK08, S. 12].

Neben den Gütekriterien nach Kubinger und Jäger [KJ03, S. 408-415] gibt es z. B. die folgenden weiteren Forderungen an die Aufgaben bzw. Items eines Tests.

### 4. Vermeidung von Reihenfolgeeffekten [Sch09, S.216-218]

Reihenfolgeeffekte können bei der Reihenfolge der Fragen (beim Vergleichstest sowohl die Reihenfolge der Themen als auch die Reihenfolge innerhalb der Themen) und bei der Aufzählung von mehreren Antwortmöglichkeiten in Multiple-Choice-Fragen auftreten und zur Verzerrung der Ergebnisse führen. Die für den Vergleichstest gewählte Reihenfolge der Themen ergibt sich aus deren 3-Gliedrigkeit: Zuerst werden so genannte prozedurale Themen (VT1, VT2 und VT3), dann Misch-Themen mit unterschiedlicher Zuordnungsmöglichkeit (VT5 und VT6) und am Ende so genannte OOP-Themen (VT67, VT8 und VT9) abgefragt. Aus motivationaler Überlegung heraus wird oft innerhalb eines Themas mit einem Item mit geringer Schwierigkeit (leichte Multiple-Choice-Aufgabe) begonnen [MK08, S. 76]. Dies ist der erste Grund, warum auf zufallsgesteuerte Präsentation der Items verzichtet wird. Der andere Grund ist, dass der Aufwand viel zu hoch gewesen wäre. Da sich ein evtl. vorhandener Reihenfolgeeffekt auf beide Klassen gleichmäßig auswirkt, können vorhandene Unterschiede im Lernerfolg trotzdem ermittelt werden. Evtl. muss man aber vorsichtig (z. B. bei der Interpretation der Schwere der einzelnen Themen) sein, da beim letzten Thema T9 die Schülerinnen und Schüler schon ca. 88-90 Minuten Test (vgl. Abbildung 62) hinter sich haben.

## 5. (Stochastische) Unabhängigkeit der Aufgaben voneinander [PTK10, S. 60]

Die Aufgaben des Vergleichstests der Hauptstudie sind vollständig unabhängig voneinander. Keine einzige Aufgabe baut auf der Lösung einer anderen Aufgabe auf. Dies gilt auch für den Vortest und den Nachhaltigkeitstest.

## 6. Itemschwierigkeit im mittleren Bereich und hohe Itemtrennschärfe [BD02, S. 218-219]

Bei einer einfachen Itemanalyse sollte ein Niveautest vorliegen, d. h. alle Aufgaben sollen zeitlich von den Schülerinnen und Schülern in Angriff genommen werden können [LR98, S. 218]. Dies wurde durch die Vorgaben zur Bearbeitungszeit in Kapitel 4.3.2 gewährleistet. Bei der Itemanalyse gibt es aber insofern ein Problem, da diese meist auf einen Versuchstest angewendet wird, um dann den endgültigen Test zu entwickeln. Als Versuchstest dieser Studie war der Vergleichstest der Vorstudie vorgesehen. Wegen der großen Veränderung des Vergleichstests in der Hauptstudie (aufgrund der Neuschneidung der abgefragten Themen) wurde aber auf eine größere Itemanalyse verzichtet, da diese fast ohne Wert gewesen wäre. Es wurde nur geprüft, ob übernommene oder teilweise übernommene bzw. abgeänderte Items ursprünglich im gewünschten mittleren Schwierigkeitsbereich lagen:

„Im allgemeinen werden Itemschwierigkeiten im mittleren Bereich (zwischen 0.2 und 0.8) bevorzugt.“ [BD02, S. 218].

In der Abbildung 57 und in der Abbildung 58 sind die Itemschwierigkeiten (Popularitätsindex  $P_i$ ) des Vergleichstests der Vorstudie aufgeführt. Eine größere Itemanalyse für den Vergleichstest der Hauptstudie auszuführen, wäre möglich gewesen, hätte aber fast keine Konsequenzen gehabt, da der Nachhaltigkeitstest dem Vergleichstest sehr ähnlich sein musste. Ziel war es, die Fragen bzw. Antworten vergleichbar zu halten, um die Ergebnisse vergleichen zu können (siehe Kapitel 4.3.4).

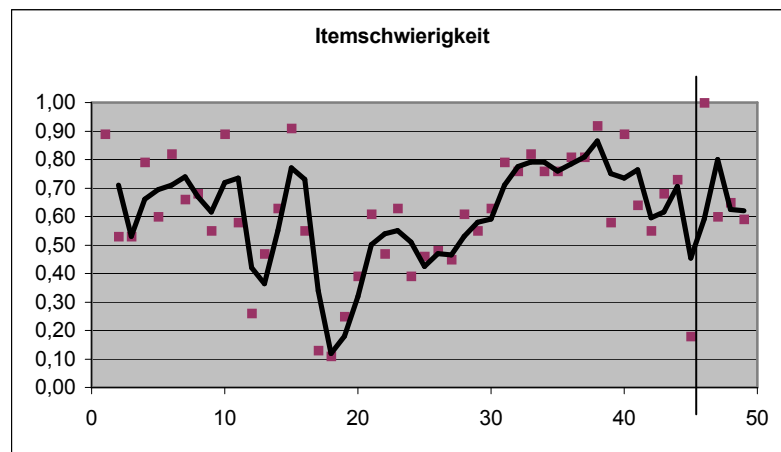


Abbildung 66: Itemschwierigkeit der 49 Aufgaben des Vergleichstests (Hauptstudie, OOP-Later)

(in der gegebenen Reihenfolge des Tests, mit Trendlinie nach Mittelwerten, die letzten 4 Aufgaben werden gesondert abgefragt)

Wenn im Folgenden eine Analyse der Schwierigkeit der Items des Vergleichstests der Hauptstudie erfolgt (siehe Abbildung 66 und Abbildung 67), dann dient sie dem nachträglichen Beweis, dass die Itemschwierigkeiten im gewünschten mittleren Bereich lagen. Auswirkungen auf den Nachhaltigkeitstest erfolgen nur bei Items, die die unerwünschte Schwierigkeit 0 (kein Schüler hat einen Punkt erreicht) und 1 (alle Schüler haben die Aufgabe



vollständig gelöst) haben (siehe Kapitel 4.3.4, speziell: Abbildung 65). Wegen der mangelnden Auswirkungen wurde auch keine Trennschärfenanalyse durchgeführt, es wird nur darauf verwiesen, dass Items mit mittleren Schwierigkeiten die höchste Trennschärfe besitzen [BD02, S. 219].

Die Abbildung 66 zeigt den zeitlichen Verlauf der Itemschwierigkeit im Vergleichstest der Hauptstudie. Dabei müssen zwei Sachverhalte beachtet werden. Erstens liegt keine zeitlich lineare Skala vor, da die einzelnen Aufgaben sehr unterschiedliche Vorgaben für die Lösungszeit haben. Zweitens werden nur die Ergebnisse der OOP-Later-Klasse dargestellt. Die Begründung hierfür findet sich in Kapitel 6.3.2. Die genauen Itemschwierigkeiten für beide Klassen sind in Anhang N aufgeführt.

Der Vergleichstest wurde in zwei Etappen geschrieben: einmal die Themenaufgaben 1-9 (Aufgaben 1-45) in 100 Minuten und einmal die beiden Zusatzaufgaben 10 und 11 (Aufgaben 46-49) in 30 Minuten. Für den Vergleichstest ergibt sich das folgende, im Nachhinein ermittelte Bild: Die Schülerinnen und Schüler sind bei den ersten 16 von 45 Aufgaben des ersten Teils der Vergleichsarbeit mit einem mittleren Itemschwierigkeitsgrad konfrontiert worden, der leicht über dem Gesamt-Durchschnitt von 0,61 liegt. Dies ist für eine Einstiegsphase optimal, um die Schüler zu motivieren. Danach folgte eine Phase (Aufgaben 17 bis 30), die deutlich schwieriger war, die einerseits spät genug kam, so dass die Schüler schon wach bzw. motiviert waren, aber andererseits früh genug kam, so dass die Schüler noch nicht ermüdet waren. Die anschließende Phase könnte man als Ruhephase bezeichnen (Aufgaben 31 bis 40) bis in der letzten Phase (Aufgaben 41 bis 44) wieder die Itemschwierigkeit im mittleren Bereich lagen. Leider war die letzte Aufgabe (Aufgabe 45) für die meisten Schüler zu schwer. Nachträglich betrachtet ist dies trotzdem kein schlechtes Bild für einen Testablauf. Bei dem zweiten Teil der Vergleichsarbeit (Aufgaben 46-49) wurden nach einem zu leichten Einstieg (Itemschwierigkeit 1,0) den Schülerinnen und Schülern Aufgaben mit einem durchgängig mittleren Schwierigkeitsbereich gestellt.

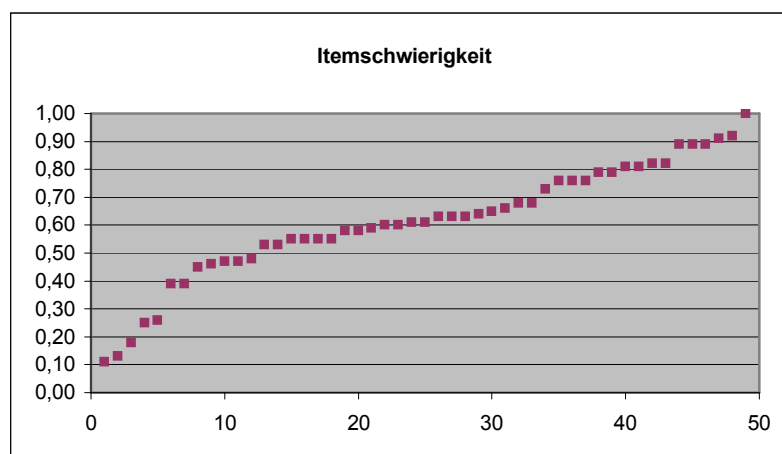


Abbildung 67: Itemschwierigkeit der 49 Aufgaben des Vergleichstests (Hauptstudie, OOP-Later)

(nach Schwierigkeit sortiert, vgl. Abbildung 66)

Die Abbildung 67 zeigt alle Itemschwierigkeiten im Vergleichstest der Hauptstudie bezogen auf die OOP-Later-Klasse und geordnet nach der Schwere. Man sieht, dass es eine große Spannbreite gibt, dass die meisten Schwierigkeiten im gewünschten mittleren Bereich zwischen 0,2 und 0,8 liegen und dass es trotzdem Items im schweren und leichten Bereich gibt. Für eine gute Differenzierung auch in den Extrembereichen, also in den oberen und unteren Leistungsbereichen, ist es nämlich wichtig, auch einige hohe und geringe Itemschwierigkeiten im Test zu behalten [LR98, S. 115]. Aus Abbildung 68 wird deutlich, dass diese

Ausdifferenzierung für die leistungsschwächeren Schüler (mit zehn Aufgaben) höher ist als die für die leistungsstarken Schüler (mit drei Aufgaben).

Ein Grund für diese Ausdifferenzierung ist, dass die mittlere Itemschwierigkeit bei 0,61 und nicht bei 0,5 liegt, somit der gesamte Schwierigkeitsbereich in Richtung „mittelleichte“ Aufgaben verschoben ist. Eine mittlere Itemschwierigkeit von 0,5 (wenn auch die Punkte so zugeordnet wären, dass die erreichte Gesamtpunktzahl der Schülerinnen und Schüler nur 50% der möglichen Gesamtpunktzahl betrüge) ist im Schulbereich gar nicht erwünscht. 50% Leistung ergibt im Beruflichen Gymnasium in Berlin die Note 4 (siehe Abbildung 153). Schriftliche Arbeiten sollten aber in der Regel besser ausfallen, so dass ein mittlerer Schwierigkeitsindex von 0,61 absolute Berechtigung hat.

Itemschwierigkeit	Aufgabenanzahl	Punkte
0,0-0,2 (schwer)	3	6
0,2-0,8 (mittel)	36	81
0,8-1,0 (leicht)	10	23
<b>Summe</b>	<b>49</b>	<b>110</b>

Abbildung 68: Itemschwierigkeit der Aufgaben des Vergleichstests (Hauptstudie, OOP-Later)

Die Punkteverteilung deckt sich größtenteils mit den definierten Anforderungsbereichen aus Abbildung 60. Dort waren dem Anforderungsbereich I (leicht) 23 Punkte und dem Anforderungsbereich II (mittel) 87 Punkte zugeteilt worden. Die vorgenommene Definition bzw. Einteilung wird somit durch die Itemschwierigkeits-Analyse bestätigt (von der Abweichung im Anforderungsbereich III (schwer) abgesehen).

Soweit die kritische Reflexion des Vergleichstests der Hauptstudie. Für den Vortest gelten alle Aussagen, die für den Vergleichstest gemacht wurden, da er nur ein stark verkürzter Vergleichstest ist (siehe Kapitel 4.3.3).

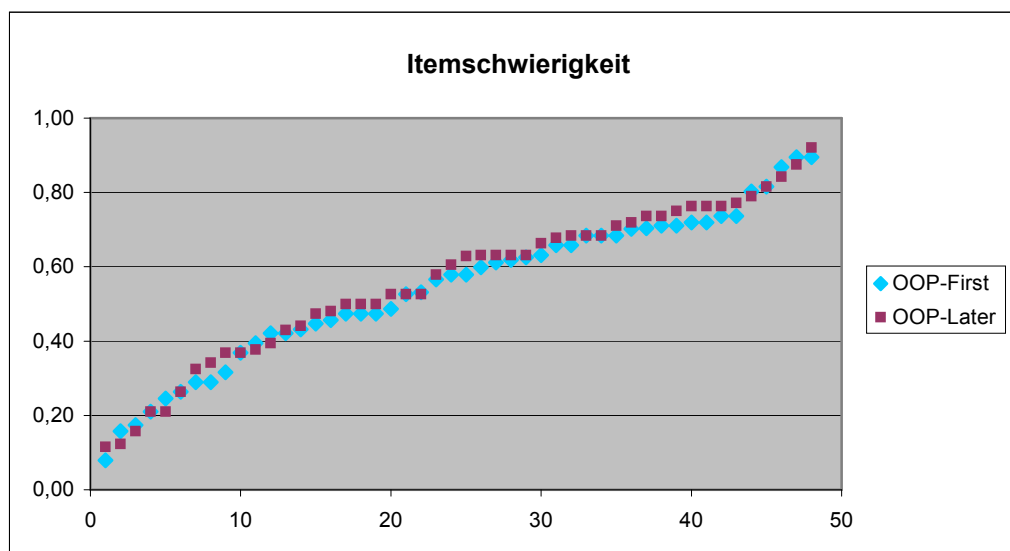


Abbildung 69: Itemschwierigkeit der 48 Aufgaben des Nachhaltigkeitstests (Hauptstudie)

(nach Schwierigkeit sortiert)

Auch für den Nachhaltigkeitstest lassen sich vergleichbare Aussagen treffen, da er dem Vergleichstest sehr ähnlich ist, sowohl von der Art und dem Umfang der Aufgaben als auch von den Anforderungsbereichen (vgl. Kapitel 4.3.4).

Eine Itemschwierigkeits-Analyse hat in Bezug auf einen Mittelwert wenig Sinn (hier kommt ja der zeitliche Aspekt des Vergessens von Lerninhalten zum Tragen). Es soll aber durch die Abbildung 69 überprüft werden, ob wieder eine ausreichend große Spannbreite bei den Itemschwierigkeiten vorhanden ist und ob es gelungen ist, den Index für die Frage 10.1 (siehe Abbildung 65) von 1,00 auf einen kleineren Wert zu senken.

Die (nicht relevanten) durchschnittlichen Itemschwierigkeiten betragen 0,54 für die OOP-First-Klasse und 0,56 für die OOP-Later-Klasse. Der Verlauf der eingezeichneten Punkte ist sehr ähnlich. Die Spannbreite der Ergebnisse der OOP-Later-Klasse im Nachhaltigkeitstest ist vergleichbar mit der Spannbreite aus dem Vergleichstest (siehe Abbildung 67), nur dass es zu einer Parallelverschiebung nach unten (also zu geringeren Itemschwierigkeitswerten) gekommen ist. Unter der Annahme, dass der Nachhaltigkeitstest die gleichen Anforderungen an die Schülerinnen und Schüler gestellt hat wie der Vergleichstest, ist dies ein erster Beleg für das Vergessen von einigen Lerninhalten aufgrund fehlender Nachhaltigkeit.

## **4.4 Zusammenfassung**

Im diesem 4. Kapitel wurden für alle neun Unterrichts-Themen die Lernziele bzw. Kompetenzen formuliert (siehe Kapitel 4.1). Danach wurden zwei zusätzliche Kategorien (mit themenübergreifenden Fachkompetenzen) abgeleitet (siehe Kapitel 4.2), die Zusatzthemen „dynamische OOP“ und „OOM“. Jedem Lernziel wurde dabei mindestens ein Indikator (bzw. eine passende Frage) zugeordnet, mit dessen Hilfe das Erreichen des Lernziels überprüft werden kann. Für den Vergleichstest der Hauptstudie, der sich vom Vergleichstest der Vorstudie stark unterscheidet (siehe Kapitel 4.3.1), wurden die Fragen aufsummiert und ihnen Wertungspunkte und Schwierigkeitsgrade (Anforderungsbereiche) zugeordnet. Durch die Einteilung in so genannte Anforderungsbereiche konnte analysiert werden, ob die Fragen zu den einzelnen Themen ungefähr gleich schwer sind (siehe Kapitel 4.3.2). Zusätzlich wurde die Gesamtzeit für den Vergleichstest bestimmt. Danach wurden der Vortest (siehe Kapitel 4.3.3) und der Nachhaltigkeitstest (siehe Kapitel 4.3.4) aus dem Vergleichstest heraus entwickelt, teilweise unter Berücksichtigung der so genannten Itemschwierigkeit. Im Kapitel 4.3.5 wurde dann die Testgenerierung, die didaktischen Prinzipien folgte, einer kritischen Reflexion im Hinblick auf die Prinzipien der allgemeinen Testtheorie unterzogen.



## 5 Subjektives Erleben des Unterrichts: Dimensionen, Frage-Bereiche und Fragestellungen

In diesem Kapitel wird das subjektive Erleben des Unterrichts in drei Dimensionen (siehe Kapitel 5.1.1) und vier Bereiche (siehe Kapitel 5.1.2) aufgeschlüsselt, so dass letztlich eine 3x4-Frage-Matrix entsteht (siehe Kapitel 5.1.3). Ziel soll es sein, über einen Fragebogen Erlebnisdaten der Schülerinnen und Schüler zu generieren. Hierfür müssen Fragen für die emotionale Dimension (siehe Kapitel 5.2.1), die kognitive Dimension (siehe Kapitel 5.2.2) und die motivationale Dimension (siehe Kapitel 5.2.3) entwickelt werden. Im Kontext zu den Fragen muss überlegt werden, welche Antwortmöglichkeiten erlaubt werden (siehe Kapitel 5.1.4), wie viele Fragen es geben und wo der Schwerpunkt der Fragen liegen soll (siehe Kapitel 5.1.3).

### 5.1 Vorgaben für den Fragebogen

Im Kapitel 3.3.13 wurden schon einige Vorgaben für den zu entwickelnden Fragebogen gemacht: Der Fragebogen soll Fragen zu den drei verschiedenen Erlebnis-Dimensionen enthalten, er soll nach jedem der neun Themen an die Schülerinnen und Schüler ausgegeben werden, er soll nicht zu lang sein, er darf auch Antworten zum Ankreuzen haben und sich auf verschiedene Bereiche (z. B. letztes Thema oder das Fach Informatik) beziehen.

Diese Vorgaben sollen im Folgenden vertieft und ergänzt werden.

#### 5.1.1 Erlebnis-Dimensionen

In den Kapiteln 2.2, 2.4.3 und 3.3.13 wurden u. a. die Arbeiten von Sembill (vgl. [Sem92] und [Sem05]) und Seifried (vgl. [Sei04] und [SK06]) in Verbindung mit der zweiten Forschungsfrage zitiert und erste Vorgaben für einen Fragebogen zum subjektiven Erleben des Unterrichts gemacht.

Seifried [Sei04] nimmt in einer empirischen Untersuchung im Rechnungswesenunterricht im Kontext zu vorausgehenden Arbeiten von Sembill [u. a. Sem92] so genannte Erlebnisdaten der Schülerinnen und Schüler auf und unterscheidet dabei drei Erlebnis-Dimensionen:

- die emotionale Dimension
- die kognitive Dimension
- die motivationale Dimension

Den Erlebnis-Dimensionen werden dabei Items zugeordnet [Sei04]:

„Die drei Dimensionen (Emotion, Kognition und Motivation) wurden wie folgt operationalisiert:

Emotionaler Bereich:           „Fühle mich Ernst genommen“  
  „Fühle mich wohl“

Kognitiver Bereich:	„Finde Anforderungen hoch“ „Verstehe, worum es geht“
Motivationaler Bereich:	„Ich kann mitgestalten“ „Bin interessiert““ [Sei04, S. 141].

Die Erlebnisdaten können durch die Abfrage einzelner Items gewonnen werden, z. B. wird eine Aussage vorgegeben, die dann die Schülerinnen und Schüler ergänzen sollen:

- „Ich fühlte mich speziell beim letzten Thema eher ...“ („sehr wohl“ bis „sehr unwohl“)
- „Ich empfand die Anforderungen beim letzten Thema an mich eher als ...“ („sehr hoch“ bis „sehr niedrig“)
- „Ich fand das letzte Thema ...“ („sehr interessant“ bis „sehr langweilig“)

Subjektive Fragen mögen auf den ersten Blick nicht unbedingt seriös erscheinen, sind aber abgeleitet aus Untersuchungen [z. B. Sem05, WS01 und SK06], die besagen, dass das emotional-motivationale Erleben des Unterrichtsgeschehens einen Einfluss auf den Lernerfolg nimmt.

Auch in anderen Bereichen wird mit subjektiven Fragebögen gearbeitet, z. B. im Leistungssport Fußball:

„Viele Klubs arbeiten mit einem subjektiven Fragebogen, um wöchentlich auf die Befindlichkeit eines Spielers reagieren zu können. Damit werden ständig alle möglichen subjektiven Wohlfühlfaktoren abgefragt. Neben dem körperlichen Befinden spielen private und psychische Faktoren eine Rolle.“ [Pet10].

Der für diese Studie zu entwickelnde subjektive Fragebogen soll sich an der Einteilung des subjektiven Erlebens des Unterrichts in die drei Erlebnis-Dimensionen halten [ES10a]:

„We distinguish between three dimensions:

The *emotional-affective climate* describes feelings and emotions like being comfortable or feeling overworked. The question is, how do learners experience the course, the current topic and their learning progress?

The *cognitive climate* describes thoughts and perceptions towards the course, like e.g. difficult to learn, important to learn, interesting to learn. The question is, how do pupils experience the difficulty of the topic and specific obstacles for learning this topic?

The *motivational climate* describes the willingness to learn (I am motivated to learn). The question is, what is the learners' motivation and interest regarding the course, or the specific topic?

All aspects of the emotional climate are from the internal point of view, as opposed to external measurements like tests of learning outcome. That is, each item for each of the three scales asks for the students' own perception of some aspect of the course.” [ES10a, S. 109, kursive Schrift im Original].

Hier wird sowohl vom Kurs (also vom Schulfach Informatik) gesprochen als auch vom letzten Unterrichtsthema. Diese Kategorisierung der Fragen wird im folgenden Kapitel erläutert.

## 5.1.2 Frage-Bereiche

Der zu entwickelnde Fragebogen soll sich an der Einteilung des subjektiven Erlebens des Unterrichts in die drei Erlebnis-Dimensionen halten. Ein Problem liegt m. E. aber darin, dass, wenn man Erlebnisdaten des Informatikunterrichts abfragt, es nicht nur die beiden Einflussvariablen „letztes Thema“ und „fachdidaktische Vorgehensweise“ gibt (vgl. Kapitel 3.3.5). Es gibt auch andere Variablen (evtl. sogar Störvariablen), die Einfluss auf die Antworten nehmen können: das private Umfeld, die anderen Fächer, die Schule, der zeitliche Kontext in der Schule (Anfang oder Ende des Schuljahres, vor oder nach den Ferien etc.) und der unterrichtende Lehrer. Daher sollen sich die Fragen auf möglichst viele dieser Variablen beziehen [vgl. ES10a, S. 109]:

- Fragen zur Schule und zum privaten Umfeld: Wie geht es den Schülern in ihrem privaten Umfeld und in der Schule? Haben sie Stress, fühlen sie sich wohl?
- Fragen zum Fach Informatik: Wie finden die Schüler das Fach Informatik? Haben sie Stress, fühlen sie sich wohl?
- Fragen zum Lehrer. Diese Fragen sind wichtig, um herauszufinden, ob die Lehrer gleichartig waren: Wie gefällt den Schülern ihr Informatik-Lehrer? Hat der Lehrer sie unterstützt?
- Fragen zum letzten Thema. Dies ist der größte bzw. wichtigste Fragebereich. Hier sollen die Erlebnisdaten bezogen auf das jeweils letzte der neun möglichen Themen erhoben werden: Wie wohl haben sich die Schüler beim letzten Thema gefühlt? Wie schwer war das letzte Thema?

## 5.1.3 Kategorisierung der Fragen mit Hilfe einer Matrix

Die drei Erlebnis-Dimensionen im Kontext zu den vier Frage-Bereichen können nun mit Hilfe einer Matrix (siehe Abbildung 70) dargestellt werden.

	<b>Emotionale Dimension</b>	<b>Kognitive Dimension</b>	<b>Motivationale Dimension</b>
<b>Fragen zur Schule bzw. zum privaten Umfeld</b>			
<b>Fragen zum Fach Informatik</b>			
<b>Fragen zum jeweiligen Lehrer</b>			
<b>Fragen zum letzten Thema</b>			

Abbildung 70: Erlebnis-Dimensionen und Frage-Bereiche als Matrix

Diese Matrix soll durch die Fragen (siehe Kapitel 5.2) des Fragebogens gefüllt werden. Allerdings muss angepasst werden, dass die Schülerinnen und Schüler nicht mit zu vielen Fragen konfrontiert werden [ES10a]:

„In order to gain data on this internal view a questionnaire was developed, and answered by each student for each of the topics taught. The problem was, not to overwhelm students with questions, in order to get an acceptable

rate of answers throughout the study (which lasted one year) – and on the other hand to gather enough information to shed light on the research questions.” [ES10a, S. 109].

Wenn man die Vorgabe macht, dass die Fragen auf ein Blatt Papier (mit Vor- und Rückseite) passen sollen, werden es wohl durchschnittlich nur zwei Fragen pro Matrix-Element sein, also ca. 24 Fragen. Einerseits sind die Erlebnisdaten umso genauer, je mehr Fragen bzw. Items es gibt. Andererseits hatte z. B. die Untersuchung von Seifried und Klüber [SK06] zum Unterrichtserleben von Schülerinnen und Schüler nur sechs Fragen (aufgeführt in Kapitel 5.1.1). Es sollten nicht zu viele Fragen sein, damit die Schülerinnen und Schüler nicht die Motivation verlieren an der Studie teilzunehmen (Gütekriterium Zumutbarkeit).

Die Anzahl der Items pro Matrix-Element wird unterschiedlich festgelegt, weil der Schwerpunkt des zweiten Aspekts der Forschungsfrage (inkl. der dahinterstehenden Thesen) einerseits auf der Messung eines Wohlfühl- bzw. Stressfaktors (beides emotionale Faktoren) liegt und andererseits auf einer Zuordnung der Erlebnisdaten zum letzten Thema basiert. Die Erlebnisdaten können dann mit den Ergebnissen der Tests verglichen werden, die ja auch themenbasiert sind. Somit soll der Schwerpunkt der Fragen in der Zeile „letztes Thema“ und in der Spalte „emotionale Dimension“ liegen (siehe farbliche Hervorhebung in Abbildung 70). Dort sollen pro Matrix-Element in der Regel mehr als zwei Fragen entwickelt werden. Dies bedeutet aber, dass viele andere Matrix-Elemente dann nur mit einem Item auskommen (müssen).

Vor der Entwicklung der Fragen für alle zwölf Matrix-Elemente muss noch geklärt werden, welche Antwortmöglichkeiten die Schülerinnen und Schüler besitzen sollen.

#### 5.1.4 Skalierung der Antwortmöglichkeiten

Bei den Antwortmöglichkeiten auf die einzelnen Fragen muss eine Entscheidung getroffen werden: Sollen diese frei sein oder skaliert? Die Wahl fällt auf skalierte Antwortmöglichkeiten, da diese eine schnellere bzw. effizientere Auswertung ermöglichen. Diese haben auch den Vorteil, dass der Fragebogen durch die Schülerinnen und Schüler schneller beantwortet werden kann, ihnen weniger Arbeit bereitet und somit mehr zumutbar ist (Gütekriterium Zumutbarkeit, vgl. Kapitel 4.3.5 und Kapitel 5.1.3).

Welcher Skalentyp soll aber in diesem Fall verwendet werden (vgl. [BL03, S. 26-29] und [SH09, S. 16-19])? Ich habe mich für eine so genannte Likert-Skala entschieden, bei der die Antworten (z. B.: „nicht“, „wenig“, „mittel“, „ziemlich“, „stark“) den natürlichen Zahlen von 1 bis 5 zugeordnet werden. Dies haben auch schon Lahtinen, Ala-Mutka und Järvinen [LAJ05] in ihrer Studie „A Study of the Difficulties of Novice Programmers“ gemacht:

„The questions asked the respondents to grade programming aspects and concepts on a five-point scale from *very easy to learn* (1) to *very difficult* (5).“ [LAJ05, S. 15].

Allerdings gebe ich nur das Minimum bzw. Maximum an, damit die Schüler die Skala von 1 bis 5 als Intervallskala betrachten, also eine Skala mit konstanten Abständen. Dann kann mit Maßzahlen wie Mittelwerten oder Standardabweichungen gerechnet werden. Damit befinde ich mich im Einklang mit der o. g. Untersuchung,



die auch Mittelwertberechnungen vornimmt [LAJ05, S. 16, Table 1]. Wäre die Rangskala nonmetrisch (Stufenabstände ungleich, z. B. bei Schulnoten) entstehen so genannte Ordinaldaten, für deren Bewertung eher Auswertungen im Hinblick auf Minimum, Maximum, Medianwert und ausgezeichnete Quantilen gemacht werden [SH09, S. 18]. Trotzdem werden im Alltag oft auch bei Ordinaldaten Mittelwertbildungen vorgenommen (z. B. die Durchschnitts-Abschlussnote bei Zeugnissen).

Die „positivsten Antworten“ (Wert 1) des Fragebogens lauten z. B.:

- „sehr wohl“, „sehr gut“, „sehr verständlich“ bzw. „sehr interessant“.

Die „negativsten Antworten“ (Wert 5) sind genau das Gegenteil, z. B.:

- „sehr unwohl“, „sehr schlecht“, „sehr unverständlich“ bzw. „sehr langweilig“.

Somit könnte eine Antwort wie in Abbildung 71 aussehen.

sehr gut	(1)	(2)	(3)	(4)	(5)	sehr schlecht
----------	-----	-----	-----	-----	-----	---------------

Abbildung 71: Beispiel einer Skalierung einer Antwort

Die Fragen sind so konzipiert, dass bei allen Fragen der bestmögliche Zustand der Wert 1 ist.

Den Werten 2, 3 und 4 wird kein Name zugewiesen, der Wert 3 ist aber eindeutig die Mitte bzw. der Mittelwert. Es kann von einer Intervallskala gesprochen werden. Will man die Werte prozentual ausdrücken, könnten die Werte aus Abbildung 71 (wie in Abbildung 72 dargestellt) umgestellt werden.

sehr gut	(100%)	(75%)	(50%)	(25%)	(0%)	sehr schlecht
----------	--------	-------	-------	-------	------	---------------

Abbildung 72: Umsetzung der Skalierung in eine Intervallskala

Nachdem alle Vorgaben formuliert wurden, können jetzt die Fragen des Fragebogens entwickelt werden.

## 5.2 Fragestellungen

In diesem Kapitel soll die Matrix aus Abbildung 70 möglichst vollständig gefüllt werden. Dafür werden für jede der drei Erlebnis-Dimensionen Fragen zu den vier Frage-Bereichen abgeleitet. Die Nummerierung der Fragen erfolgte nachträglich durch die spätere Nummerierung auf dem realen Fragebogen. Dieser ist in die Frage-Bereiche und nicht in die Erlebnis-Dimensionen aufgeteilt.

### 5.2.1 Emotionale Fragestellungen

Laut Vorgabe aus Kapitel 5.1.3 soll ein Schwerpunkt der Fragen in der emotionalen Dimension liegen. Daher sollen im Folgenden pro Matrix-Element mindestens zwei Fragen entwickelt werden.

Bei der emotionalen Dimension wird als erstes eine Frage bzw. ein Item von Seifried (vgl. [Sei04, S. 104]) übernommen (siehe Kapitel 5.1.1): eine Frage zum so genannten Wohlfühlfaktor. Um dieses Item mehr abzusichern,

wird ein gegensätzlicher Faktor eingeführt: der Stressfaktor. Beide Faktoren können in drei der vier Frage-Bereichen ermittelt werden: im schulischen Umfeld, im Fach Informatik und im Kontext zum letzten unterrichteten Thema. Damit sind schon sechs Fragen kreiert.

- Schule (und privates Umfeld)

Der Wohlfühlfaktor und der Stressfaktor in der Schule ist sicher zeitlich abhängig von unterschiedlichsten Einflussgrößen: Erfolgs- oder Misserfolgserlebnisse in den einzelnen Fächern, Zensurenvergabe, Versetzungsproblematik, anstehende Klausuren oder Zeugnisse etc. Diese Einflussgrößen sollen durch zwei Fragen zum Wohlfühl- und zum Stressfaktor (siehe Fragen 2 und 3, Abbildung 73) ermittelt werden.

Zusätzlich wird eine Frage eingeführt, die auch zur statistischen Kontrolle dient: es wird nach dem privaten Umfeld gefragt (siehe Frage 1, Abbildung 73). Probleme im privaten Umfeld können sich natürlich negativ auf die Schülerleistungen auswirken und müssen daher durch eine Frage mit berücksichtigt werden. In der Summe (bezogen auf eine ganze Klasse) wird aber ein konstanter Wert (bezogen auf ein Jahr) bei der Beantwortung einer Frage nach dem privaten Umfeld erwartet. Auch sollten mit einer hohen Wahrscheinlichkeit beide Klassen den gleichen Mittelwert haben (siehe auch statistische Diskussion in Kapitel 7.4.6).

1. Mir geht es zur Zeit in meinem privaten Umfeld eher ...						
sehr gut	(1)	(2)	(3)	(4)	(5)	sehr schlecht
2. Ich fühle mich zur Zeit in der Schule eher ...						
sehr wohl	(1)	(2)	(3)	(4)	(5)	sehr unwohl
3. Mein Stress in der Schule ist zur Zeit eher ...						
sehr niedrig	(1)	(2)	(3)	(4)	(5)	sehr hoch

Abbildung 73: Fragen zur Schule und zum privaten Umfeld (Emotionale Dimension)

- Fach Informatik

Der Wohlfühlfaktor (vgl. [Sei04, S. 141]) und der Stressfaktor werden auch bezogen auf das Fach Informatik (Fragen 8 und 9, siehe Abbildung 74) abgefragt.

Hier soll aber ein weiteres Item erhoben werden: Wie ist die Sichtweise auf das Fach Informatik bzw. wie wird das Fach empfunden (Frage 6, Abbildung 74)? Hintergrund ist, dass die Schüler mit sehr eigenen Vorstellungen an das OSZ IMT kommen, z. B. mit Informatik das Anwenden von Programmen (Word, Excel etc.) verbinden. Wenn sie dann mit der „Programmier- und Modellierungs-Realität“ konfrontiert werden, könnte das ihr Empfinden zum Fach verändern. Da dieses Empfinden Einfluss auf den Lernerfolg nehmen kann, muss dieses Item erhoben werden.

- Lehrer

Da der Lehrer als potentielle Störgröße gilt (insbesondere die unterschiedlichen Lehrer) muss unbedingt überprüft werden, ob die Lehrer vergleichbar sind. Dazu wird zum einen ein Fragebogen des OSZ IMT am

Ende des Schuljahres herangezogen (siehe Kapitel 7.2.1), zum anderen die Einstellung der Schülerinnen und Schüler zu ihrem Lehrer im Verlaufe des Schuljahres ermittelt.

6. Das Fach Informatik (innerhalb der Technik) finde ich ...						
sehr gut	(1)	(2)	(3)	(4)	(5)	sehr schlecht
8. Ich fühle mich allgemein beim Unterricht im Fach Informatik eher ...						
sehr wohl	(1)	(2)	(3)	(4)	(5)	sehr unwohl
9. Mein Stress im Fach Informatik ist eher ...						
sehr niedrig	(1)	(2)	(3)	(4)	(5)	sehr hoch

Abbildung 74: Fragen zum Fach Informatik (Emotionale Dimension)

Die erste und wahrscheinlich wichtigste Emotion bei Schülerinnen und Schülern im Kontext zu Lehrern (siehe auch Foren im Internet wie „schülerVZ“ und „spickmich“) ist oft: „Ich finde den Lehrer ...“. Diese Emotion soll abgefragt werden (siehe Frage 7, Abbildung 75).

Dies ist aber noch kein hinreichender Indikator, ob der Lernerfolg vom Lehrer gut unterstützt wird. Daher werden zwei weitere Items hinzugefügt: Unterstützt der Lehrer die Schüler bei ihren Lernversuchen (siehe Frage 19, Abbildung 75) und sorgt er für eine gute Arbeitsatmosphäre (siehe Frage 18, Abbildung 75)?

7. Den unterrichtenden Lehrer finde ich ...						
keine Aussage	(0)					
sehr gut	(1)	(2)	(3)	(4)	(5)	sehr schlecht
18. Die Arbeitsatmosphäre in der Klasse beim letzten Thema war eher ...						
sehr gut	(1)	(2)	(3)	(4)	(5)	sehr schlecht
19. Die Unterstützung des Lehrers bei meinen Lernversuchen beim letzten Thema war ...						
keine Aussage	(0)					
sehr gut	(1)	(2)	(3)	(4)	(5)	sehr schlecht

Abbildung 75: Fragen zum Lehrer (Emotionale Dimension)

Bei den Fragen zu den beteiligten Lehrern ist aber Vorsicht geboten. Die Schüler wissen, dass man über ihren Code, der teilweise aus den Vornamen der Mutter und des Vaters besteht (siehe Kapitel 3.3.7), ihren Namen herausbekommen könnte. Die Untersuchung ist nach außen hin anonym, nicht aber nach innen. Daher muss es eine Möglichkeit geben, dass die Schüler auch keine Aussage machen.

- Thema

Der Wohlfühlfaktor (vgl. [Sei04, S. 141]) und der Stressfaktor werden jetzt auch auf das konkrete Informatik-Thema bezogen (siehe Fragen 15 und 20, Abbildung 76).

15. Ich fühlte mich speziell beim letzten Thema eher ...						
sehr wohl	(1)	(2)	(3)	(4)	(5)	sehr unwohl
20. Mein Stress beim letzten Thema war eher ...						
sehr niedrig	(1)	(2)	(3)	(4)	(5)	sehr hoch

Abbildung 76: Fragen zum letzten Thema (Emotionale Dimension)

Somit ist die Spalte „emotionale Dimension“ der Matrix aus Abbildung 70 mit Fragen gefüllt worden (siehe Abbildung 77).

	Emotionale Dimension
Fragen zur Schule bzw. zum privaten Umfeld	1,2,3
Fragen zum Fach Informatik	6,8,9
Fragen zum jeweiligen Lehrer	7,18,19
Fragen zum letzten Thema	15,20

Abbildung 77: Fragen zur emotionalen Dimension

Nach den emotionalen Fragestellungen sollen jetzt die kognitiven Fragestellungen herausgearbeitet werden.

## 5.2.2 Kognitive Fragestellungen

Auch die kognitiven Fragestellungen werden wieder den vier verschiedenen Frage-Kategorien zugeordnet. Laut Vorgabe aus Kapitel 5.1.3 reicht bis auf den Frage-Bereich „letztes Thema“ ein Item bzw. eine Frage pro Matrix-Element aus.

- Schule und Fach Informatik

Für die Schule und für das Fach Informatik wird wieder eine Frage von Seifried (vgl. [Sei04, S. 141] und Kapitel 5.1.1) übernommen: eine Frage zum Anforderungs-Empfinden (siehe Frage 5, Abbildung 78 und Frage 11, Abbildung 79).

5. Ich empfinde die Anforderungen der Schule an mich zur Zeit eher als ...

sehr niedrig      (1)      (2)      (3)      (4)      (5)      sehr hoch

Abbildung 78: Frage zur Schule (Kognitive Dimension)

11. Ich empfinde die Anforderungen des Faches Informatik an mich eher als ...

sehr niedrig      (1)      (2)      (3)      (4)      (5)      sehr hoch

Abbildung 79: Frage zum Fach Informatik (Kognitive Dimension)

- Lehrer

Bei der kognitiven Fragestellung zum Lehrer soll eins der ältesten fachdidaktischen Prinzipien abgefragt werden, das Prinzip der Anschaulichkeit (z. B. nach Comenius): Konnte der Lehrer die Lerninhalte verständlich und anschaulich darstellen (siehe Frage 17, Abbildung 80)?

17. Die Darstellung und Veranschaulichung der Lerninhalte des letzten Themas waren ...

sehr verständlich      (1)      (2)      (3)      (4)      (5)      sehr unverständlich

Abbildung 80: Frage zum Lehrer (Kognitive Dimension)

- Thema

Die letzte Fragekategorie ist m. E. sehr wichtig. Hier sollen die Schülerinnen und Schüler eine Rückmeldung zu dem letzten Thema geben. Dies ist in zweierlei Hinsicht von Bedeutung: Zum einen kann man hier aus Schülersicht Hinweise auf das Empfinden zu einem bestimmten Informatik-Thema bekommen, zum anderen können diese Daten dann später mit den Lernerfolgen verglichen werden. Ist die innere Schüler-Sicht (Thema wird z. B. als leicht empfunden oder als gut beherrscht eingeordnet) in Einklang mit der äußeren Lernerfolgskontrollen-Sicht (Fragen zum Thema können z. B. nur mangelhaft beantwortet werden).

Es wird also ein Bündel von Fragen bzw. Items erstellt. Die ersten Fragen fächern das eine Anforderungs-Item nach Seifried (vgl. [Sei04, S. 141] und Kapitel 5.1.1) weiter auf:

- Wie schwer war das Thema, wie groß waren der Stoffumfang und der nötige Arbeitsaufwand (siehe Fragen 12, 14, und 23, Abbildung 81)?

Gerade das „subjektive Schwere-Empfinden“ kann später mit den Ergebnissen der Tests („objektive Schwere“) verglichen werden.

Die nächsten Fragen beziehen sich auf den subjektiv empfundenen Lernerfolg:

- Wie gut wird jetzt das Thema beherrscht und wie groß war der Wissenszugewinn (siehe Fragen 21 und 22, Abbildung 81)?

Gerade der „subjektive Lernerfolg“ kann später mit den Ergebnissen der Tests („objektiver Lernerfolg“) verglichen werden.

12. Den Stoff des letzten Themas empfand ich als ...	sehr leicht	(1)	(2)	(3)	(4)	(5)	sehr schwer
14. Den Stoffumfang des letzten Themas empfand ich als ...	sehr niedrig	(1)	(2)	(3)	(4)	(5)	sehr hoch
21. Ich beherrsche jetzt das letzte Thema ...	sehr gut	(1)	(2)	(3)	(4)	(5)	sehr schlecht
22. Mein Wissenszugewinn war ...	sehr hoch	(1)	(2)	(3)	(4)	(5)	sehr niedrig
23. Mein Arbeitsaufwand beim letzten Thema war ...	sehr gering	(1)	(2)	(3)	(4)	(5)	sehr hoch
24. Der Wiedererkennungswert zu bereits behandelten Themen war ...	sehr hoch	(1)	(2)	(3)	(4)	(5)	sehr niedrig
25. Die Nützlichkeit des Wissens bereits behandelte Themen für das letzte Thema war ...	sehr hoch	(1)	(2)	(3)	(4)	(5)	sehr niedrig

Abbildung 81: Fragen zum letzten Thema (Kognitive Dimension)

Die letzten Fragen beziehen sich auf eine Vernetzung zu den bisher unterrichteten Themen:

- Wie hoch war der Wiedererkennungswert zu anderen Themen und wie nützlich war das bisherige Wissen (siehe Fragen 24 und 25, Abbildung 81)?

Da die beiden fachdidaktischen Vorgehensweisen (OOP-First und OOP-Later) die Themen unterschiedlich sequenziell darbieten, kann dadurch herausgefunden werden, ob eine Vorgehensweise die Themen besser vernetzt als die andere Vorgehensweise (Ergebnisse siehe Kapitel 6.4.3 und Kapitel 7.4.5).

	Kognitive Dimension
Fragen zur Schule bzw. zum privaten Umfeld	5
Fragen zum Fach Informatik	11
Fragen zum jeweiligen Lehrer	17
Fragen zum letzten Thema	12,14,21,22, 23,24,25

Abbildung 82: Fragen zur kognitiven Dimension

Somit ist die Spalte „kognitive Dimension“ der Matrix von Abbildung 70 mit Fragen gefüllt worden (siehe Abbildung 82) und erfüllt die Vorgabe, dass der Schwerpunkt der Fragen in dem Bereich „letztes Thema“ liegt.

Jetzt fehlen nur noch Fragen zur dritten Erlebnis-Dimension, der motivationalen Dimension. Diese werden im nächsten Kapitel entwickelt.

### 5.2.3 Motivationale Fragestellungen

In vielen Untersuchungen wurde nachgewiesen, dass ein Zusammenhang besteht zwischen der Motivation bzw. den Motivationskomponenten und der nachfolgenden Lernleistung. Rheinberg, Vollmeyer und Burns [RVB01] haben hierfür extra einen Fragebogen zur Erfassung aktueller Motivation (FAM) entwickelt. Die dort vorgesehenen 18 Items sind allerdings für den hier zu entwickelnden Fragebogen zuviel: laut Vorgabe aus Kapitel 5.1.3 reicht bis auf den Frage-Bereich „letztes Thema“ ein Item bzw. eine Frage pro Frage-Bereich aus. Daher wird entweder direkt oder indirekt (mit Hilfe eines so genannten „Interesse-Items“) nach der Motivation gefragt.

Auch die motivationalen Fragestellungen werden wieder den vier verschiedenen Frage-Bereichen zugeordnet.

- Schule und Fach Informatik

In diesen Frage-Bereichen wird direkt nach der Motivation im Kontext zur Leistung gefragt, es wird dabei unterschieden zwischen der Motivation in der Schule gute Leistungen zu erbringen und der Motivation im Fach Informatik gute Leistungen zu erbringen (siehe Frage 4, Abbildung 83 und Frage 10, Abbildung 84).

4. Meine Motivation in der Schule gute Leistungen zu erbringen ist zur Zeit eher ...						
sehr hoch	(1)	(2)	(3)	(4)	(5)	sehr niedrig

Abbildung 83: Frage zur Schule und zum privaten Umfeld (Motivationale Dimension)

10. Meine Motivation im Fach Informatik gute Leistungen zu erbringen ist zur Zeit ...						
sehr hoch	(1)	(2)	(3)	(4)	(5)	sehr niedrig

Abbildung 84: Frage zum Fach Informatik (Motivationale Dimension)

- Lehrer

Fragen zum Lehrer gibt es nicht in dieser Erlebnis-Dimension. Die mögliche Fragestellung (26. „Meine Motivation bei meinem Informatik-Lehrer gute Leistungen zu erbringen ist zur Zeit ...“) wäre m. E. diskussions- bzw. fragwürdig. Es wird vielmehr davon ausgegangen, dass positive Emotionen zum Lehrer (vgl. Fragen 7, 18 und 19, Abbildung 75) die Motivation fördern.

- Thema

Wie kann ein Thema motivierend auf die Schüler wirken? Z. B. indem das Thema als interessant oder wichtig von den Schülern eingeordnet wird (siehe Fragen 13 und 16, Abbildung 85).

13. Unabhängig von der Schwere des Stoffes fand ich das letzte Thema ...						
sehr interessant	(1)	(2)	(3)	(4)	(5)	sehr langweilig
16. Das letzte Thema ist meiner Meinung nach für die Informatik ...						
sehr wichtig	(1)	(2)	(3)	(4)	(5)	sehr unwichtig

Abbildung 85: Fragen zum letzten Thema (Motivationale Dimension)

Damit sind auch die Fragen für die letzte Dimension der drei Erlebnis-Dimensionen (siehe Matrix in Abbildung 70) formuliert worden und können in die Spalte „motivationale Dimension“ eingetragen werden (siehe Abbildung 86).

	Motivationale Dimension
Fragen zur Schule bzw. zum privaten Umfeld	4
Fragen zum Fach Informatik	10
Fragen zum jeweiligen Lehrer	
Fragen zum letzten Thema	13,16

Abbildung 86: Fragen zur motivationalen Dimension

Die fehlende Frage zum Lehrer wird dabei bewusst in Kauf genommen (siehe oben).

Nunmehr kann die vollständige Matrix dargestellt und analysiert werden.

## 5.2.4 Kategorisierungs-Matrix

Es wurden 25 Fragen generiert und somit die Vorgabe aus dem Kapitel 5.1.3 (mit ca. 24 Fragen) erfüllt. Wenn diese in die Matrix von Abbildung 70 aus Kapitel 5.1.3 eingetragen werden, ergibt sich eine fast vollständige Füllung der Matrix (siehe Abbildung 87).

	Emotionale Dimension	Kognitive Dimension	Motivationale Dimension
Fragen zur Schule bzw. zum privaten Umfeld	1,2,3	5	4
Fragen zum Fach Informatik	6,8,9	11	10
Fragen zum jeweiligen Lehrer	7,18,19	17	
Fragen zum letzten Thema	15,20	12,14,21,22, 23,24,25	13,16

Abbildung 87: Fragen im Kontext zu den Erlebnis-Dimensionen und den Frage-Bereichen  
(vgl. [ES10a, Figure 3, S. 109])

An der Matrix erkennt man, dass teilweise verschiedene Aussagen (Items) zu einer Skala zusammengefasst werden. In der Regel ist es so, dass die einzelnen Skalen umso genauer sind, je mehr Items dahinter stecken.



Man sieht an der Matrix mit den zwölf Skalen deutlich, dass der „Genauigkeits-Schwerpunkt“, also der Schwerpunkt der Fragen, auf der emotionalen Dimension bzw. auf den Fragen zum letzten Thema liegt [ES10a]. Dies war auch eine Vorgabe aus dem Kapitel 5.1.3:

„For the three dimensions and the four locations scales with several items were combined in a questionnaire with 25 items. Each item is answerable on a five-point-rating scale. [...] As can be seen, focus is on the emotional dimension with regard to all locations, and on the teaching topic with regard to all dimensions.” [ES10a, S. 109].

Somit kann nun der Fragebogen erstellt werden (siehe Anhang K).

### **5.3 Zusammenfassung**

In diesem Kapitel wurde ein Fragebogen für das subjektive Erleben des Unterrichts entwickelt, mit dem die Erlebnisdaten der Schülerinnen und Schüler erhoben werden können. Dazu wurde das subjektive Erleben in drei Dimensionen (siehe Kapitel 5.1.1) und vier Bereiche (siehe Kapitel 5.1.2) aufgeschlüsselt, so dass letztlich eine 3x4-Frage-Matrix entstand (siehe Kapitel 5.1.3). Danach wurden Fragen für die emotionale Dimension (siehe Kapitel 5.2.1), die kognitive Dimension (siehe Kapitel 5.2.2) und die motivationale Dimension (siehe Kapitel 5.2.3) entwickelt. Der Schwerpunkt der Fragestellungen (siehe Kapitel 5.2.4) liegt einerseits in der emotionalen Dimension, andererseits auf dem letzten Unterrichtsthema. Im Fragebogen selber (siehe Anhang K) sind die Fragen aber nach den vier Bereichen „die Schule bzw. das private Umfeld“, „das Fach Informatik“, „der Lehrer“ und „das letzte Thema“ geordnet, die Schülerinnen und Schüler bekommen daher die Aufteilung der Fragen in die drei Erlebnis-Dimensionen nicht mit.



## 6 Ergebnisse und Auswertung der empirischen Studie

In den folgenden Unterkapiteln sollen die Ergebnisse der empirischen Studie ausgewertet werden. Hierfür werden zuerst die statistischen Grundlagen erläutert (siehe Kapitel 6.1).

Danach wird der Fokus kurz auf die Ergebnisse der Vorstudie im Hinblick auf den objektiven Aspekt (siehe Kapitel 6.2.1) und den subjektiven Aspekt (siehe Kapitel 6.2.2) gelegt. Diese Ergebnisse werden jeweils ausgewertet und interpretiert.

Der Hauptfokus liegt aber auf den Ergebnissen der Hauptstudie. Hier widmen sich zwei große Unterkapitel der Beantwortung der beiden Forschungsfragen, zum einen nach dem Lernerfolg nach einem Schuljahr (siehe Kapitel 6.3) und zum anderen nach dem subjektiven Erleben des Unterrichts (siehe Kapitel 6.4).

Zur Analyse des Lernerfolgs nach einem Schuljahr werden die Ergebnisse des Vortests (siehe Kapitel 6.3.1), des Vergleichstests (siehe Kapitel 6.3.2) und des Nachhaltigkeitstests (siehe Kapitel 6.3.3) erläutert, um danach den Verlauf des Lernerfolgs herauszuarbeiten (siehe Kapitel 6.3.4). Am Ende werden noch ausgewählte Lerngruppen untersucht: die leistungsstärksten bzw. leistungsschwächsten Schüler und die Schülerinnen (siehe Kapitel 6.3.5).

Für das subjektive Erleben des Unterrichts werden die Ergebnisse der Befragung der Schülerinnen und Schüler im Hinblick auf die emotionale Dimension (siehe Kapitel 6.4.2), die kognitive Dimension (siehe Kapitel 6.4.3) und die motivationale Dimension (siehe Kapitel 6.4.4) dargestellt bzw. erläutert. Alle drei Erlebnis-Dimensionen werden dann zusammengefasst zum Faktor „subjektives Erleben“ (siehe Kapitel 6.4.5). Am Ende werden auch hier noch ausgewählte Lerngruppen untersucht: die leistungsstärksten bzw. leistungsschwächsten Schüler und die Schülerinnen (siehe Kapitel 6.4.6).

### 6.1 Statistische Grundlagen

Wenn es um den Vergleich zweier Stichproben hinsichtlich der Mittelwerte geht, wird in der Regel der so genannte Student'sche t-Test angewendet. Bei ihm wird wie folgt vorgegangen [Els04, S. 112-114]:

- Die Nullhypothese ( $H_0: \mu_1 = \mu_2$ ) und die Alternativhypothese (z. B.  $H_1: \mu_1 > \mu_2$ ) werden definiert.
- Die Signifikanzzahl bzw. das Signifikanzniveau  $\alpha$  (Irrtumswahrscheinlichkeit) wird gewählt, z. B. 5% oder 1%.
- Die kritischen Werte werden aus einer vorgegebener Tabelle ermitteln, z. B.  $t_{\text{krit}} = t_{0,95;18}$  (wobei die letzte Zahl der Freiheitsgrad der t-Verteilung ist).
- Aus dem Stichprobenwert wird die Prüfgröße  $t_{\text{st}}$  berechnet.
- Wenn  $t_{\text{st}} > t_{\text{krit}}$  ist, wird die Nullhypothese verworfen und die Alternativhypothese angenommen.

Im Kapitel 2.4.6 wurden die Nullhypothesen und die Alternativhypothesen formuliert. Um möglichst objektiv die beiden fachdidaktischen Vorgehensweisen vergleichen zu können, wurden letztlich die Alternativhypothesen ungerichtet formuliert, so dass sie durch einen zweiseitigen t-Test überprüft werden müssen. Sollte bei dieser Signifikanzprüfung („Sind die (Mittelwerts-)Unterschiede relevant?“) das Signifikanzniveau erreicht werden, kann zusätzlich die Effektstärke quantitativ geprüft werden („Wie groß sind die Unterschiede?“).

Darf überhaupt der t-Test angewendet werden? Detten, Faude und Meyer [DFM08] geben in ihrem „Leitfaden zur statistischen Auswertung von empirischen Studien“ ein Raster an, mit dem das benötigte statistische Verfahren in Abhängigkeit von der Art des Versuchs und der verwendeten Messskala abgelesen werden kann (siehe Abbildung 88). Da es sich in dieser Studie um zwei unverbundene Stichproben handelt und in Kapitel 5.1.4 begründet wurde, dass nicht nur bei den Testergebnissen, sondern auch bei den Fragen zum subjektiven Erleben eine Intervallskala vorliegt, ist daher der t-Test als statistisches Verfahren richtig gewählt.

Messskala	Art des Versuchs				
	2 (unverbundene) Stichproben mit verschiedenen Individuen	3 oder mehr Stichproben mit verschiedenen Individuen	2 verbundene Stichproben (vor und nach einer Therapie am selben Individuum)	Mehr als 2 verbundene Stichproben (mehrere Therapien an denselben Individuen)	Zusammenhang zweier Variablen
Intervall (Daten stammen aus normalverteilten Grundgesamtheiten)	t-Test für unverbundene Stichproben (4.1)	Varianzanalyse (4)	t-Test für verbundene Stichproben (5.1)	Varianzanalyse für mehrfache Messungen (5.2)	Lineare Regression, Pearson-sche Produktmomentkorrelation oder Analyse nach Bland-Altman (10)
Ordinal	Mann-Whitney-Rangtest (8.1)	Kruskal-Wallis-Test (8.2)	Wilcoxon-Rang-Test (8.3)	Friedmann-Test (8.4)	Spearman'sche Rangkorrelation (10.3)
Nominal	Chi-Quadrat-Test für Kontingenztafeln (9.3)	Chi-Quadrat-Test für Kontingenztafeln (9.3)	McNemar-Test (9.4)	Q-Test nach Cochran	Kontingenzkoeffizient

Abbildung 88: Statistische Verfahren zum Testen von Hypothesen [DFM08, S. 9, Tabelle 2]

Mit dem t-Test können die Mittelwerte zweier Stichproben verglichen werden. Vorher muss aber mit dem F-Test die Frage beantwortet werden, ob die Varianzen in den zwei Gruppen verschieden sind.

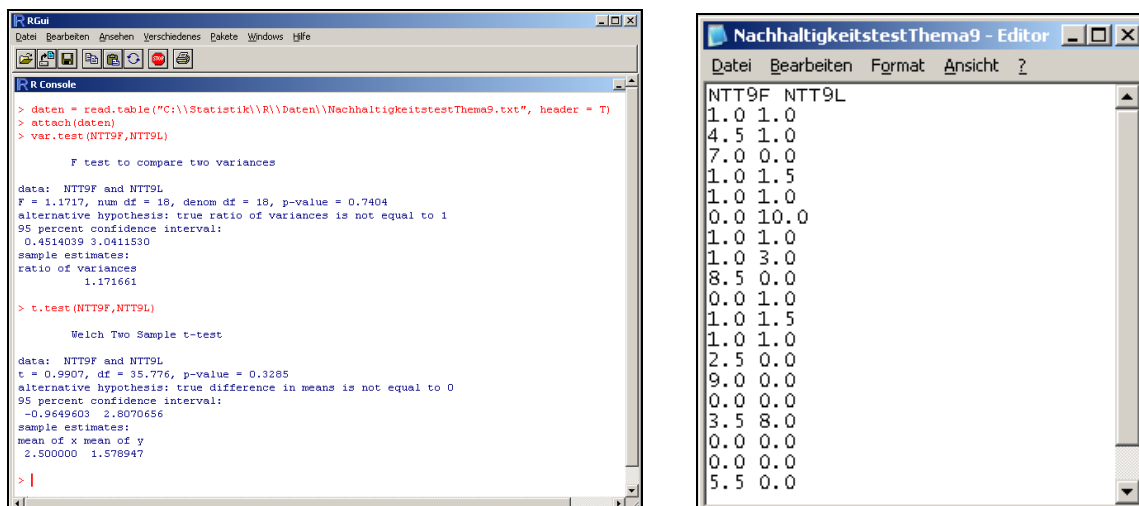


Abbildung 89: Anwendung des t-Tests (Welch Two Sample t-Test) mit Hilfe des Statistik-Tools R

(vgl. p-Wert für das Thema 9 in Abbildung 107)

Dies geschieht bei dem R-Tool mit (siehe auch Abbildung 89):

```
var.test(Daten1, Daten2)
```

Sind die Varianzen verschieden, dann wird beim t-Test die so genannte „Welch-Korrektur“ für unterschiedliche

Varianzen in den beiden Stichproben angewendet:

```
t.test(Daten1, Daten2, var.equal=FALSE)
```

Der letzte Parameter kann aber auch weggelassen werden, da das R-Tool die Welch-Korrektur als Voreinstellung für unterschiedliche Varianzen anwendet (siehe Abbildung 89).

Das Ziel der Datenbeschreibung und Verallgemeinerung in dieser Studie ist durch die Angabe des p-Werts mit Hilfe des statistischen t-Tests eine Nullhypothese mit vorgegebener Unsicherheit abzulehnen. Das Ziel ist nicht durch die Angabe des Konfidenzintervalls den Mittelwert eines Parameters mit vorgegebener Ungenauigkeit zu schätzen [SH09, S. 273]. Das Konfidenzintervall ist dabei das Intervall, das den wahren aber unbekanntem Parameter mit einer vorgegebenen Vertrauenswahrscheinlichkeit überdeckt [SH09, S. 291]. Statt der Angabe des Vertrauensbereiches wird in der Studie der p-Wert mit Hilfe der Angabe der Effektstärke abgesichert (s. u.).

Bei der Festlegung des Signifikanzniveaus hält sich die Studie an übliche Konventionen (vgl. z. B. [BL03, S. 52] und [SH09, S. 361]). Gewählt wird:

$$\alpha = 0,05$$

Der p-Wert eines Signifikanztests gibt an, mit welcher Wahrscheinlichkeit in der kleinen Stichprobe  $H_1$  gilt, obwohl in der Wirklichkeit  $H_0$  gilt. Der p-Wert ist die Überschreitungswahrscheinlichkeit, mit der man sich irrt, wenn man die Nullhypothese ablehnt [SH09, S. 378-379]. Bei dem vorgegebenen Signifikanzniveau sollte der p-Wert kleiner als 5% sein, d. h. es liegt eine Irrtumswahrscheinlichkeit von kleiner oder gleich 5% vor:

$$p \leq 0,05$$

Der so genannte  $\alpha$ -Fehler gibt an, mit welcher Wahrscheinlichkeit eine an sich richtige Nullhypothese  $H_0$  aufgrund eines Stichprobenergebnisses verworfen wird. Der so genannte  $\beta$ -Fehler gibt dagegen an, mit welcher Wahrscheinlichkeit eine an sich falsche Nullhypothese  $H_0$  beibehalten wird, obwohl die Alternativhypothese  $H_1$  richtig ist. Die so genannte Effektgröße  $\Delta$  gibt dabei das Ausmaß der Abweichung an:

$$H_1: \mu_A \geq \mu_B + \Delta$$

Da der Stichprobenumfang ( $N = 19$  bzw.  $N = 22$ , abhängig vom Anwendungsfall) durch die Anzahl der Schüler in einer Klasse vorgegeben war, kann nun nachträglich die Effektstärke bzw. Effektgröße angegeben werden. Die Effektstärke  $d$  ergibt sich aus der Differenz der Mittelwerte dividiert durch die Streuung der Gesamtgruppe:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{(s_1^2 + s_2^2)/2}}$$

Dabei werden laut Cohen [Coh88, S. 20-27] Werte von 0,2 bis 0,5 als schwache Effekte („small effect size“), von 0,5 bis 0,8 als mittelstarke Effekte („medium effect size“) und Werte über 0,8 als starke Effekte („large effect size“) gewertet. Die Effektstärke  $d$  kann dabei auch den Wert 1 überschreiten.

Da bei den beiden Stichproben im subjektiven Fragebogen insgesamt 25 Items erhoben werden, besteht zusätzlich die Möglichkeit festzustellen, ob einzelne Items zusammenhängen und ob diese Korrelationen (mit Hilfe des so genannten Korrelationskoeffizienten  $r$ ) statistisch signifikant sind [BL03, Kapitel 5]. Laut Schlittgen [Sch03a, S. 179] werden dabei Werte bis 0,5 als schwache, von 0,5 bis 0,8 als mittlere und Werte über 0,8 als starke Korrelation interpretiert.

Die statistischen Berechnungen und die Darstellungen in den folgenden Kapiteln werden mit Hilfe des Open-Source-Statistik-Tools *R* (Version 2.7.1) und der Anwendungssoftware *Excel* (Version 2000) von Microsoft vorgenommen bzw. erzeugt.

## 6.2 Vorstudie: Ergebnisse und Interpretationen

Im Folgenden sollen die Resultate der Vorstudie aus dem Schuljahr 2006/2007 nur kurz und in Ausschnitten vorgestellt werden, da die Vorstudie hauptsächlich das Studiendesign überprüfen sollte. Dass die Vorstudie tatsächlich einige Design-Schwächen offenbart hat, wurde im Kapitel 3.3.14 schon dargestellt. Dort wurde auch erläutert, inwiefern die Lehren der Vorstudie für die Hauptstudie genutzt werden. Da die erhobenen Daten in Bezug zu den Schwächen der Vorstudie gesehen werden müssen, werden die Ergebnisse nur punktuell und mit der gebotenen Vorsicht ausgewertet. Auf Signifikanzberechnungen wird daher verzichtet (Ausnahme: Vergleichstest).

### 6.2.1 Objektiver Aspekt

Dieses Kapitel ist aus der Veröffentlichung zu der INFOS-Konferenz 2009 [ES09c] übernommen, teilweise wörtlich und teilweise überarbeitet.

Bei der Vorstudie im Schuljahr 2006/2007 gab es nur einen Test, den Vergleichstest (Post-Test). Dieser gliederte sich in sechs Themengebiete (vgl. Kapitel 4.3.1). Die OOP-Later-Klasse war in der Summe der Ergebnisse um 5% besser (siehe Abbildung 90) als die OOP-First-Klasse. Geht man von dem im Jahr 2007 gültigen Notenschlüssel für die Einführungsphase des Beruflichen Gymnasiums aus (siehe Abbildung 153), dann hat die OOP-First-Klasse durchschnittlich die Note 3 erreicht, die OOP-Later-Klasse durchschnittlich die Note 3+.

Thema	Punkte	OOP-First	OOP-Later
Daten- und Steuerstrukturen	20	61%	59%
Methoden	20	67%	64%
Struktogramme	20	60%	64%
„Statische OOP“	20	72%	79%
UML-Klassendiagramme und -Objektdiagramme	20	<b>72%*</b>	<b>85%*</b>
Zusatz: „Dynamische OOP“	10	<b>50%*</b>	<b>67%*</b>
<b>Gesamt bzw. Durchschnitt</b>	<b>110</b>	<b>65%</b>	<b>70%</b>

Abbildung 90: Ergebnisse des Vergleichstests in der Vorstudie 2006/2007

([ES09a, Fig. 5, S. 19] [ES09c, Abb. 6, S. 128]; \* signifikante Ergebnis-Unterschiede sind fett gedruckt; siehe Daten in Anhang G)

Die Lernergebnisse (siehe Anhang G) waren bei vier von sechs Themen bei der OOP-Later-Klasse besser als bei der OOP-First-Klasse. Zwei von drei OOP-Themen (nicht-schraffierte Balken in Abbildung 91) waren sogar signifikant besser (siehe fettgedruckte Zahlen in Abbildung 90):

- UML-Klassendiagramme und UML-Objektdiagramme

- „Dynamische OOP“

Dieses Ergebnis ist in zweierlei Hinsicht überraschend. Zum einen, dass gerade einige OOP-Themen von der OOP-Later-Klasse besser gelernt worden sind (obwohl erst später im Unterricht fokussiert), zum anderen, dass gerade das OOP-Thema „UML“, welches ja einen starken Modellierungsaspekt beinhaltet, von der OOP-First-Klasse schlechter gelernt worden ist (obwohl von Anfang an im Unterricht fokussiert).

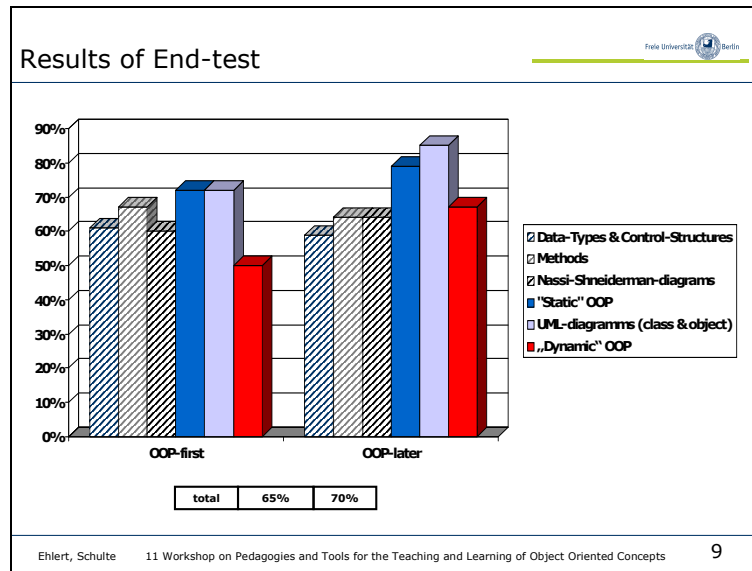


Abbildung 91: Darstellung der Ergebnisse auf der ECOOP-Konferenz 2007 [ES07b, Folie 9]

Im Folgenden werden drei Interpretationsansätze für dieses Ergebnis aufgezeigt:

1. Der OOP-Later-Weg bringt durch seine spezielle Reihenfolge der Unterrichts-Themen signifikante Vorteile bei einigen OOP-Themen.

Begründungsversuch: Die Themen werden mit aufsteigender Schwierigkeit hintereinander angeboten, es wird immer wieder weiterer Stoff hinzugefügt und damit ein systematischer Wissensaufbau erreicht (vgl. Kapitel 1.2.4). Zusätzlich werden die Schüler mit einem relativ großen Vorwissen (über die prozedurale bzw. imperative Programmierung) in das OO-Paradigma eingeführt und lernen es dadurch besser. Dies wäre im Sinne der ersten Thesen (siehe Kapitel 2.4.5), die Vorteile bei einem OOP-Later-Vorgehen sehen.

2. Die Schülerinnen und Schüler können das zuletzt Gelernte am besten abrufen.

Begründungsversuch: Oft ist es beim Lernen so, dass die zuletzt gelernten Themen am besten abgerufen werden können. Dies waren bei der OOP-Later-Klasse ja die OOP-Themen [ES09a, S. 19].

3. Das Unterrichtsgeschehen in beiden Klassen war zu unterschiedlich, also mangelhaft synchronisiert.

In Kapitel 3.3.14 wurde schon ausführlich die mangelhafte Synchronisation des Unterrichts in der Vorstudie dargestellt bzw. bemängelt, z. B. dass sich die unterrichteten Module unterschieden:

„Overall, the modules have been taught as planned, although in detail there are differences. For example, in the object first group the modules 11 and 12 were taught together. Another issue is that within modules some details

differ, and maybe even the level of depth of the topics taught. Based on these experiences we plan to repeat the study next school year.” [ES07a, S. 4].

Auffällig ist, dass die OOP-First-Klasse mit den Themen (damals noch Module genannt) schneller vorangekommen ist, so dass ein kurzes Projekt das Schuljahr abschloss. Dies kann ein Vorteil des OOP-First-Weges sein, evtl. aber auch die Erklärung dafür sein, warum die Schüler schlechter abgeschnitten haben. Vielleicht „verführt“ der OOP-First-Einstieg zu einem schnelleren Vorgehen, so dass für die einzelnen Themen weniger Zeit verwandt wird und sie somit weniger detailliert dargeboten werden:

„However, during the year of the pre-study, the OOP-first class progressed much faster so there was time for additional topics at the end of the school year. This was not the case for the OOP-later group which only learned the OOP-topics at the end. Consequently, the results may be ascribed to the fact that OOP-first was not taught at the same level of detail (but covered more topics), and the OOP-later students were better in those topics they learned last.

The overall interpretation of this result is difficult. One conclusion was to check in the main study more precisely that both classes learn the same topics at the same length. In addition, these differences might be vanishing some time later on. One interpretation though is the feeling that OOP-first approaches might ‘seduce’ the teacher to proceed (too) quickly.“ [ES09a, S. 19].

Um mehr Klarheit zu gewinnen, wird über eine Reihe von Maßnahmen die Synchronisation des Unterrichts in der Hauptstudie entscheidend verbessert (siehe Kapitel 3.3.14). Dann kann überprüft werden, ob sich die Ergebnisse immer noch unterscheiden (also reproduzierbar sind) oder nicht (siehe Kapitel 6.3.2 und Kapitel 6.3.3). Unterscheiden sich die Ergebnisse weiterhin, dann erzielt ein OOP-Later-Vorgehen im allgemeinen anscheinend bessere Lernergebnisse. Werden andere Ergebnisse erzielt (z. B. eine Reduzierung oder eine Aufhebung der Unterschiede), dann trifft folgende (eingeschränkte) Aussage zu: „Das spezielle OOP-Later-Vorgehen der Vorstudie erzielte bessere Lernerfolge als das spezielle OOP-First-Vorgehen der Vorstudie!“.

Es wird so oder so deutlich, dass unterschiedlicher Unterricht unterschiedlichen Lernerfolg bewirken kann.

## 6.2.2 Subjektiver Aspekt

Dieses Kapitel orientiert sich überwiegend an der Veröffentlichung [ES07a] zu einem Workshop („Eleventh Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts“) auf der ECOOP-Konferenz 2007.

Der Fragebogen der Vorstudie (siehe Anhang D) bestand nur aus zehn Fragen, davon bezogen sich acht Fragen auf das subjektive Erleben des Unterrichts bzw. des letzten Moduls (Nummerierung laut Fragebogen):

2. Das **Fach Informatik** (innerhalb der Technik) gefällt mir ...



3. Der **unterrichtende Lehrer** gefällt mir ...
4. Meine Motivation, die Themen des **Faches Informatik** zu verstehen (evtl. auch durch verstärktes Lernen zu Hause) ist ...
5. Ich fühle mich allgemein beim Unterricht im **Fach Informatik** eher ...
6. Der Stoff des **letzten Moduls** hat zu meinem Gesamtverständnis (der Themen) des Faches Informatik ...
7. Den Stoff des **letzten Moduls** empfand ich als ...
8. Unabhängig von der Schwere des Stoffes fand ich das Thema des **letzten Moduls** ...
9. Ich fühlte mich speziell beim **letzten Modul** eher ...

Dieser Fragebogen wurde nach jedem der unterrichteten Module von den Schülerinnen und Schülern ausgefüllt. Wie in Kapitel 3.3.14 ausführlich dargestellt, wurden nicht alle Module gleich unterrichtet bzw. abgefragt, so dass für die Datenauswertung im Hinblick auf einen Vergleich der beiden Klassen nur die sieben Module aus Abbildung 92 zur Verfügung stehen (Hinweis: Die Modul-Nummerierung erfolgte in der Vorstudie willkürlich).

<b>Modul-Nr.</b>	<b>Modul-Thema</b>	<b>Später: Thema-Nr.</b>
1	Einführung in die OOP mit BlueJ	T1
10	Einführung in die IDE, Variablen und Datentypen	T2
11/12	Selektionen (und Sequenz)	T5 (ohne Sequenz)
2	Setter, Getter, Konstruktoren, Fachkonzept, Nutzerebene	T3
20	Schleifen	T6
3	Methoden	T4
7	Vererbung	T8

Abbildung 92: Auswertbare Fragebögen im Hinblick auf die Modul-Themen (vgl. Kapitel 3.3.14)

Im weiteren Abschnitt werden die Daten (siehe Anhang E) punktuell ausgewertet. Die Antworten zu den Fragen 2, 4 und 5 sind auf das Fach Informatik bezogen und die Antworten zu den Fragen 6, 7, 8 und 9 sind auf die einzelnen Modul-Themen bezogen. Bei den Antworten zur Frage 3 (Frage zum Lehrer) wird nur der Durchschnittswert erhoben (um diesen später mit dem Wert der Hauptstudie zu vergleichen), die Antworten zu Modul 11 und Modul 12 wurden bei der OOP-Later-Klasse gemittelt, da in der OOP-First-Klasse beide Module gleichzeitig abgefragt wurden. Es gibt keine Untersuchung im Hinblick auf statistische Signifikanz wegen der mangelhaften Synchronisation des Unterrichts (vgl. Kapitel 3.3.11 und Kapitel 3.3.14).

Die folgenden Darstellungen (Abbildung 93 bis Abbildung 100) sind auf die Themen bezogen, sie spiegeln also keine zeitliche Reihenfolge wider. Erst die Abbildung 101 und die Abbildung 102 beziehen sich auf den zeitlichen Verlauf.

#### a) Das Fach Informatik

Das Fach Informatik gefällt den Schülern unabhängig von der fachdidaktischen Vorgehensweise (siehe Abbildung 93).

Beide Klassen haben am Anfang (OOP-First: „BlueJ“ / OOP-Later: „IDE und Datentypen“) bei der ersten Umfrage die gleiche gute Sichtweise (Wert: 1,8) auf das Fach Informatik. Bei der letzten Umfrage (nach dem Thema „Vererbung“) sind die Werte etwas schlechter und unterscheiden sich in den beiden Klassen (OOP-First: 2,3 / OOP-Later: 2,0).

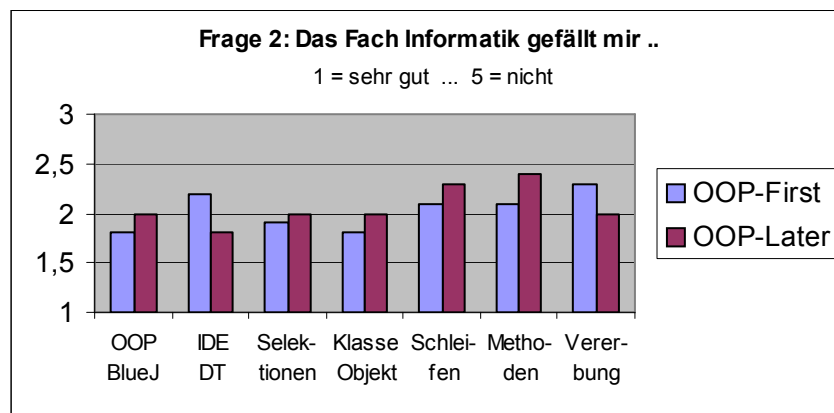


Abbildung 93: Ergebnisse zur Frage 2 (OOP-First:  $\bar{x}$  = 2,0 / OOP-Later:  $\bar{x}$  = 2,1)

Die Durchschnittswerte der Antworten zur Frage 4 (also zur Motivation) liegen höher (also schlechter) als die zur Frage 2, sie liegen aber immer noch im positiven Bereich (unter dem Wert 3, siehe Abbildung 94).

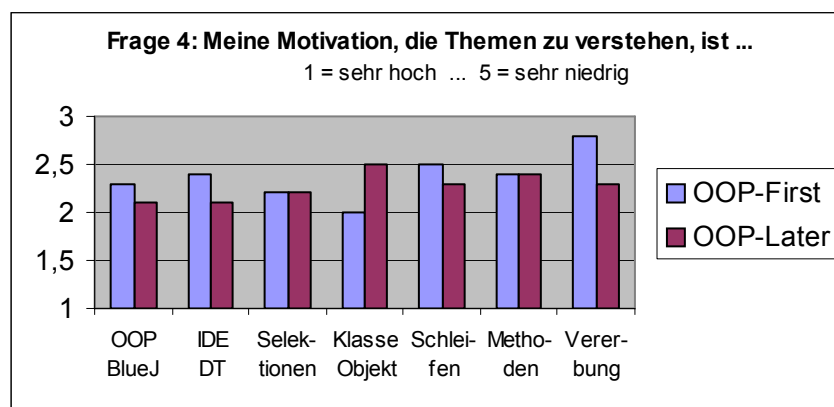


Abbildung 94: Ergebnisse zur Frage 4 (OOP-First:  $\bar{x}$  = 2,4 / OOP-Later:  $\bar{x}$  = 2,3)

Die Frage nach der Motivation impliziert auch, ob die Schülerinnen und Schüler bereit sind, sich den Stoff durch Arbeit (z. B. durch verstärktes Lernen zu Hause) besser zu erschließen.

Die OOP-Later-Schüler sind zwar etwas (Differenz-Wert: 0,1) motivierter, dafür fühlen sich die OOP-First-Schüler etwas (Differenz-Wert: 0,1) wohler (Frage 5, siehe Abbildung 95).

Der so genannte „Wohlfühlfaktor“ beider Klassen (bezogen auf das Fach Informatik) ist im positiven Bereich, also unter dem Durchschnittswert 3 auf der Skala von 1 bis 5. Ähnlich wie bei Frage 2 fühlen sich beide Klassen am Anfang (OOP-First: „BlueJ“ / OOP-Later: „IDE und Datentypen“) bei der ersten Umfrage wohl (Wert: 2,2). Bei der letzten Umfrage (nach dem Thema „Vererbung“) haben sich die Werte verschlechtert (OOP-First: 2,7 / OOP-Later: 2,6).

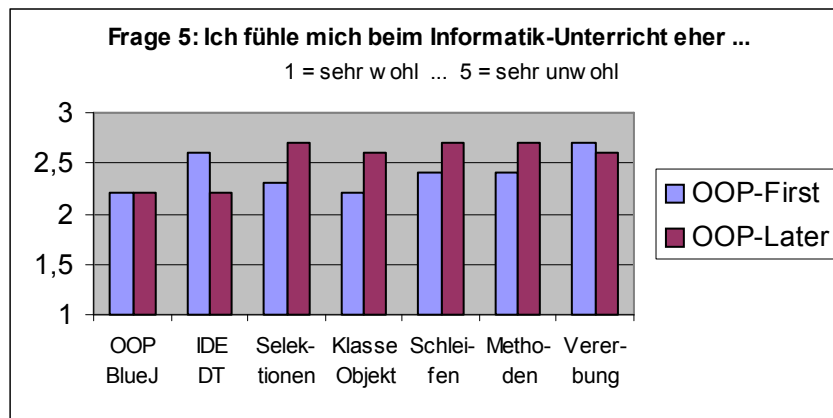


Abbildung 95: Ergebnisse zur Frage 5 (OOP-First:  $\bar{x} = 2,4$  / OOP-Later:  $\bar{x} = 2,5$ )

b) Die Lehrer

Die beiden Lehrer werden von den beiden Klassen unterschiedlich wahrgenommen bzw. beurteilt (siehe Abbildung 96).

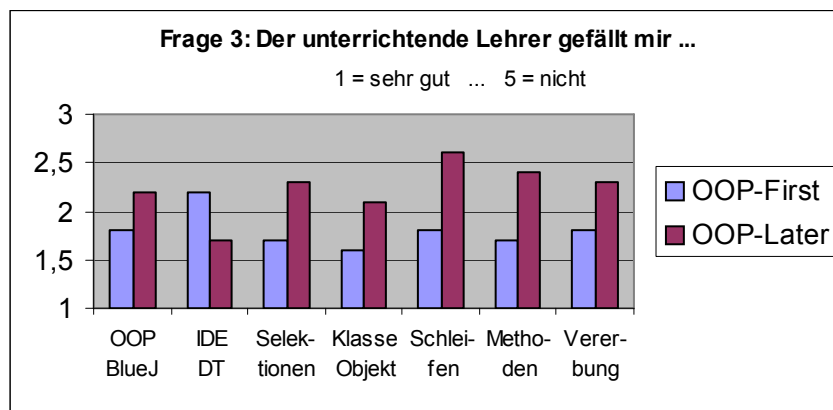


Abbildung 96: Ergebnisse zur Frage 3 (OOP-First:  $\bar{x} = 1,8$  / OOP-Later:  $\bar{x} = 2,2$ )

Der Durchschnittswert-Unterschied (OOP-First:  $\bar{x} = 1,8$  / OOP-Later:  $\bar{x} = 2,2$ ) beträgt immerhin 0,4 Punkte, beide Werte sind aber (sehr) positiv. Diese Werte werden später mit denen der Hauptstudie verglichen (siehe Kapitel 7.2.1).

c) Die Modul-Themen

Vier Fragen beziehen sich auf das letzte Modul-Thema. Neben dem öfters schon erwähnten Schwierigkeitsfaktor (Frage 7) und dem Wohlfühlfaktor (Frage 9) gibt es noch einen „Interessanzfaktor“ (Frage 8) und einen „Gesamtverständnisfaktor“ (Frage 6).

Für die OOP-First-Schüler war der Stoff der Themen „BlueJ“ und „Klasse und Objekt“ der Stoff, der am meisten zu ihrem Gesamtverständnis (der Themen) des Faches Informatik beigetragen hat (Frage 6, siehe Abbildung 97).

Bei den OOP-Later-Schülern gilt dies aber nur für das Thema „BlueJ“.

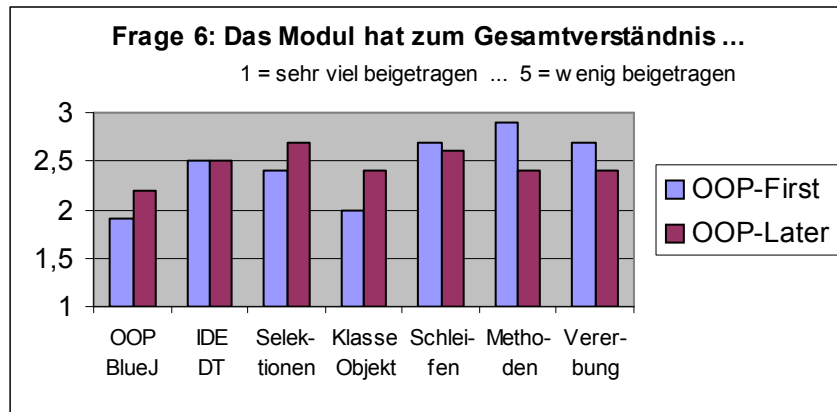


Abbildung 97: Ergebnisse zur Frage 6 (OOP-First:  $\bar{x} = 2,4$  / OOP-Later:  $\bar{x} = 2,5$ )

Das subjektive „Schwere-Empfinden“ der Schüler bezogen auf die einzelnen Modul-Themen ist teilweise sehr unterschiedlich (Frage 7, siehe Abbildung 98) und wird weiter unten diskutiert (in Bezug zur sequenziellen Abfolge).

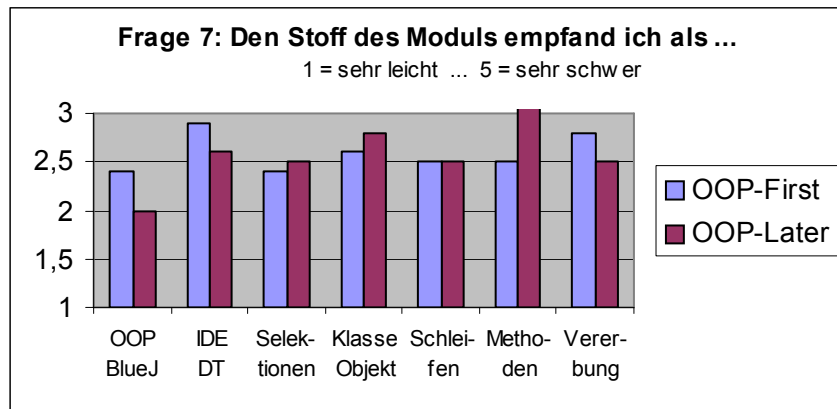


Abbildung 98: Ergebnisse zur Frage 7 (OOP-First:  $\bar{x} = 2,6$  / OOP-Later:  $\bar{x} = 2,6$ )

Die Themen werden von beiden Klassen als interessant angesehen (Frage 8, siehe Abbildung 99), der Durchschnittswert der Frage 8 ist deutlich besser (um den Faktor 0,4) als der zur Frage 7.

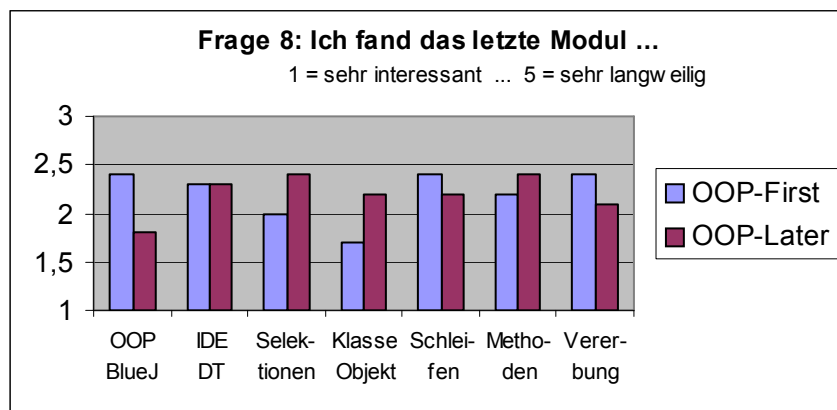


Abbildung 99: Ergebnisse zur Frage 8 (OOP-First:  $\bar{x} = 2,2$  / OOP-Later:  $\bar{x} = 2,2$ )

Das Thema „Klasse und Objekt“ ist für die OOP-First-Schüler das interessanteste Thema, die Themen „BlueJ“, „Schleifen“ und „Vererbung“ landen auf den letzten Plätzen. Für die OOP-Later-Schüler ist das Thema „BlueJ“ am interessantesten.

Der Wohlfühlfaktor ist bei beiden Klassen im Mittel ungefähr gleich (Frage 9, siehe Abbildung 100).

Große Unterschiede im Wohlfühlfaktor gibt es aber bei dem Thema „Klasse und Objekt“ und dem Thema „Methoden“, hier fühlen sich die OOP-Later-Schüler deutlich schlechter. Die Schüler der OOP-First-Klasse fühlen sich bei dem Thema „Klasse und Objekt“ und dem Thema „BlueJ“ am wohlsten, die Schüler der OOP-Later-Klasse beim Thema „BlueJ“.

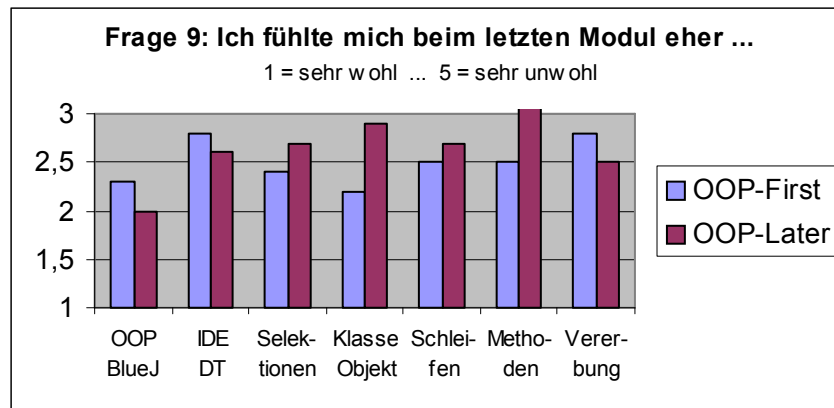


Abbildung 100: Ergebnisse zur Frage 9 (OOP-First:  $\bar{x} = 2,5$  / OOP-Later:  $\bar{x} = 2,6$ )

Es sollen im Folgenden einzelne Aspekte der Antworten der Schülerinnen und Schüler der beiden Klassen herausgegriffen bzw. interpretiert werden:

- Das Schwierigkeits-Empfinden in Bezug auf die zeitliche Abfolge der Unterrichtsthemen
- Das Schwierigkeits-Empfinden in Bezug zu den Ergebnissen des Vergleichstests
- Erklärungsversuch für die Unterschiede im Schwere-Empfinden beim Thema „Methoden“
- Erklärungsversuche für die Unterschiede im Schwere-Empfinden beim Thema „IDE (Texteditor) und Datentypen“
- Erklärungsversuch für die Unterschiede in der Rangfolge der Themen im Hinblick auf die Wichtigkeit der Themen für das Gesamtverständnis
- Fachdidaktische Reaktionen

c1) Das Schwierigkeits-Empfinden in Bezug auf die zeitliche Abfolge der Unterrichtsthemen

Die nächsten beiden Abbildungen zeigen die Abweichung der „empfundenen Schwere“ der einzelnen Themen bezogen auf den Durchschnittswert. Positive Werte bedeuten, dass die Themen schwerer als im Durchschnitt empfunden werden [ES07a, S. 5]. Die Null-Linie entspricht in beiden Abbildungen dem Mittelwert 2,6.

Für die OOP-First-Klasse können folgende Aussagen (siehe Abbildung 101) gemacht werden [ES07a, S. 5]:

- Der „BlueJ-Einstieg“ wird als leicht empfunden.
- Der Wechsel vom ersten Modul (BlueJ: 2,4) auf das zweite Modul (IDE: 2,9) scheint den Schülern schwerzufallen, die Differenz beträgt 0,5.
- Die Module „Steuerstruktur Selektion“ und „Steuerstruktur Iteration“ werden als eher leicht empfunden.
- Außer dem „Einstieg in den Texteditor“ wird nur noch das Thema „Vererbung“ als deutlich schwerer als der Durchschnitt empfunden.

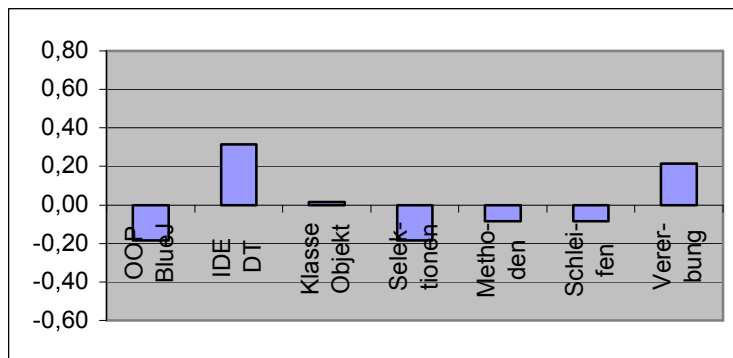


Abbildung 101: Schwere-Empfinden der OOP-First-Klasse (Null-Linie:  $\emptyset = 2,6$ , vgl. [ES07a, Figure 8])

Rein optisch betrachtet kann man von einer „ruhigen Wellenbewegung“ sprechen, der Extremwert (bezogen auf den Mittelwert) beträgt nur 0,3. Oder seriöser formuliert (siehe Box-Whisker-Plot in Abbildung 103): Alle Messwerte befinden sich innerhalb der Spannweite.

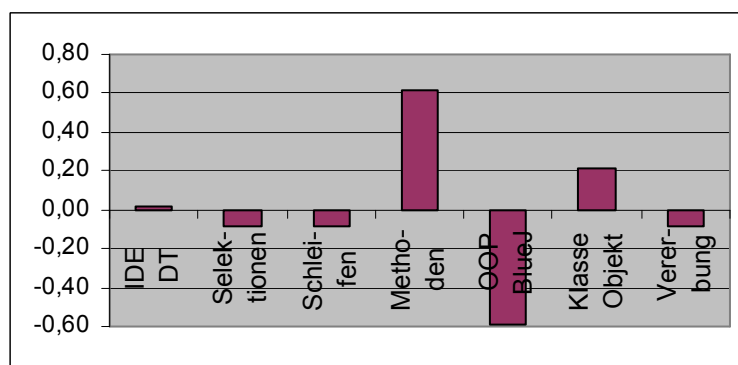


Abbildung 102: Schwere-Empfinden der OOP-Later-Klasse (Null-Linie:  $\emptyset = 2,6$ , vgl. [ES07a, Figure 8])

Für die OOP-Later-Klasse können folgende Aussagen (siehe Abbildung 102) gemacht werden [ES07a, S. 5]:

- Das erste Modul, der „Einstieg in den Texteditor“, wird als durchschnittlich schwer empfunden (Wert: 2,6) im Gegensatz zum Empfinden der OOP-First-Klasse, die dieses Modul als das schwerste Modul (Wert: 2,9) erachtet.
- Die nächsten beiden Module („Steuerstruktur Selektion“ und „Steuerstruktur Iteration“) werden (Wert: 2,5) etwas leichter (im Vergleich zur durchschnittlichen Schwere) empfunden, vergleichbar mit dem Empfinden der OOP-First-Schüler.

- Das Thema „Methoden“ (vor dem OOP-Einstieg) wird als schwer erachtet. Dies ist der einzige Wert (mit 3,2), der über dem Durchschnittswert (3,0) der Skala von 1 bis 5 liegt und sich somit im „negativen subjektiven Bereich“ befindet.
- Auch das Thema „Klasse und Objekt“, also die Klassen-Implementierung und Objekt-Erzeugung, wird als schwer erachtet. Der Wert (2,8) liegt etwas über dem Durchschnittswert (2,6) und das Thema wird auch als etwas schwerer als bei der OOP-First-Klasse (2,6) empfunden. Im Gegensatz zum Thema „Methoden“ ist es aber immer noch im „positiven subjektiven Bereich“.
- Das BlueJ-Modul wird als leichtestes Modul empfunden (2,0), noch leichter als es die OOP-First-Schüler sehen (2,4). Die Differenz zum vorangegangenen Modul „Methoden“ beträgt aber 1,2.
- Dafür fällt den OOP-Later-Schülern das Thema „Vererbung“ (2,5) leichter als den OOP-First-Schülern (2,8).

Rein optisch betrachtet wirkt das Diagramm der OOP-Later-Klasse „unruhig“. Das Maximum ist mit 0,6 wesentlich größer als das beim Diagramm der OOP-First-Klasse. Die Themen werden zwar durchschnittlich als gleich schwer erachtet, aber die Ausprägung bei den einzelnen Themen ist doch sehr unterschiedlich.

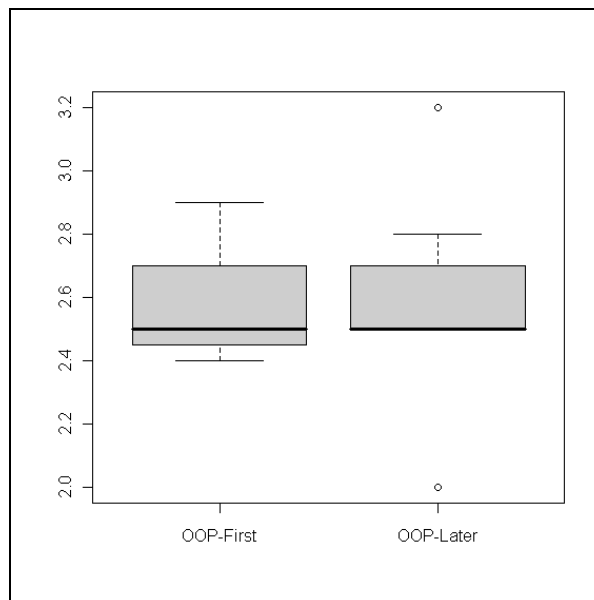


Abbildung 103: Vergleich des Schwere-Empfindens mit Hilfe eines Box-Whisker-Plots (vgl. [Fae07, S. 75-78])

Oder mit Hilfe des Box-Whisker-Plots in Abbildung 103 formuliert: Zwei Messwerte befinden sich außerhalb der Spannweite, gelten also als Ausreißer [SH09, S. 69]. Das untere Quartil ist gleich dem Minimum. Die Spannweite ist deutlich geringer als die des Plots der OOP-First-Werte. Die beiden Darstellungen sind auffällig unterschiedlich (bis auf die Position des Medianwertes).

#### c2) Das Schwierigkeits-Empfinden in Bezug auf die Ergebnisse des Vergleichstests

Bei der Vorstudie bestand der Test ja nur aus fünf Themen und einem Zusatzthema (vgl. Kapitel 3.3.12). Es findet sich von den Modul-Themen im Vergleichstest explizit nur das Thema „Methoden“ und evtl. das Thema „Klasse und Objekt“ (als „statische OOP“) wieder.

	Thema: Methoden		Thema: statische OOP	
	„Schwerefaktor“	Testergebnis	„Schwerefaktor“	Testergebnis
OOP-First	2,5	67%	2,6	72%
OOP-Later	3,2	64%	2,8	79%

Abbildung 104: Subjektiver Schwierigkeitsfaktor und objektives Testergebnis (vgl. Abbildung 90 und Abbildung 98)

Es scheint keinen Zusammenhang (siehe Abbildung 104) zwischen der „erlebten Schwere“ eines Themas (vgl. Abbildung 98) und den erzielten Lernergebnissen (vgl. Abbildung 90) zu geben.

c3) Erklärungsversuch für die Unterschiede im Schwierigkeits-Empfinden beim Thema „Methoden“

Das Modul 3 („Methoden“) ist für Schüler der OOP-First-Klasse leichter als für Schüler der OOP-Later-Klasse (siehe Abbildung 98). Die OOP-First-Klasse hat Methoden vom ersten Tag an (im BlueJ-Tool) benutzt. Im Modul 3 lernen dann die Schüler Methoden zu schreiben, die sie ein paar Wochen vorher schon benutzt haben [ES07a, S. 6]. Währenddessen findet bei der OOP-Later-Klasse die Implementierung und die Benutzung von Methoden zur gleichen Zeit statt. Sie implementieren diese Methoden als statische Methoden in der Hauptklasse, die die main-Methode enthält, da die Einführung in die OOP mit BlueJ erst später kommt. Dabei gibt es viele Fachtermini: aktueller und formaler Parameter, Rückgabewert, „call by value“ etc. Diese Vorgehensweise ist anscheinend ungünstig, da dieses Modul von den Schülerinnen und Schülern mit einem Schwierigkeitsfaktor von 3,2 mit Abstand als schwerstes Modul gewertet wird. Daher soll in der Hauptstudie das Modul 3 erst nach der „Einführung in die OOP mit BlueJ“ und nach dem Thema „Klasse und Objekt“ unterrichtet werden. Es soll dadurch ein niedrigeres Schwierigkeits-Empfinden bewirkt werden [ES07a, S. 6].

c4) Erklärungsversuche für die Unterschiede im Schwierigkeits-Empfinden beim Thema „IDE (Texteditor) und Datentypen“

Die Benutzung des textorientierten Java-Editors (Modul 10) fällt der OOP-Later-Klasse leichter als der OOP-First-Klasse (siehe Abbildung 98). Nach dem Start mit dem BlueJ-Tool setzt die OOP-First-Klasse den Unterricht mit dem Text-Editor fort, während die OOP-Later-Klasse im Rahmen des „Strukturierten-Programmierung-Einstiegs“ den Informatik-Anfangsunterricht gleich mit dem Text-Editor beginnt. Es scheint möglich zu sein, dass das BlueJ-Tool (mit der einfachen Benutzung der Objekte) einen „zu simplen“ Eindruck der Programmierung vermittelt [ES07a, S. 6]. Unterstützt wird dieser Eindruck evtl. auch durch die „bunte Oberfläche“ der ersten Beispiele, z. B. im Projekt „Figuren“ bzw. im Projekt „Bild“. Wenn dann im Texteditor Klassen implementiert und Objekte „per Hand“ erzeugt werden müssen, wird die Erstellung des Quellcodes als relativ komplex empfunden, auch im Hinblick auf die zu lernenden syntaktischen Regeln. Die OOP-Later-Klasse hat evtl. zwei Vorteile: Erstens kennt sie die relativ einfache und bunte BlueJ-Welt nicht und nimmt den Text-Editor daher eher als „normales“ Programmierung-Tool an und zweitens sind die Anfangsbeispiele einfacher, weil im Quellcode nicht Klassen implementiert, sondern in der main-Methode Variablen deklariert und einfache sequenzielle Algorithmen nach dem EVA-Prinzip implementiert werden.



c5) Erklärungsversuch für die Unterschiede in der Rangfolge der Themen im Hinblick auf die Bedeutung der Themen für das Gesamtverständnis

In diesem Kapitel wurde dargestellt, dass für die OOP-First-Schüler der Stoff der Themen „BlueJ“ und „Klasse und Objekt“ der Stoff ist, der am meisten zu ihrem Gesamtverständnis (der Themen) des Faches Informatik beigetragen hat (siehe Abbildung 97). Die Themen „Schleifen“, „Methoden“ und „Vererbung“ erhalten dagegen die schlechtesten Werte.

Bei den OOP-Later-Schülern ist das wichtigste Thema das Thema „BlueJ“. Die Themen „Schleifen“ und „Selektionen“ landen auf den letzten Plätzen, also zwei Themen, die als „prozedurale“ bzw. „strukturierte“ Programmierungsthemen noch vor den objektorientierten Themen unterrichtet wurden.

Die OOP-Later-Klasse hat kleinere Extremwerte (2,2 und 2,7) als die OOP-First-Klasse (1,9 und 2,9). Vielleicht liegt das daran (da die OOP-First-Klasse von Anfang an (bis zum Ende) auf die Benutzung von Klassen und Objekten fokussiert ist), dass die beiden Hauptthemen dazu („BlueJ“ und „Klasse und Objekt“) als wichtigste Themen erachtet werden. In der OOP-Later-Klasse hat das Thema „Klasse und Objekt“ eine andere Bedeutung, es ist eins von vielen Themen und wird nicht gleich am Anfang unterrichtet. Vielleicht wird daher (bis auf „BlueJ“) kein spezielles Thema herausgehoben, da die einzelnen Themen eher als gleich wichtig angesehen werden [ES07a, S. 6].

c6) Fachdidaktische Reaktionen

Beide fachdidaktischen Vorgehensweisen scheinen spezifische Vor- und Nachteile zu haben. Am Ende sollen die Interpretationen der Ergebnisse in dieser Arbeit dazu führen, dass Hinweise bzw. Verbesserungsvorschläge für den Informatik-Anfangsunterricht erfolgen. Hier könnten dies z. B. die folgenden fachdidaktischen Reaktionen sein:

- Die Umfrage-Ergebnisse zur Frage 6 sprechen (im Hinblick auf das Verständnis) sehr für den Einsatz des BlueJ-Tools im Informatikunterricht, und zwar für beide fachdidaktischen Vorgehensweisen.
- Der OOP-First-Klasse fällt der Umstieg auf den Texteditor schwer. Vielleicht kann der Lehrer (bei einem OOP-First-Vorgehen) die Schüler schon verstärkt im BlueJ-Tool mit dem dort vorhandenen Texteditor arbeiten lassen (also nicht nur Objekte erzeugen und Methoden aufrufen lassen) und damit den späteren Umstieg auf den Texteditor erleichtern.
- Der OOP-Later-Klasse fällt das Thema „Methoden“ schwer. Vielleicht kann der Lehrer (bei einem OOP-Later-Vorgehen) das Thema „Methoden“ erst nach der Einführung in die OOP explizit unterrichten und dadurch ein niedrigeres Schwere-Empfinden bewirken.

Speziell im Kapitel 7.5.4 werden weitere Vorschläge zur Verbesserung des Informatik-Anfangsunterrichts gemacht.

### 6.2.3 Zusammenfassung

Da die Vorstudie einige Design-Schwächen aufgezeigt hat (siehe Kapitel 3.3.14), wurden die Daten im letzten Kapitel nur punktuell ausgewertet und bei den Daten zum subjektiven Erleben keine Signifikanzberechnungen vorgenommen.

Auffällig bei dem Vergleichstest (siehe Kapitel 6.2.1) zum objektiven Aspekt ist, dass zwei Themen von drei OOP-Aufgaben-Bereichen (bei insgesamt sechs Themen) von der OOP-Later-Klasse signifikant besser bearbeitet wurden, das Thema „UML-Klassendiagramme und UML-Objektdiagramme“ und das Thema „Dynamische OOP“. Drei mögliche Gründe wurden formuliert. Entweder war das OOP-Later-Vorgehen besser, oder die zuletzt gelernten Themen waren am besten abrufbar oder der Unterricht war durch die mangelnde Synchronisation nicht gleich (z. B. wurden in der OOP-First-Klasse die einzelnen Themen schneller unterrichtet und somit die einzelnen Themen weniger detailliert dargeboten). Insgesamt hat die OOP-Later-Klasse beim Vergleichstest um 5% besser abgeschnitten als die OOP-First-Klasse. Wegen der tatsächlichen mangelnden Synchronisation des Unterrichts sollte dieses Ergebnis allerdings nicht verallgemeinert werden. Es zeigt aber deutlich, dass unterschiedlicher Unterricht unterschiedlichen Lernerfolg bewirken kann.

Der Fragebogen (siehe Kapitel 6.2.2) zum subjektiven Aspekt generierte Antworten zu acht Fragen. Bei den drei Fragen zum subjektiven Erleben des Unterrichts gefällt den Schülern das Fach Informatik unabhängig von der fachdidaktischen Vorgehensweise, sie fühlen sich wohl und motiviert. Bei der einzigen Frage zum Lehrer ergeben die Antworten, dass die beiden Lehrer zwar unterschiedlich gesehen werden, aber beide einen relativ hohen Sympathiewert besitzen. Auswirkungen der Unterschiede sind auf den ersten Blick nicht zu erkennen: Die Klasse mit dem sympathischeren Lehrer erzielt sogar schwächere Ergebnisse. Bei den vier Fragen zum subjektiven Erleben der einzelnen Themen lagen alle Mittelwerte im positiven Bereich, die Mittelwerte der beiden Klassen sind entweder gleich oder unterscheiden sich nur um den Faktor 0,1.

Interessant ist eine Betrachtung der Antworten bezogen auf die einzelnen Themen, die Themenreihenfolge oder die Themenübergänge. Hier wurden im Kapitel 6.2.2 einzelne Aspekte herausgegriffen mit dem Schwerpunkt „Schwierigkeits-Empfinden“. Auffällig ist z. B., dass das subjektive Schwierigkeits-Empfinden der einzelnen Themen nicht korreliert mit den objektiven Ergebnissen des Vergleichstests und dass das BlueJ-Tool als sehr leicht und wirksam von den Schülerinnen und Schülern erlebt wird (unabhängig von der fachdidaktischen Vorgehensweise). Am Ende des Kapitels 6.2.2 wurden dann schon erste mögliche fachdidaktische Reaktionen auf die Ergebnisse genannt.

Vertiefende fachdidaktische Diskussionen sollen aber erst mit den Ergebnissen der Hauptstudie erfolgen (siehe Kapitel 6.3, Kapitel 6.4 und Kapitel 7), deren Studiendesign im Gegensatz zur Vorstudie wesentlich verbessert wird (siehe Kapitel 3.3.14). Der Unterricht wird mehr synchronisiert, statt Modulen gibt es Themen, die Themen werden gleichartiger gestaltet, die Anzahl der subjektiven Fragen wird deutlich erhöht und der Vergleichstest wird sich auf alle unterrichteten Themen beziehen. Außerdem werden ein Vortest und ein Nachhaltigkeitstest eingeführt. Damit sollen die Datenerhebung und die Datenauswertung seriöser bzw. auf ein höheres wissenschaftliches Niveau gestellt werden.

## 6.3 Hauptstudie: Der Lernerfolg nach einem Schuljahr

Im Folgenden sollen die Resultate der Hauptstudie aus dem Schuljahr 2007/2008 im Hinblick auf den Lernerfolg nach einem Schuljahr ausführlich dargestellt werden. Hierfür muss der Nachweis über den Kenntnisstand am Anfang des Schuljahres erbracht werden (mit Hilfe eines Vortests, siehe Kapitel 6.3.1) und der Lernerfolg direkt am Ende des Schuljahres (mit Hilfe eines Vergleichstests, siehe Kapitel 6.3.2) bzw. zu Beginn des darauf folgenden Schuljahres (mit Hilfe eines Nachhaltigkeitstests, siehe Kapitel 6.3.3) ermittelt werden. Letztlich kann daraus eine Lernkurve generiert werden (siehe Kapitel 6.3.4). Das letzte Unterkapitel 6.3.5 widmet sich speziellen Lerngruppen, zum einen den sehr leistungsstarken bzw. sehr leistungsschwachen Schülern, zum anderen den Schülerinnen.

Die ersten vier Unterkapitel sind (teilweise wörtlich, teilweise überarbeitet) aus zwei Veröffentlichungen aus dem Jahr 2009 entnommen, den Papieren zur ICER-Konferenz in Berkeley/USA (siehe [ES09a]) und zur INFOS in Berlin (siehe [ES09c]).

### 6.3.1 Vortest (Pre-Test)

Im Kapitel 4.3.3 wurde begründet, warum der Vortest ein um die Hälfte verkürzter Vergleichstest im Hinblick auf die Themen 1 bis 8 ist, warum das Thema 9 („Assoziation“) und die beiden Zusatzthemen („Dynamische OOP“ und „OOM“) nicht abgefragt werden und warum der Vortest in einem anderen Jahrgang geschrieben wird.

Die beiden Klassen OG81 und OG82 mussten gleich zu Beginn des Schuljahres 2008/2009 am 03.09.2008 (in der gleichen Woche schrieben die ehemaligen Schüler der OG71 und OG72 den Nachhaltigkeitstest) den Vortest schreiben. In jeder Klasse bekam ein Teil der Schüler den Teil 1 (siehe Anhang Q) mit den Aufgaben 1 und 3 und der kombinierten Aufgabe 6/7 und der andere Teil der Schüler den Teil 2 (siehe Anhang R) mit den Aufgaben 2, 4, 5 und 8. Jeder Schüler musste also nur vier Aufgaben lösen und konnte somit maximal 40 Punkte erreichen. Die Ergebnisse des Vortests (siehe Anhang S) zeigt Abbildung 105.

Nr.	Thema	Punkte	OG81	OG82	Durchschnitt
1	Datentypen und Steuerstruktur Sequenz	10	1,8	1,7	1,75
3	Steuerstruktur: Iteration	10	1,0	0,7	0,85
6 / 7	Einführung in die OOP / Klasse und Objekt	20	1,7	1,5	1,60
	<b>Summe (Gruppe 1)</b>	<b>40</b>	<b>4,5</b>	<b>3,9</b>	<b>4,20</b>
2	Steuerstruktur: Selektion	10	0,9	1,2	1,05
4	Arrays und Strings	10	0,7	0,5	0,60
5	Methoden	10	1,2	1,2	1,20
8	Vererbung	10	0,4	1,1	0,75
	<b>Summe (Gruppe 2)</b>	<b>40</b>	<b>3,2</b>	<b>4,0</b>	<b>3,60</b>
	<b>Summe</b>	<b>80</b>	<b>7,7</b>	<b>7,9</b>	<b>7,80</b>
	<b>Prozentual</b>	<b>100</b>	<b>9,6%</b>	<b>9,9%</b>	<b>9,75%</b>

Abbildung 105: Ergebnisse des Vortests am Ende der Hauptstudie im Spätsommer 2008 (siehe Daten in Anhang S)

Die Ergebnisse bestätigen die Vermutung, dass die Schülerinnen und Schüler nahezu ohne Programmier-Vorkenntnisse an das OSZ IMT kommen: Sie erzielten im Durchschnitt nur 3,9 von 40 Punkten. Dies sind weniger als 10% und würde der Schulnote 6 (siehe Bewertungsschlüssel in Abbildung 153) entsprechen.

Und das, obwohl der Vortest (mit AFI / AFII / AFIII = 24 / 76 / 0) geringfügig leichter ist als der Vergleichstest (mit AFI / AFII / AFIII = 21 / 79 / 0), da drei Themen gar nicht abgefragt werden.

Dass überhaupt einige Punkte erzielt wurden, liegt m. E. auch weniger am Vorwissen als an dem Anteil der Multiple-Choice-Aufgaben.

Die Spaltenüberschriften in Abbildung 105 haben die Klassennamen (OG81 / OG82) und nicht die Namen der fachdidaktischen Vorgehensweisen (OOP-First / OOP-Later), weil ja noch keine fachdidaktische Vorgehensweise angewendet wurde. Daher sind auch die Spalten nicht mit „Blassblau“ bzw. „Pflaume“ farblich gekennzeichnet.

### 6.3.2 Vergleichstest (Post-Test)

Die beiden Klassen OG71 und OG72 schrieben am Ende des Schuljahres 2007/2008 den Vergleichstest (siehe Anhang M und Kapitel 4.3.2). Am 01.07.2008 bekamen alle Schüler beider Klassen zur gleichen Zeit die beiden Zusatzaufgaben 10 und 11 und hatten 30 Minuten Zeit für die Lösung. Zwei Tage später, am 03.07.2008, bekamen die Schüler die Aufgaben 1 bis 9 und 100 Minuten Zeit. Jeder Schüler musste insgesamt 49 Teil-Aufgaben lösen und konnte maximal 110 Punkte erreichen. Die Ergebnisse des Vergleichstests (siehe Anhang N) zeigt Abbildung 106<sup>1</sup>.

Folgende Aussagen [ES09a, S. 19] können getroffen werden (siehe Abbildung 106):

- Die OOP-First-Klasse ist im Gesamtergebnis (um 4,3%) besser als die OOP-Later-Klasse. Der Unterschied ist aber nicht signifikant.
- Die Differenz zwischen den Mittelwerten der beiden Klassen beim Thema „Arrays und Strings“ (OOP-First: 5,7 / OOP-Later: 3,7) ist signifikant ( $p = 0,022$ ). Die Irrtumswahrscheinlichkeit beträgt nur 2,2%. Die Effektstärke ( $d = 0,78$ ) liegt allerdings knapp unter dem Wert 0,8; der Mittelwertsunterschied stellt somit keinen starken, sondern einen mittelstarken Effekt dar (vgl. [Coh88, S. 20-27]).
- Die Differenz zwischen den Mittelwerten der beiden Klassen beim Thema „Assoziation“ (OOP-First: 6,2 / OOP-Later: 4,9) ist nicht signifikant ( $p = 0,068$ ). Die Irrtumswahrscheinlichkeit beträgt 6,8%.
- Die Differenz zwischen den Mittelwerten der beiden Klassen beim Thema „Methoden“ (OOP-First: 6,1 / OOP-Later: 4,8) ist auch nicht signifikant ( $p = 0,119$ ). Die Irrtumswahrscheinlichkeit beträgt 12%.

Alle drei zuletzt genannten Themen sind die schwersten Themen des Vergleichstests.

---

<sup>1</sup> Hinweis: Die Ergebnisse sind insofern schon angepasst, dass hier nur Schülerinnen und Schüler aufgeführt sind, die auch den späteren Nachhaltigkeitstest mitgeschrieben haben, damit die Ergebnisse vergleichbar sind. Dies war durch die Cut-off-Kriterien (siehe Abbildung 42) in Kapitel 3.3.15 festgelegt worden. Am Ende verbleiben für die Test-Auswertung in beiden Klassen je 19 Personen. Nur für diese Personen kam es auch zu einer Neuberechnung der Mathematik-Durchschnittsnote, so dass es kleine Werte-Abweichungen in Abbildung 106 (mit 8,7 und 8,2) im Vergleich zu den Werte-Angaben in den zitierten Quellen (8,8 und 7,8) gibt.

Lfd. Nr.	Thema	Punkte	OOP-First	OOP-Later	p-Wert
1	2) Datentypen und Steuerstruktur Sequenz	10	6,8	7,0	0,742
2	5) Steuerstruktur Selektion	10	6,3	7,1	0,211
3	6) Steuerstruktur Iteration	10	5,8	6,0	0,780
4	7) Arrays und Strings	10	<b>5,7*</b>	<b>3,7*</b>	0,022
5	4) Methoden	10	6,1	4,8	0,119
6 / 7	1) Einführung in die OOP 3) Klassenimplementierung und Objekterzeugung	20	14,6	14,9	0,756
8	8) Vererbung	10	7,6	6,9	0,354
9	9) Assoziation	10	6,2	4,9	0,068
10	Zusatz: Dynamische OOP	10	7,1	7,0	0,861
11	Zusatz: Objektorientierte Modellierung (OOM)	10	6,7	5,9	0,404
	<b>Gesamt</b>	110	72,9	68,2	
	<b>Prozentual</b>	100	66,3	62,0	
	<i>Mathematik-Note (von 15 Punkten)</i>		8,7	8,2	

Abbildung 106: Ergebnisse des Vergleichstests in der Hauptstudie 2007/2008

([ES09a, Fig. 6, S. 19] [ES09c, Abb. 7, S. 128]; \* signifikante Ergebnis-Unterschiede sind fett gedruckt; siehe Daten in Anhang N)

Drei Erklärungsversuche sollen für das bessere Gesamtergebnis der OOP-First-Klasse gegeben werden:

1. Die OOP-First-Klasse ist von der mathematischen Leistungsfähigkeit (um ca. 3,3%) besser als die OOP-Later-Klasse (zur Erinnerung: Es wurde derselbe Mathematiklehrer in beiden Klassen eingesetzt). Dieser Unterschied macht sich nun auch in der Informatik bemerkbar, wenn man eine Korrelation zwischen Mathematik- und Informatik-Note voraussetzt [ES09c, S. 128].
2. Das OOP-First-Vorgehen ist verantwortlich für das bessere Resultat bzw. das OOP-Later-Vorgehen ist verantwortlich für das schlechtere Resultat.
3. Wenn man sich die Ergebnisse des Nachhaltigkeits- bzw. Follow-Up-Tests (acht Wochen später) ansieht (siehe nächstes Kapitel), dann fallen die deutlichen Einbrüche bei den Ergebnissen der OOP-First-Klasse auf, sowohl im Endergebnis als auch speziell bei den vorher signifikant unterschiedlichen Themen-Ergebnissen. Bei der Ursachenforschung wurde schnell klar: Beim Vergleichstest (im Gegensatz zum Nachhaltigkeitstest) sind die Klassen von den beiden beteiligten Informatik-Lehrern sehr unterschiedlich vorbereitet worden [ES09c, S. 129] [ES09a, S. 19]. Daher sind die deutlich objektiveren Ergebnisse jene des Nachhaltigkeitstests (auf den überhaupt nicht vorbereitet wurde). Diese Ergebnisse werden daher im nächsten Unterkapitel ausführlich dargestellt.

Wenn man die dritte Erklärung favorisiert, dann können die Ergebnisse des Vergleichstests leider nicht weiter diskutiert werden. Bis auf eine interessante Aussage: Gerade bei schweren Themen (für die OOP-Later-Klasse sind dies die Themen „Arrays und Strings“, „Methoden“ und „Assoziation“ mit den Ergebnissen 3,7 / 4,8 / 4,9) lassen sich durch eine spezielle Präparation einer Klasse deutlich bessere Ergebnisse (OOP-First: 5,7 / 6,1 / 6,2) erreichen. Bei leichteren Themen macht sich dies nicht so stark bemerkbar.

### 6.3.3 Nachhaltigkeitstest (Follow-Up-Test)

Die Schülerinnen und Schüler der beiden ehemaligen Klassen OG71 und OG72 schrieben zu Beginn des Schuljahres 2008/2009 (also gleich nach den Sommerferien am Anfang des Kurssystems) den Nachhaltigkeitstest (siehe Anhang O und Kapitel 4.3.4). Am 02.09.2008 bekamen alle so genannten Q1-Schüler des Beruflichen Gymnasiums des OSZ IMT zur gleichen Zeit die Aufgaben 1 bis 9 und hatten 100 Minuten Zeit für die Lösung. Einen Tag später, am 03.09.2008, bekamen die Schüler die Zusatzaufgaben 10 und 11 und 30 Minuten Zeit für die Bearbeitung. Jeder Schüler musste insgesamt 48 Teil-Aufgaben lösen und konnte maximal 110 Punkte erreichen. Die Ergebnisse des Nachhaltigkeitstests (siehe Anhang P) zeigt Abbildung 107<sup>2</sup>.

Lfd. Nr.	Thema	Punkte	OOP-First	OOP-Later	p-Wert
1	2) Datentypen und Steuerstruktur Sequenz	10	6,4	7,2	0,177
2	5) Steuerstruktur Selektion	10	6,5	7,4	0,258
3	6) Steuerstruktur Iteration	10	5,7	5,4	0,693
4	7) Arrays und Strings	10	3,6	3,8	0,785
5	4) Methoden	10	5,3	4,6	0,453
6 / 7	1) Einführung in die OOP 3) Klassenimplementierung und Objekterzeugung	20	13,4	13,0	0,763
8	8) Vererbung	10	5,8	6,0	0,832
9	9) Assoziation	10	2,5	1,6	0,329
10	Zusatz: Dynamische OOP	10	5,3	5,1	0,733
11	Zusatz: Objektorientierte Modellierung (OOM)	10	5,3	6,3	0,346
	<b>Gesamt</b>	110	59,8	60,4	
	<b>Prozentual</b>	100	54,4	54,9	

Abbildung 107: Ergebnisse des Nachhaltigkeitstests in der Hauptstudie 2007/2008

([ES09a, Fig. 7, S. 19] [ES09c, Abb. 8, S. 129]; es gibt keine signifikanten Ergebnisunterschiede; siehe Daten in Anhang P)

Es lassen sich die folgenden Aussagen (siehe Abbildung 106, Abbildung 107 und Abbildung 108) treffen:

- Die Ergebnisse der OOP-First-Klasse und der OOP-Later-Klasse im Nachhaltigkeitstest sind fast gleich. Es gibt keine (signifikanten) Unterschiede, weder im Gesamtergebnis noch in den Teilresultaten.
- Der eine signifikante Unterschied des Vergleichstests ist verschwunden.
- Jetzt ist die OOP-Later-Klasse etwas besser (0,5%). Und dies, obwohl sie vom Mathematik-Leistungsvermögen um 3,3% schlechter bewertet worden ist.
- Die OOP-First-Klasse hat um 11,9% schlechter abgeschnitten als im Vergleichstest, die OOP-Later-Klasse nur um 7,1%.

<sup>2</sup> Hinweis: Die Ergebnisse sind insofern schon angepasst, dass hier nur Schülerinnen und Schüler aufgeführt sind, die auch den Vergleichstest mitgeschrieben haben, damit die Ergebnisse vergleichbar sind. Dies war durch die Cut-off-Kriterien (siehe Abbildung 42) in Kapitel 3.3.15 festgelegt worden. Am Ende verbleiben für die Test-Auswertung zu Beginn des Kurssystems 19 Personen.

Im Kapitel 7.3.1 werden die Ergebnisse dahingehend analysiert, welche Themen leicht und schwer waren und im Kapitel 7.3.2 wird beschrieben, ob objektorientierte Themen schwerer waren als prozedurale Themen.

Folgende zwei Erklärungsversuche sollen gegeben werden:

1. Das OOP-First-Vorgehen zeigt kurzfristig bessere Erfolge bei schlechter Nachhaltigkeit bzw. das OOP-Later-Vorgehen zeigt kurzfristig schlechtere Erfolge bei besserer Nachhaltigkeit.
2. Die OOP-First-Klasse ist für den Vergleichstest speziell präpariert worden. Daher ist sie in diesem Test deutlich besser, speziell bei den schweren Themen. Diese spezielle Vorbereitung konnte im Nachhaltigkeitstest unterbunden werden. Daher schneidet jetzt die OOP-First-Klasse deutlich schlechter ab als im Vergleichstest.

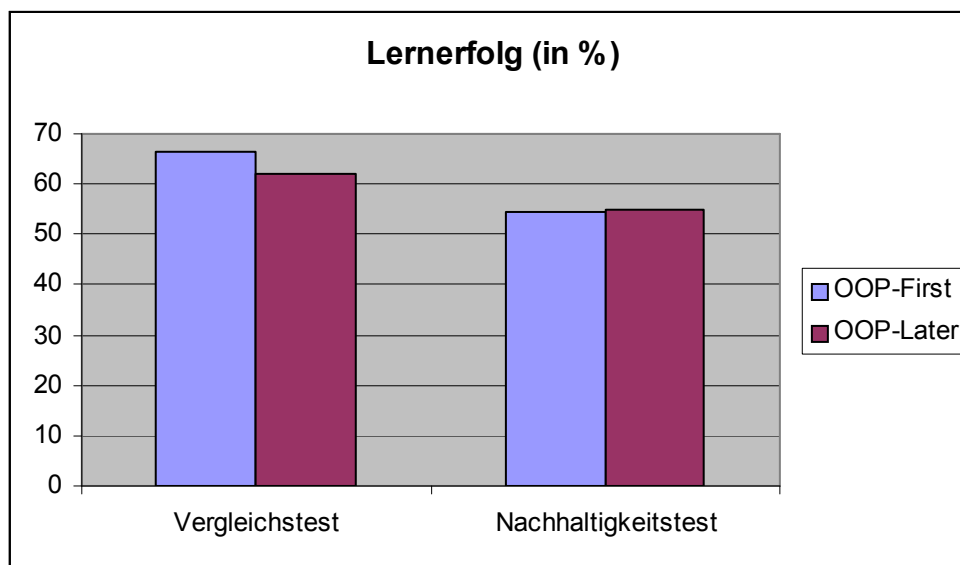


Abbildung 108: Vergleichs- und Nachhaltigkeitstest in der Hauptstudie 2007/2008 (bezogen auf alle elf Themengebiete)

Geht man von der Richtigkeit der zweiten Erklärung aus, so können folgende Ergebnisse formuliert werden:

- Es wurden keine signifikanten Unterschiede zwischen dem OOP-First-Vorgehen und dem OOP-Later-Vorgehen im Hinblick auf den Lernerfolg festgestellt bzw. gemessen. Sowohl für das Gesamtergebnis als auch für die einzelnen Themenresultate ergaben sich keine signifikanten Unterschiede [ES09c, S. 129]. Beide Vorgehensweisen produzierten (spätestens nach zwei Monaten) vergleichbare Lernerfolge im verwendeten Test.
- Die Ergebnisse stimmen überein mit den Untersuchungen von Vilner, Zur und Gal-Ezer [VZG07] bzw. Tew, McCracken und Guzdial [TMG05], bei denen auch keine Unterschiede gefunden wurden bzw. vorhandene Unterschiede nach einiger Zeit verschwanden [ES09a, S. 19].

Ein positives Ergebnis lässt sich für den Schulalltag ableiten: Der Lernerfolg in dem Fach Informatik ist relativ nachhaltig. In knapp zwei Monaten Ferien werden nur 7,1% des vorher gelernten Stoffes vergessen (ausgehend von den Daten der OOP-Later-Schüler).

### 6.3.4 Verlauf des Lernerfolgs

Im Folgenden soll der Verlauf des Lernerfolgs (Ergebnisse aller drei Tests in Prozent, siehe Abbildung 109 und Abbildung 110) bezogen auf das ganze Schuljahr dargestellt werden. Im Vortest wurden nur die ersten acht Themen abgefragt, daher bezieht sich der Vergleich nur auf diese Themen und somit wurden vom Vergleichs- und Nachhaltigkeitstest auch nur die entsprechenden Themenergebnisse berücksichtigt. Die maximal erreichbare Punktzahl war 80, dies entspricht in den beiden Abbildungen 100%.

Klasse	„Vortest“		Vergleichstest (nur Aufgaben 1-8)		Nachhaltigkeitstest (nur Aufgaben 1-8)	
	Punkte (max. 80)	Prozentual	Punkte (max. 80)	Prozentual	Punkte (max. 80)	Prozentual
OG81	7,7	9,6				
OG82	7,9	9,9				
OG81/82: Ø	7,8	9,75				
OG71 (OOP-First)			52,9	66,1	46,7	58,4
OG72 (OOP-Later)			50,4	63,0	47,4	59,3

Abbildung 109: Ergebnisse der Tests in der Hauptstudie 2007/2008

Die Schüler beginnen praktisch ohne Programmier-Vorkenntnisse das Schuljahr. Die 9,75% als gemittelter Wert der beiden Klassen OG81 und OG82 im „Vortest“ (dieser Wert entspricht der Schulnote 6, vgl. Abbildung 153) sind eher den Multiple-Choice-Aufgaben geschuldet als dem Vorwissen der Schüler. Die Schüler der beiden Klassen OG71 und OG72 erreichen innerhalb eines Schuljahres einen Kenntnisstand von 66,1% (OOP-First, Note: 3) bzw. 63,0% (OOP-Later, Note: 3-). Nach den Sommerferien sinken die Werte auf 58,4% (Note: 4+) bzw. auf 59,3% (Note: 4+).

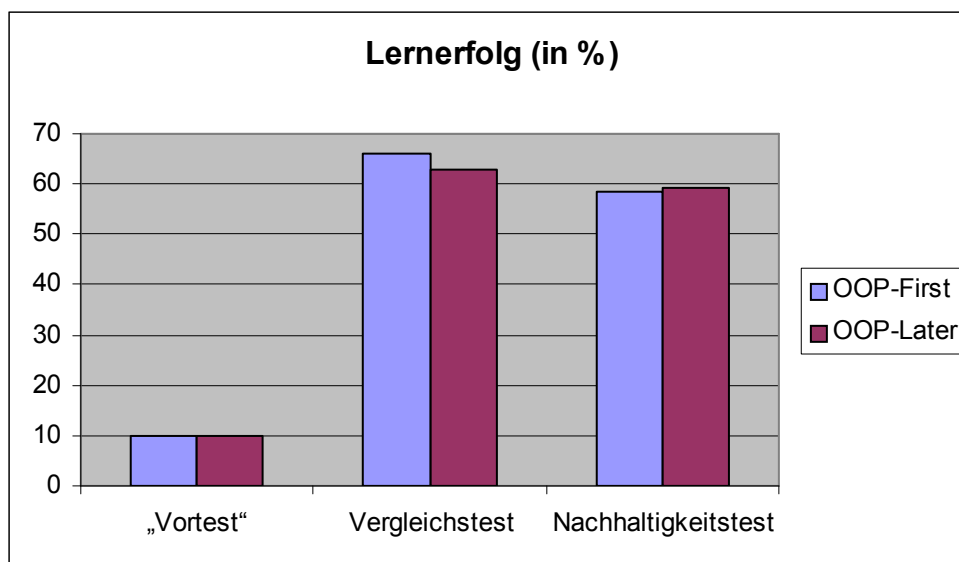


Abbildung 110: Lernerfolg in der Hauptstudie 2007/2008 bezogen auf die ersten acht Themengebiete ([ES09a, Fig. 8, S. 20] [ES09c, Abb. 10, S. 130]; Vortest erfolgte zeitgleich zum Nachhaltigkeitstest im Schüler-Jahrgang 2008/2009)



Die Unterschiede in den Wissenskurven sind nicht signifikant. Nimmt man (aus dem mehrfach erwähnten Grund) wieder nur die OOP-Later-Klasse in den Fokus, so kann man sagen, dass bei den Themen 1 bis 8 nur 3,7% des Wissens verloren geht [ES09a, S. 20] [ES09c, S. 130]. Die Nachhaltigkeit (bezogen auf zwei Monate) ist also relativ groß.

### 6.3.5 Ausgewählte Lerngruppen

Im Folgenden sollen drei spezielle Stichproben bzw. Lerngruppen untersucht werden, die (sehr) leistungsstarken Schülerinnen und Schüler, die (sehr) leistungsschwachen Schülerinnen und Schüler und nur die Schülerinnen.

#### 1. Die Lerngruppen der leistungsstärksten und leistungsschwächsten Schülerinnen und Schüler

Erst einmal muss die Frage geklärt werden, wie man die Gruppe der (sehr) leistungsstarken bzw. (sehr) leistungsschwachen Schüler definiert?

Man könnte zum einen den „Extremgruppen-Begriff“ von Bortz und Döring [BD02, S. 219] zu Grunde legen. Sie sprechen von zwei Extremgruppen, den 25% Untersuchungsteilnehmern mit den höchsten Testwerten und den 25% Untersuchungsteilnehmern mit den niedrigsten Testwerten. 25% von 19 Schülern, die in beiden Klassen sowohl den Vergleichs- als auch den Nachhaltigkeitstest geschrieben haben, ergibt die Zahl **4,75**.

Man könnte zum anderen von dem Notenschlüssel im Schulbereich ausgehen und z. B. definieren, dass es sich ab der Note 2 um einen leistungsstarken Schüler handelt. Nach dem Notenschlüssel der gymnasialen Oberstufe (siehe Abbildung 153) gibt es ab 80%-Leistung die Note 2. Sieht man sich die Ergebnisse aus dem Vergleichstest für beide Klassen (siehe Anhang N) im Hinblick darauf an, welche Schüler die 80%-Marke erreicht haben, dann sind dies in beiden Klassen nur **3** Schüler.

Aus diesen beiden Überlegungen heraus wird jetzt für die Aufteilung in leistungsstarke und leistungsschwache Schüler als Kompromiss zwischen den Zahlen 4,75 und 3 die Schülerzahl **4** genommen. Und die vier leistungsstärksten und die vier leistungsschwächsten Schüler aus beiden Klassen werden aus dem Nachhaltigkeitstest ermittelt (wg. der unterschiedlichen Vorbereitung der beiden Klassen im Vergleichstest).

Die Abbildung 111 zeigt die Code-Namen der ermittelten leistungsstärksten Schüler, ihre Leistungen im Vergleichs- und im Nachhaltigkeitstest und ihre Mathematiknoten (in erreichten Punkten von max. 15 Punkten).

Obwohl die vier leistungsstärksten Schüler der OOP-First-Klasse in Mathematik durchschnittlich um einen Punkt besser sind (entspricht ca. 5% Leistungsunterschied), erreichen sie sowohl bei dem Vergleichstest als auch beim Nachhaltigkeitstest durchschnittlich weniger Punkte als die vier leistungsstärksten Schüler der OOP-Later-Klasse. Die Unterschiede sind aber nicht signifikant.

Es lassen sich die folgenden Aussagen machen:

- Es gibt bei der Lerngruppe der leistungsstärksten Schüler keine signifikanten Unterschiede im Lernerfolg in Abhängigkeit von der fachdidaktischen Vorgehensweise (Nachhaltigkeitstest:  $p = 0,90$ ).

- Eine spezielle Präparierung leistungsstarker Schüler auf einen Test hin zeigt keine offensichtliche Auswirkung.
- Die Nachhaltigkeit des Wissens ist bei den leistungsstarken OOP-First-Schülern größer (nur 4,1% Wissensverlust) als bei den leistungsstarken OOP-Later-Schülern (7% Wissensverlust). Dieser Unterschied ist aber nicht signifikant.

OOP-First	Vergleichstest	Nachhaltigkeitstest	Mathematiknote
ANPE	75,5	70,9	13
SIJO	82,7	81,4	10
SUNO	85,5	76,4	10
NUYU	79,5	78,2	14
<b>Durchschnitt</b>	<b>80,8</b>	<b>76,7</b>	<b>11,8</b>
OOP-Later	Vergleichstest	Nachhaltigkeitstest	Mathematiknote
BIHO	74,1	71,4	7
INWO	84,5	76,8	10
LASA	87,3	84,1	14
SADI	90,9	76,4	12
<b>Durchschnitt</b>	<b>84,2</b>	<b>77,2</b>	<b>10,8</b>

Abbildung 111: Ergebnisse der leistungsstärksten Schüler

Die Abbildung 112 zeigt die Code-Namen der ermittelten leistungsschwächsten Schüler, ihre Leistungen im Vergleichs- und im Nachhaltigkeitstest und ihre Mathematiknoten (in erreichten Punkten von max. 15 Punkten).

OOP-First	Vergleichstest	Nachhaltigkeitstest	Differenz	Mathematiknote
CLUL	57,7	32,3	25,4	7
LINO	42,3	22,7	19,6	6
REDE	50,5	33,6	16,9	8
SIPE	42,7	24,1	18,6	7
<b>Durchschnitt</b>	<b>48,3</b>	<b>28,2</b>	<b>20,1*</b>	<b>7,0</b>
OOP-Later	Vergleichstest	Nachhaltigkeitstest	Differenz	Mathematiknote
BETH	52,7	44,5	8,2	8
CHMI	38,6	30,5	8,1	9
JUKL	33,6	28,6	5,0	6
TAWA	56,8	40,5	16,3	10
<b>Durchschnitt</b>	<b>45,4</b>	<b>36,0</b>	<b>9,4*</b>	<b>8,3</b>

Abbildung 112: Ergebnisse der leistungsschwächsten Schüler (\* signifikanter Unterschied)

Die vier leistungsschwächsten Schüler der OOP-First-Klasse sind in Mathematik durchschnittlich um 1,3 Notenpunkte schlechter (entspricht ca. 6,5% Leistungsunterschied) als die vier leistungsschwächsten Schüler der OOP-Later-Klasse. Während die OOP-First-Schüler bei dem Vergleichstest mit durchschnittlich 48,3% noch ein

bisschen besser sind als die leistungsschwächsten OOP-Later-Schüler (mit 45,4%), zeigt der Nachhaltigkeitstest eine große Differenz zu Gunsten der OOP-Later-Schüler (28,2% bzw. 36%). Mit einem p-Wert von 0,1542 (dies entspricht einer Irrtumswahrscheinlichkeit von 15,42%) beim t-Test ist der Unterschied beim Nachhaltigkeitstest aber nicht signifikant.

Es lassen sich die folgenden Aussagen machen:

- Es gibt bei der Lerngruppe der leistungsschwächsten Schüler offensichtliche Unterschiede im Lernerfolg (im Nachhaltigkeitstest) in Abhängigkeit von der fachdidaktischen Vorgehensweise, die aber nicht signifikant sind.
- Die Nachhaltigkeit des Wissens erscheint bei den leistungsschwachen OOP-First-Schülern wesentlich geringer (20,1% Wissensverlust) als bei den leistungsschwachen OOP-Later-Schülern (nur 9,4% Wissensverlust). Diese Differenz ist signifikant ( $p = 0,0139$ ), die Irrtumswahrscheinlichkeit beträgt nur 1,4%.

Die Ergebnisse lassen sich wie folgt interpretieren:

- Der Ergebnis-Unterschied im Nachhaltigkeitstest (7,9%), der nicht signifikant ist, lässt sich sowohl mit der unterschiedlichen mathematischen Leistungsfähigkeit der Schüler als auch mit der unterschiedlichen fachdidaktischen Vorgehensweise erklären. Welche Begründung zutrifft, bleibt an dieser Stelle offen. Sollte die zweite Begründung zutreffen, so müsste sie wie folgt formuliert werden: Das OOP-Later-Vorgehen zeigt bei leistungsschwachen Schülern einen höheren Lernerfolg (im Hinblick auf die Nachhaltigkeit nach 2 Monaten) als das OOP-First-Vorgehen.
- Der Differenz-Unterschied in den beiden Klassen zwischen Vergleichstest und Nachhaltigkeitstest (also die Nachhaltigkeit des Wissens), der signifikant ist, lässt sich durch die spezielle Präparierung der leistungsschwachen OOP-First-Schüler auf den Vergleichstest hin erklären. Diese spezielle Vorbereitung zeigt bei leistungsschwachen Schülern (im Gegensatz zu leistungsstarken Schülern) offensichtliche Auswirkung, sie profitieren stark davon.

## 2. Die Lerngruppe der Schülerinnen

Im Folgenden soll die spezielle Lerngruppe der Schülerinnen untersucht werden. In beiden Klassen waren von den 19 Personen je zwei weiblich.

<b>OOP-First</b>	<b>Vergleichstest</b>	<b>Nachhaltigkeitstest</b>	<b>Mathematiknote</b>
ANFR	65,0	46,8	6
LINO	42,3	22,7	6
<b>Durchschnitt</b>	<b>53,7</b>	<b>34,8</b>	<b>6</b>
<b>OOP-Later</b>	<b>Vergleichstest</b>	<b>Nachhaltigkeitstest</b>	<b>Mathematiknote</b>
BERE	63,2	44,5	7
CARU	69,5	50,5	5
<b>Durchschnitt</b>	<b>66,4</b>	<b>47,5</b>	<b>6</b>

Abbildung 113: Ergebnisse der Schülerinnen

Die Abbildung 113 zeigt die Code-Namen der Schülerinnen, ihre Leistungen im Vergleichs- und im Nachhaltigkeitstest und ihre Mathematiknoten (in erreichten Punkten von max. 15 Punkten). Aufgrund der Tatsache, dass es sich um eine sehr kleine Stichprobe handelt, soll im Folgenden sehr vorsichtig formuliert werden.

Man könnte auf den ersten Blick einen Vorteil des OOP-Later-Vorgehens sehen: Sowohl im Vergleichstest als auch im Nachhaltigkeitstest sind die durchschnittlichen Ergebnisse der OOP-Later-Schülerinnen deutlich höher. Die Ursache für den Unterschied würde ich aber weniger in der fachdidaktischen Vorgehensweise sehen als in der Tatsache, dass die sehr schwache Schülerin LINO den Durchschnitt der OOP-First-Stichprobe stark nach unten zieht, weil in jeder Lerngruppe nur zwei Schülerinnen enthalten sind. Ich entscheide mich aufgrund der kleinen Stichprobe dazu, keine Aussagen zu formulieren und auch keinen Signifikanztest vorzunehmen.

Ein interessanter Nebenaspekt sei aber erwähnt: Obwohl nur eine der vier Schülerinnen auch in der Lerngruppe der leistungsschwächsten Schüler vertreten ist (die Schülerinnen-Lerngruppe also in den Informatik-Ergebnissen „höherwertiger“ ist), ist die durchschnittliche Mathematik-Note bei den Schülerinnen niedriger als bei den leistungsschwächsten Schülern. Sie beträgt gerade einmal 6 Punkte (Note: 4+). Die leistungsschwächsten OOP-First-Schüler kommen im Durchschnitt auf 7 Punkte (Note: 3-) und die leistungsschwächsten OOP-Later-Schüler sogar auf 8,3 Punkte (Note: 3). Die Schülerinnen erreichen im Fach Informatik bessere Zensuren als im Fach Mathematik.

### 6.3.6 Zusammenfassung

Die Ergebnisse des Vortests (siehe Kapitel 6.3.1) bestätigen die Vermutung, dass die Schülerinnen und Schüler nahezu ohne Programmier-Vorkenntnisse an das OSZ IMT kommen. Sie erzielen im Durchschnitt nur 3,9 von 40 Punkten. Dies sind weniger als 10% und würde der Schulnote 6 entsprechen.

Auf den Vergleichstest (siehe Kapitel 6.3.2) sind die beiden Klassen von den zwei beteiligten Informatik-Lehrern leider sehr unterschiedlich vorbereitet worden und es ergeben sich dadurch (teilweise signifikante) Ergebnisunterschiede. Gerade bei schweren Themen (für die OOP-Later-Klasse sind dies die Themen „Arrays und Strings“, „Methoden“ und „Assoziation“) ließen sich durch die spezielle Präparation einer Klasse deutlich bessere Ergebnisse erreichen. Bei leichteren Themen machte sich dies nicht so stark bemerkbar.

Beim Nachhaltigkeitstest (siehe Kapitel 6.3.3) sind alle (signifikanten) Unterschiede des Vergleichstests verschwunden. Es wurden im Nachhaltigkeitstest keine signifikanten Unterschiede zwischen dem OOP-First-Vorgehen und dem OOP-Later-Vorgehen im Hinblick auf den Lernerfolg festgestellt bzw. gemessen. Sowohl im Gesamtergebnis als auch in den einzelnen Themenresultaten ergaben sich keine signifikanten Unterschiede. Beide Vorgehensweisen produzierten (spätestens zwei Monate nach dem Ende des Informatik-Anfangsunterrichts) vergleichbare Lernerfolge im Hinblick auf den verwendeten Test.

Betrachtet man den Lernverlauf mit Hilfe aller drei Tests (siehe Kapitel 6.3.4), ergeben sich fast identische Lern- bzw. Wissenskurven (bezogen auf die ersten acht Themen). Ausgehend von ca. 10% Vorkenntnissen erreichen die Schülerinnen und Schüler nach einem Schuljahr ca. 64,6% Kenntnisse der unterrichteten Informatik-Inhalte. Die Nachhaltigkeit des Wissens ist relativ groß. Der Kenntnisstand sinkt innerhalb von zwei Monaten nur um durchschnittlich 5,7% auf ca. 58,9%.

Das letzte Kapitel 6.3.5 widmete sich dann speziellen Lerngruppen, zum einen den (sehr) leistungsstarken bzw. (sehr) leistungsschwachen Schülern, zum anderen den Schülerinnen.

Es werden bei der Lerngruppe der leistungsstärksten Schüler keine signifikanten Unterschiede im Lernerfolg in Abhängigkeit von der fachdidaktischen Vorgehensweise gemessen. Eine spezielle Präparierung leistungsstarker Schüler auf einen Test hin zeigt keine offensichtliche Auswirkung.

Es gibt bei der Lerngruppe der leistungsschwächsten Schüler (teilweise signifikante) Unterschiede in den Ergebnissen des Vergleichstests im Verhältnis zu denen des Nachhaltigkeitstests. Eine spezielle Präparierung leistungsschwacher Schüler auf einen Test hin zeigt eine offensichtliche Auswirkung, die aber nicht nachhaltig ist.

Es lassen sich bei der Lerngruppe der Schülerinnen aufgrund der geringen Anzahl an Teilnehmerinnen so gut wie keine Aussagen treffen. Erfreulich ist aber, dass nur eine von vier Schülerinnen der Gruppe der leistungsschwachen Schüler angehört und dass die Schülerinnen bessere Ergebnisse in der Informatik als in der Mathematik erzielen.

Damit ist der Lernerfolg der Schülerinnen und Schüler (in der Hauptstudie) nach einem Schuljahr in Abhängigkeit vom fachdidaktischen Vorgehen dokumentiert und analysiert und die Forschungsfrage zum objektiven Aspekt (vgl. Kapitel 2.4.1) könnte beantwortet werden. Dies geschieht aber erst in Kapitel 7 (speziell Kapitel 7.3, Kapitel 7.5.1 und Kapitel 7.5.2). Vorher sollen die Ergebnisse zum subjektiven Aspekt betrachtet werden.

## **6.4 Hauptstudie: Das emotionale, kognitive und motivationale Erleben**

Im Folgenden sollen die Resultate der Hauptstudie aus dem Schuljahr 2007/2008 im Hinblick auf das subjektive Erleben des Informatik-Anfangsunterrichts ausführlich dargestellt werden. Die Ergebnisse der Befragung der Schüler im Hinblick auf das subjektive Erleben des Unterrichts werden zuerst mit Hilfe einer Matrix (vgl. Kapitel 5.1.3 und Abbildung 87) im Hinblick auf die drei Erlebnis-Dimensionen und die vier Frage-Bereiche im Überblick dargestellt (siehe Kapitel 6.4.1). Danach werden die Ergebnisse aufgeschlüsselt in die emotionale Dimension (siehe Kapitel 6.4.2), die kognitive Dimension (siehe Kapitel 6.4.3) und die motivationale Dimension (siehe Kapitel 6.4.4) und unter speziellen Gesichtspunkten erörtert und interpretiert. Dann wird der zeitliche Verlauf des subjektiven Erlebens dargestellt (siehe 6.4.5). Zum Schluss werden noch spezielle Lerngruppen von leistungsstarken bzw. leistungsschwachen Schülern und von Schülerinnen untersucht (siehe Kapitel 6.4.6).

Die ersten fünf Unterkapitel sind überwiegend aus einer Veröffentlichung aus dem Jahr 2010 entnommen, dem Papier [ES10a] zur ITiCSE-Konferenz („Conference on Innovation and Technology in Computer Science Education“) in Bilkent (Ankara) in der Türkei.

### **6.4.1 Ergebnis-Überblick**

Zuerst soll noch einmal in Erinnerung gerufen werden, inwiefern die 25 generierten Fragen sich auf die drei Erlebnis-Dimensionen und die vier Frage-Bereiche beziehen. Dafür soll die Abbildung 87 als Vorlage für die

Abbildung 114 dienen. Die einzigen Unterschiede in den Abbildungen sind, dass ein Hinweis darauf gegeben wird, dass beim Ergebnis-Überblick die Frage 1 (Frage nach dem privaten Umfeld) und die Frage 6 nicht berücksichtigt werden und dass die Fragen 2/3, 8/9 und 15/20 gleichartige Fragen (bezogen auf den Wohlfühl- und den Stressfaktor) sind. Der Schwerpunkt der Fragen liegt auf der emotionalen Dimension bzw. auf dem jeweils letzten unterrichteten Thema.

	Emotionale Dimension	Kognitive Dimension	Motivationale Dimension
Fragen zur Schule (bzw. zum privaten Umfeld)	(1),2,3	5	4
Fragen zum Fach	(6),8,9	11	10
Fragen zum jeweiligen Lehrer	7,18,19	17	
Fragen zum letzten Thema	15,20	12,14,21,22, 23,24,25	13,16

Abbildung 114: Fragen im Kontext zu den Erlebnis-Dimensionen und den Frage-Bereichen (vgl. [ES10a, Figure 3, S. 109] und Abbildung 87)

Bei dem Ergebnis-Überblick (siehe Abbildung 115) sollen jetzt für jedes Feld der Matrix (also bezogen auf eine spezielle Erlebnis-Dimension und einen speziellen Fragebereich) der Durchschnittswert der Fragen angegeben werden. Nur die Frage 1 wird nicht berücksichtigt, da sie ein späterer Indikator für die „Gleichartigkeit des subjektiven Erlebens“ beider Klassen ist (siehe Kapitel 7.4.6). Den fünf Ankreuzmöglichkeiten zu jeder der 25 Fragen werden die Zahlen von 1 bis 5 zugeordnet, so dass Zahlen unter dem Durchschnittswert 3,0 ein positives subjektives Erleben anzeigen, Werte über 3,0 in einem negativen subjektiven Erlebnisbereich liegen.

	Emotionale Dimension		Kognitive Dimension		Motivationale Dimension		Durchschnitt*** Subjektives Erleben	
	OOP-First	OOP-Later	OOP-First	OOP-Later	OOP-First	OOP-Later	OOP-First	OOP-Later
Fragen zur Schule	<b>2,9*</b> p=0,0007	<b>2,5*</b> d=2,51	<b>3,1*</b> p=0,0496	<b>3,0*</b> d=1,01	<b>2,6*</b> p=0,0057	<b>2,3*</b> d=1,52	2,87 (2,86)**	2,60 (2,60)**
Fragen zum Fach Informatik	<b>2,6*</b> p=0,012	<b>2,3*</b> d=1,36	3,1 p=0,15	3,0 entfällt	<b>2,3*</b> p=0,0002	<b>1,9*</b> d=2,34	2,67 (2,56)**	2,40 (2,31)**
Fragen zum jeweiligen Lehrer	<b>2,2*</b> p=0,0016	<b>1,9*</b> d=1,90	<b>2,4*</b> p=0,0025	<b>2,0*</b> d=1,69			2,30 (2,31)**	1,95 (1,94)**
Fragen zum letzten Thema	2,7 p=0,08	2,5 entfällt	<b>2,6*</b> p=0,016	<b>2,4*</b> d=1,26	<b>2,0*</b> p=0,0155	<b>1,8*</b> d=1,30	2,43 (2,40)**	2,23 (2,22)**
Durchschnitt*** Erlebnis-Dimension	2,6	2,3	2,8	2,6	2,3	2,0	2,57 (2,53)**	2,30 (2,27)**

Abbildung 115: Subjektives Erleben im Kontext zu den Erlebnis-Dimensionen und den Frage-Bereichen

(ohne Fragen 1 und 6, vgl. [ES10a, Figure 4, S. 109], p  $\triangleq$  Irrtumswahrscheinlichkeit, d  $\triangleq$  Effektstärke, \* signifikante Ergebnis-Unterschiede sind fett gedruckt, \*\* tatsächlicher Durchschnitt (siehe z. B. Kapitel 6.4.5), \*\*\* ohne Signifikanzhinweis)

Folgende Aussagen können getroffen werden [ES10a, S. 109]:

- 18 von 22 Werten liegen unter dem Mittelwert von 3,0, spiegeln also positives subjektives Erleben wider, zwei Werte treffen genau den Durchschnitt und nur zwei Werte spiegeln geringfügig negatives subjektives Erleben wider.
- Der Durchschnitt der elf Werte der OOP-First-Klasse ( $\bar{X} = 2,59$ ) ist höher (also schlechter) als der Durchschnitt der elf Werte der OOP-Later-Klasse ( $\bar{X} = 2,33$ ).
- Alle Werte der OOP-First-Klasse sind höher (also schlechter) als die entsprechenden Werte der OOP-Later-Klasse. Neunmal sind die Werte sogar signifikant unterschiedlich.

Die Interpretation der Ergebnisse ist trotzdem schwierig, weil die OOP-First-Klasse sowohl höhere Werte bei den Frage-Bereichen „Schule“ und „Lehrer“ als auch bei der ausgeklammerten Frage 1 zum „privaten Umfeld“ hat (siehe Kapitel 7.4.6: Erleben die OOP-Later-Schüler den Unterricht wirklich subjektiv angenehmer?). Evtl. neigt die OOP-First-Klasse generell dazu etwas höhere Werte anzukreuzen.

In den folgenden drei Kapiteln sollen einzelne Details zu den einzelnen Erlebnis-Dimensionen erläutert werden.

## 6.4.2 Emotionale Dimension

Im Folgenden sollen einzelne Ergebnisse zu den Fragen zur emotionalen Dimension in Abhängigkeit von den drei Frage-Bereichen „Schule“, „Fach Informatik“ und „letztes Thema“ (siehe Abbildung 116) näher dargestellt und diskutiert werden. Die Antworten zum Fragebereich „Lehrer“ werden erst im Kapitel 7.2.1 untersucht.

	Fragen zur emotionalen Dimension	OOP-First	OOP-Later	Durchschnitt
Fragen zur Schule	(1),2,3	<b>2,9*</b> p=0,00007	<b>2,5*</b> d=2,51	2,70
Fragen zum Fach Informatik	(6),8,9	<b>2,6*</b> p=0,012	<b>2,3*</b> d=1,36	2,45
Fragen zum jeweiligen Lehrer	7,18,19	<b>2,2*</b> p=0,0016	<b>1,9*</b> d=1,90	2,05
Fragen zum letzten Thema	15,20	2,7 p=0,08	2,5 entfällt	2,60
	<b>Durchschnitt</b>	2,60	2,30	2,45

Abbildung 116: Subjektives Erleben im Kontext zur emotionalen Dimension und zu den Frage-Bereichen

(\* signifikante Ergebnis-Unterschiede sind fett gedruckt)

Das emotionale Erleben ist somit der Durchschnitt aus dem so genannten Wohlfühlfaktor und dem so genannten Stressfaktor. Die Fragen in den einzelnen Bereichen zur Ermittlung dieser Faktoren waren wie folgt formuliert (vgl. Kapitel 5.2.1):

Frage 2: Ich fühle mich zur Zeit in der **Schule** eher ...  
(sehr wohl/ sehr unwohl)

- Frage 3: Mein Stress in der **Schule** ist zur Zeit eher ...  
(sehr niedrig / sehr hoch)
- Frage 8: Ich fühle mich allgemein beim Unterricht im **Fach Informatik** eher ...  
(sehr wohl / sehr unwohl)
- Frage 9: Mein Stress im **Fach Informatik** ist eher ...  
(sehr niedrig / sehr hoch)
- Frage 15: Ich fühlte mich speziell beim **letzten Thema** eher ...  
(sehr wohl / sehr unwohl)
- Frage 20: Mein Stress beim **letzten Thema** war eher ...  
(sehr niedrig / sehr hoch)

In den nächsten Abschnitten sollen die folgenden Sachverhalte dargestellt und erläutert werden [ES10a, S. 109-110]:

- Zeitlicher Verlauf des emotionalen Erlebens
- Emotionales Erleben in Abhängigkeit von den Themen
- Rangfolge des emotionalen Erlebens in Bezug auf die Themen
- Direkte Einschätzung des Fachs Informatik (Frage 6)

Die Ergebnisse zur Frage 1 werden erst in Kapitel 7.4.6 dargestellt.

#### a) Zeitlicher Verlauf des emotionalen Erlebens

Zuerst sollen die Antworten zu den Fragen der emotionalen Dimension in Abhängigkeit von der Zeit und von den drei Bereichen „Schule“, „Fach Informatik“ und „letztes Thema“ dargestellt werden (OOP-First: siehe Abbildung 117, OOP-Later: siehe Abbildung 118).

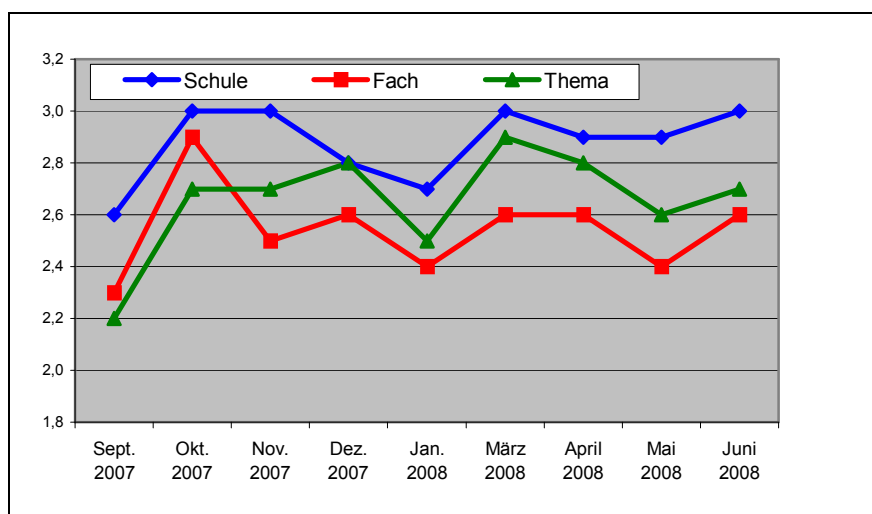


Abbildung 117: Das emotionale Erleben der OOP-First-Klasse [ES10a, Figure 5, S. 110]



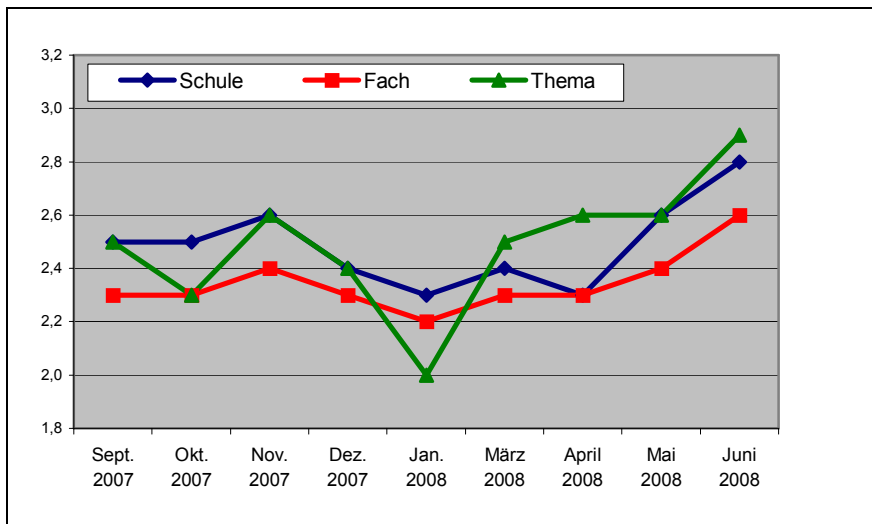


Abbildung 118: Das emotionale Erleben der OOP-Later-Klasse [ES10a, Figure 6, S. 110]

Folgende Aussagen können getroffen werden [ES10a, S. 110]:

- Das emotionale Erleben bzw. das emotionale Klima in der OOP-First-Klasse ist nicht so gut wie in der OOP-Later-Klasse. Da aber der positive subjektive Bereich unter dem Durchschnitt von dem Wert 3 liegt, haben beide Klassen ein gutes emotionales Erleben in allen drei Bereichen. Bei der Interpretation der Unterschiede im Erleben des Fachs und der Themen im Hinblick auf die fachdidaktische Vorgehensweise zwischen beiden Klassen muss man sehr vorsichtig sein, da auch im Frage-Bereich „Schule“ die gleichen Unterschiede vorhanden sind.
- In der OOP-First-Klasse ist das Erleben der Informatik-Themen und des Faches Informatik immer besser als das emotionale Erleben der Schule insgesamt. Bei sieben von neun Messpunkten wird das Fach Informatik besser erlebt als das gerade unterrichtete Thema.
- In der OOP-Later-Klasse kann man nicht stark differenzieren zwischen den Werten des emotionalen Erleben der Schule und der einzelnen Themen. Aber auffällig ist, dass alle Werte des Fachs Informatik unter (bzw. gleich) den Werten der beiden anderen Bereiche liegen: Das Fach Informatik stresst am wenigstens und die Schüler fühlen am wohlsten.
- Am Ende des Schuljahres gehen vom vorletzten Messpunkt (Mai 2008) zum letzten Messpunkt (Juni 2008) alle Werte für alle drei Bereiche für die beiden Klassen nach oben: die Schülerinnen und Schüler fühlen sich mehr überarbeitet (gestresst) und weniger wohl. In der OOP-Later-Klasse sind die letzten Messwerte auch jeweils die höchsten je gemessenen Werte in allen drei Bereichen im gesamten Schuljahr: Das emotionale Klima verschlechtert sich auffällig zum Ende des Schuljahres.
- In beiden Klassen gibt es genau in der Mitte der Befragung (Januar 2008) eine Senke in allen drei Kurven: das emotionale Klima hat einen kurzfristigen Höhepunkt, die Schülerinnen und Schüler fühlen sich wahrscheinlich durch die Weihnachtsferien weniger gestresst und insgesamt wohler.
- Interessanterweise startete die OOP-Later-Klasse im Januar mit den objektorientierten Themen. Die „Einführung in die OOP“ mit dem BlueJ-Tool erreicht mit Abstand den besten emotionalen Wert für alle Themen. Aber alle nachfolgenden Themen, welches auch alles OO-Themen sind, bewegen sich über den

emotionalen Werten der Bereiche „Fach“ und „Schule“: Diese Themen stressen die Schüler mehr als das „Fach“ oder die „Schule“ insgesamt.

- Im Vergleich dazu haben zwar in der OOP-First-Klasse nach der „Einführung in die OOP“ mit dem BlueJ-Tool (mit dem besten emotionalen Wert) alle folgenden Themen schlechtere Werte, dieses schlechtere emotionale Befinden ist aber relativ konstant im Unterschied zur OOP-Later-Klasse.

b) Emotionales Erleben in Abhängigkeit von den Themen

In diesem Abschnitt sollen nur die Antworten zum Fragebereich „Thema“ analysiert werden, d. h. der Mittelwert auf die Antworten zur Frage 15 und zur Frage 20: Wohlfühl- und Stressfaktor bezogen auf das letzte Thema.

In Abbildung 119 sind die Ergebnisse zum Bereich „Thema“ dieselben wie in Abbildung 117 bzw. Abbildung 118. Jetzt sind sie aber nicht nach der Zeit geordnet, sondern nach den Themen.

Folgende Aussagen können getroffen werden [ES10a, S. 110]:

- Bis auf zwei Ausnahme (Thema 9: Assoziation und Thema 4: Methoden) ist die OOP-Later-Klasse weniger gestresst und fühlt sich wohler als die OOP-First-Klasse.
- Das Thema „Assoziation“ hat bei der OOP-Later-Klasse den mit Abstand schlechtesten Wert.
- Bei der OOP-First-Klasse hingegen gibt es gleich drei „emotional schlechte“ Themen: Thema 5 (Steuerstruktur Selektion), Thema 6 (Steuerstruktur Iteration) und Thema 7 (Arrays und Strings), alles Themen (bis auf Strings, die aber nur 30% des Themas 7 einnehmen, vgl. Kapitel 7.3.3), die prozedurale Themen sind.

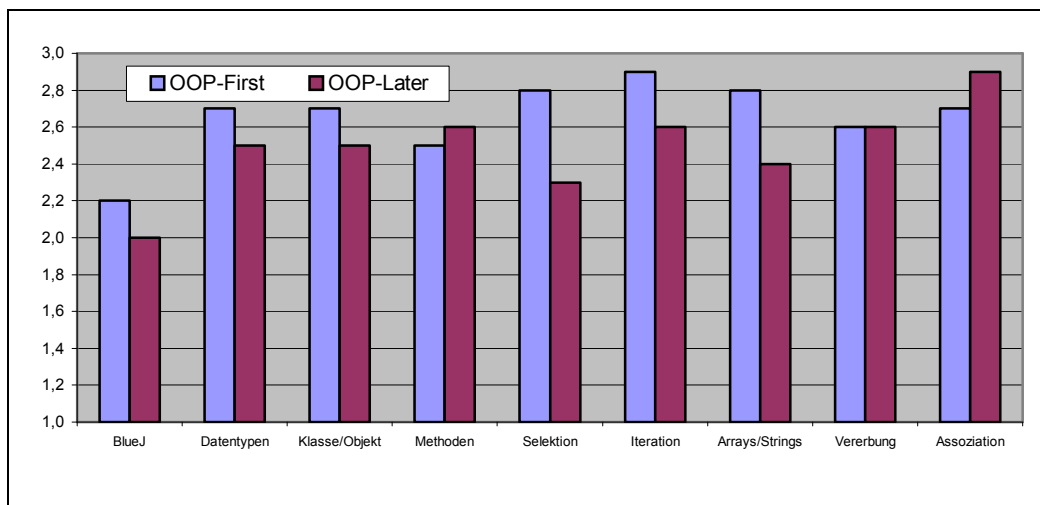


Abbildung 119: Emotionales Erleben in Abhängigkeit von den Themen [ES10a, Figure 7, S. 110]

(vgl. kognitives Erleben in Abbildung 126 und motivationales Erleben in Abbildung 137)

- Aber auch diese drei Themen sind letztlich im Bereich des positiven emotionalen Klimas: Alle Werte liegen unter dem Mittelwert 3. Es werden also Unterschiede im guten emotionalen Klima-Bereich diskutiert.
- Bei beiden Klassen ist das Thema 1 („Einführung in die OOP mit BlueJ“) das „emotional beste“ Thema. Die Anwendung des BlueJ-Tools scheint sehr günstig im Kontext zur emotionalen Dimension zu sein.
- Obwohl dieser Fragebereich der einzige der vier Frage-Bereiche im Hinblick auf die emotionale Dimension ist, bei dem die Mittelwerts-Unterschiede (OOP-First: 2,7 / OOP-Later: 2,5) nicht signifikant sind, gibt es

bei drei Themen auffällige Ergebnis-Unterschiede: Thema 5, Thema 6 und Thema 7. Diese sind aber nicht signifikant (bei einem Signifikanzniveau von 95%), z. B. ist beim Thema 5 (OOP-First: 2,76 / OOP-Later: 2,34) der Wert für p gleich 0,11: Die Irrtumswahrscheinlichkeit ist mit 11% also zu hoch. Zwei Erklärungen gibt es für das „Wahrnehmen eines auffälligen Unterschiedes“. Zum einen sind die Werte in der Darstellung gerundet (OOP-First: 2,8 / OOP-Later: 2,3) und vergrößern damit die augenfällige Differenz. Zum anderen liegt es am gewählten Werte-Bereich. Statt der Gesamtskala von 1 bis 5 ist eine Teilskala von 1,0 bis 3,0 gewählt worden, so dass die Unterschiede optisch größer erscheinen.

c) Rangfolge des emotionalen Erlebens in Bezug auf die Themen

Im Folgenden sollen das emotionale Erleben („eE“) pro Thema in einer Rangfolge dargestellt werden (siehe Abbildung 120), damit einige Aussagen aus dem vorherigen Abschnitt besser visualisiert werden bzw. weitere Aussagen getroffen werden können. Wenn man die Extremgruppen-Definition (jeweils 25% der Untersuchungsteilnehmer) von Bortz und Döring [BD02, S. 219] zur Grundlage einer Entscheidung für eine Extremwert-Definition nimmt, können die zwei größten und die zwei kleinsten Werte ( $25\% * 9 = 2,25$ ) fokussiert werden.

Nr.	OOP-First	„eE“	Nr.	OOP-Later	„eE“
1	Einführung in die OOP	2,2	1	Einführung in die OOP	2,0
4	Methoden	2,5	5	Steuerstruktur Selektion	2,3
8	Vererbung	2,6	7	Arrays und Strings	2,4
9	Assoziation	2,7	3	Klasse und Objekt	2,5
3	Klasse und Objekt	2,7	2	Datentypen / Steuerstruktur Sequenz	2,5
2	Datentypen / Steuerstruktur Sequenz	2,7	8	Vererbung	2,6
7	Arrays und Strings	2,8	6	Steuerstruktur Iteration	2,6
5	Steuerstruktur Selektion	2,8	4	Methoden	2,6
6	Steuerstruktur Iteration	2,9	9	Assoziation	2,9
	<b>Durchschnitt</b>	<b>2,66</b>		<b>Durchschnitt</b>	<b>2,49</b>

Abbildung 120: Rangfolge des „emotionalen Erlebens“ (Wohlfühl- und Stressfaktor) der Themen

Folgende Aussagen können getroffen werden:

- Beide Klassen fühlen bei der „Einführung in die OOP“ mit dem BlueJ-Tool die positivsten Emotionen.
- Die OOP-First-Klasse hat interessanterweise die schlechtesten emotionalen Werte in den so genannten prozeduralen Themen (Teil-Thema „Strings“ ist Ausnahme).
- Es gibt sehr unterschiedliche Gefühle der beiden Klassen im Hinblick auf zwei Themen: das Thema „Methoden“ steht in der OOP-First-Klasse auf einem positiven Platz, in der OOP-Later-Klasse aber auf einem negativen Platz, das Thema „Steuerstruktur Selektion“ ist dagegen in der OOP-First-Klasse auf einem negativen Platz, währenddessen es in der OOP-Later-Klasse auf einem positiven Platz steht.
- Einen Zusammenhang zwischen den vier „emotional negativsten“ Themen der OOP-Later-Klasse zu finden fällt schwer. Es sei nur erwähnt, dass zwei dieser Themen die letzten beiden Themen im Schuljahr waren.
- Alle Werte beider Klassen liegen unter dem Mittelwert 3, also im positiven emotionalen Bereich.

d) Direkte Einschätzung des Fachs Informatik (Frage 6)

Bisher war die Frage 6 ausgeklammert worden, damit die emotionale Befindlichkeit mit denselben zwei Fragen in den drei Frage-Bereichen verglichen werden kann. Die beiden bisherigen Fragen zum Fach Informatik (Frage 8 und Frage 9) waren Fragen zum Wohlfühlfaktor und zum Stressfaktor. Mit ihnen wurde indirekt auf die emotionale Befindlichkeit in dem Fach Informatik geschlossen. Die Frage 6 dagegen fragt sehr direkt die Schülerinnen und Schüler nach dem Fach:

Frage 6: Das **Fach Informatik** (innerhalb der Technik) finde ich ...  
(sehr gut / sehr schlecht)

Die Abbildung 119 und die Abbildung 122 zeigen sowohl den bisher ermittelten Verlauf der emotionalen Befindlichkeit (siehe Abbildung 117 und Abbildung 118) als auch den Verlauf der Werte für die Antworten zu der Frage 6.

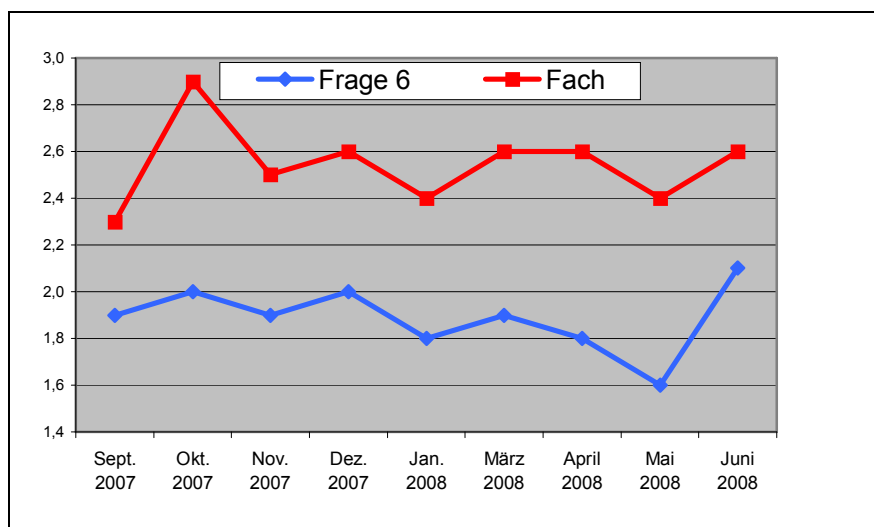


Abbildung 121: Direkte Frage nach dem Fach (Frage 6) und Stress- und Wohlfühlfaktor bei OOP-First

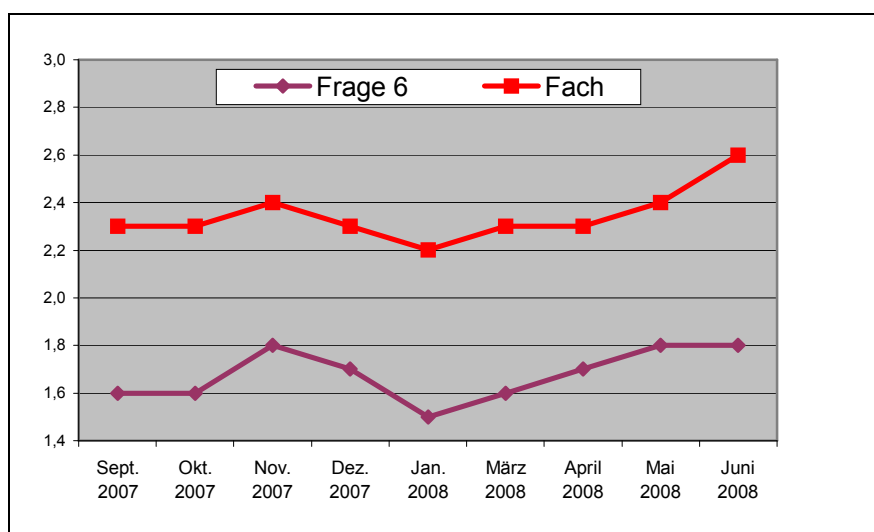


Abbildung 122: Direkte Frage nach dem Fach (Frage 6) und Stress- und Wohlfühlfaktor bei OOP-Later

Folgende Aussagen können getroffen werden:

- Die Schülerinnen und Schüler beider Klassen (OOP-First: 1,9 / OOP-Later: 1,7) finden das Fach Informatik gut. Die OOP-Later-Klasse zeigt wie so oft dabei die (noch) besseren Werte.
- Alle Werte der Antworten zur Frage 6 sind deutlich kleiner als die Mittelwerte zum Wohlfühl- und Stressfaktor.
- Die beiden Kurven der OOP-First-Klasse korrelieren mit einem Korrelationskoeffizienten von 0,516.
- Die beiden Kurven der OOP-Later-Klasse besitzen eine hohe Korrelation (Korrelationskoeffizient = 0,798).
- Die Korrelationen zeigen, dass das Item 6 gut die beiden anderen Items zur Erfassung der emotionalen Befindlichkeit zum Frage-Bereich „Thema“ ergänzt. Es bestätigt beziehungsweise verstärkt eine Aussage, die im Kontext zu der Abbildung 117 und der Abbildung 118 getroffen wurde: Das Fach Informatik stresst am wenigstens die Schüler. Sie fühlen sich am wohlsten von allen drei Frage-Bereichen und finden das Fach mehr als gut.

Mit diesem positiven Ergebnis zum Schulfach Informatik ist das Kapitel zur emotionalen Dimension abgeschlossen. Jetzt fehlen nur noch die Ergebnisse zur kognitiven und zur motivationalen Dimension.

### 6.4.3 Kognitive Dimension

Im Folgenden sollen einzelne Ergebnisse zu den Fragen zur kognitiven Dimension in Abhängigkeit von den drei Frage-Bereichen „Schule“, „Fach Informatik“ und „letztes Thema“ (siehe Abbildung 123) näher dargestellt und diskutiert werden. Die Antworten zum Fragebereich „Lehrer“ (Frage 17) werden erst im Kapitel 7.2.1 untersucht.

	Fragen zur kognitiven Dimension	OOP-First	OOP-Later	Durchschnitt
Fragen zur Schule	5	<b>3,1*</b> p=0,0496	<b>3,0*</b> d=1,01	3,05
Fragen zum Fach Informatik	11	3,1 p=0,15	3,0 entfällt	3,05
Fragen zum jeweiligen Lehrer	17	<b>2,4*</b> p=0,0025	<b>2,0*</b> d=1,69	2,20
Fragen zum letzten Thema	12,14,21,22, 23,24,25	<b>2,6*</b> p=0,016	<b>2,4*</b> d=1,26	2,50
	<b>Durchschnitt</b>	2,80	2,60	2,70

Abbildung 123: Subjektives Erleben im Kontext zur kognitiven Dimension und zu den Frage-Bereichen

(\* signifikante Ergebnis-Unterschiede sind fett gedruckt)

Das kognitive Erleben wird in der Regel nur durch ein Item ermittelt. Einzige Ausnahme ist der Schwerpunkt-Frage-Bereich „Thema“, indem aber die Frage nach der Schwierigkeit (der so genannte Schwierigkeits- bzw. Schwerefaktor) im Mittelpunkt der folgenden Analysen stehen soll.

Die Fragen zur Ermittlung dieser Faktoren waren wie folgt formuliert (vgl. Kapitel 5.2.2):

- Frage 5: Ich empfinde die Anforderungen der **Schule** an mich zur Zeit eher als ...  
(sehr niedrig / sehr hoch)
- Frage 11: Ich empfinde die Anforderungen des **Faches Informatik** an mich eher als ...  
(sehr niedrig / sehr hoch)
- Frage 12: Den Stoff des **letzten Themas** empfand ich als ...  
(sehr leicht / sehr schwer)
- Frage 14: Den Stoffumfang des **letzten Themas** empfand ich als ...  
(sehr niedrig / sehr hoch)
- Frage 21: Ich beherrsche jetzt das **letzte Thema** ...  
(sehr gut / sehr schlecht)
- Frage 22: Mein Wissenszugewinn beim **letzten Thema** war ...  
(sehr hoch / sehr niedrig)
- Frage 23: Mein Arbeitsaufwand beim **letzten Thema** war ...  
(sehr gering / sehr hoch)
- Frage 24: Der Wiedererkennungswert zu bereits behandelten **Themen** war ...  
(sehr hoch / sehr niedrig)
- Frage 25: Die Nützlichkeit des Wissens bereits behandelter Themen für das **letzte Thema** war ...  
(sehr hoch / sehr niedrig)

In den folgenden Abschnitten soll das kognitive Erleben (speziell der Anforderungs- bzw. Schwierigkeitsfaktor) im zeitlichen Kontext und im thematischen Kontext näher analysiert werden:

- Zeitlicher Verlauf des kognitiven Erlebens
- Kognitives Erleben in Abhängigkeit von den Themen
- Bestandteile des kognitiven Erlebens
- Empfundene Schwere bzw. Schwierigkeit in Abhängigkeit von den Themen
- Das Schwere-Empfinden im Kontext zu der zeitlichen Abfolge der Unterrichtsthemen

#### a) Zeitlicher Verlauf des kognitiven Erlebens

Die folgenden zwei Abbildungen stellen die kognitive Dimension in Abhängigkeit von den drei Frage-Bereichen dar. Für die Frage-Bereiche „Schule“ und „Fach“ sind die Kurven nur das Ergebnis eines Items, dem so genannten Anforderungsfaktor. Der kognitive Wert für den Frage-Bereich „letztes Thema“ wird dagegen aus sieben Items gebildet. Ein Item (der so genannte Schwierigkeitsfaktor) dieser Itemgruppe ist extra noch einmal abgebildet, zum einen, weil es dem Anforderungsfaktor nahe kommt, zum anderen weil es das Item ist, was in den nächsten Abschnitten am meisten fokussiert wird.

Als erstes soll das kognitive Erleben der OOP-First-Klasse dargestellt werden (siehe Abbildung 124).

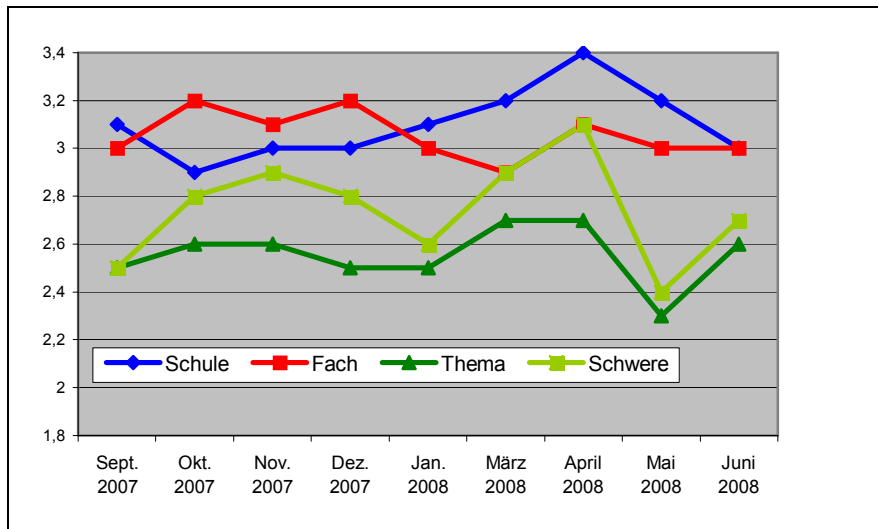


Abbildung 124: Das kognitive Erleben der OOP-First-Klasse

Folgende Aussagen können getroffen werden:

- Die Schülerinnen und Schüler erleben die Anforderungen, die die Schule und das Fach Informatik stellen, durchschnittlich als gleich hoch (Anforderungsfaktor:  $\bar{X} = 3,1$ ). Der Wert trifft fast den Durchschnitt der Skala von 1 bis 5, also den Wert 3. Dies ist für die Schule und das Fach erfreulich: Die Schüler waren durchschnittlich weder über- noch unterfordert.
- Alle Werte des kognitiven Erlebens zum Frage-Bereich „Thema“ sind niedriger (also positiver) als die Werte zur Schule und zum Fach.
- Der Schwerefaktor der einzelnen Themen liegt nie über dem Anforderungsfaktor des Faches, die Themen werden leichter erlebt als die Anforderungen des Faches insgesamt.
- Das subjektive Schwere-Empfinden des letzten Themas ( $\bar{X} = 2,7$ ) liegt höher als das kognitive Erleben zum letzten Thema ( $\bar{X} = 2,6$ ) insgesamt. Aber es besteht eine hohe Korrelation (Korrelationskoeffizient: 0,87) zwischen den beiden Messreihen.

Als zweites soll das kognitive Erleben der OOP-Later-Klasse dargestellt werden (siehe Abbildung 125).

Folgende Aussagen können getroffen werden:

- Die Schülerinnen und Schüler erleben die Anforderungen, die die Schule und das Fach Informatik stellen, durchschnittlich als gleich hoch (Anforderungsfaktor:  $\bar{X} = 3,0$ ). Der Wert trifft genau den Durchschnitt der Skala von 1 bis 5, also den Wert 3. Er liegt nur um den Faktor 0,1 besser als der vergleichbare Wert der OOP-First-Klasse. Dieser Unterschied ist für die Schule ganz knapp signifikant ( $p = 0,0496$ ) und für das Fach Informatik nicht signifikant ( $p = 0,15$ ).
- Alle Werte des gesamten kognitiven Erlebens zum Frage-Bereich „Thema“ sind genauso wie in der OOP-First-Klasse niedriger (also positiver) als die Werte zur Schule und zum Fach.

- Der Schwerefaktor der einzelnen Themen liegt bei den ersten sieben Themen immer deutlich unter dem Anforderungsfaktor des Faches, bei den letzten beiden Themen („Vererbung“ und „Assoziation“) werden die Themen aber ähnlich schwer erlebt wie die Anforderungen des Faches.

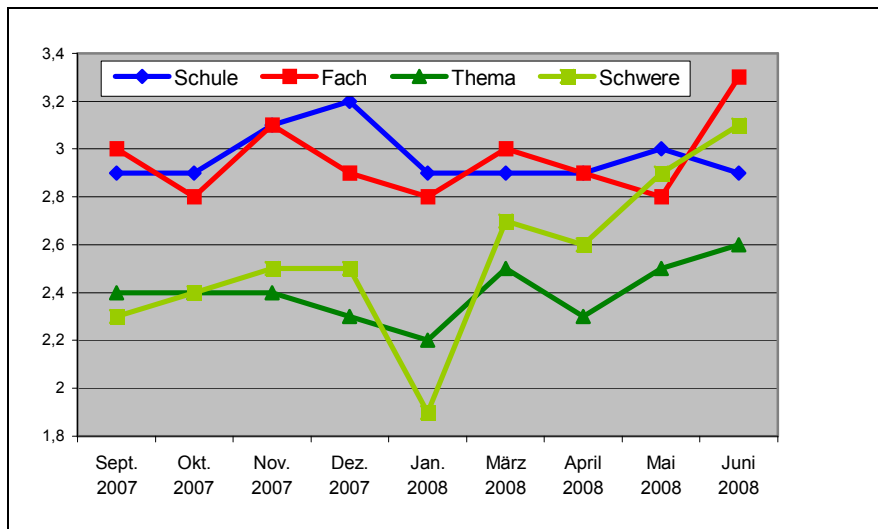


Abbildung 125: Das kognitive Erleben der OOP-Later-Klasse

- Das subjektive Schwere-Empfinden des letzten Themas ( $\bar{X} = 2,5$ ) und das gesamte kognitive Erleben zum letzten Thema ( $\bar{X} = 2,4$ ) sind im Gegensatz zur OOP-First-Klasse ähnlich. Es besteht eine hohe Korrelation (Korrelationskoeffizient: 0,85) zwischen den beiden Messreihen.

#### b) Kognitives Erleben in Abhängigkeit von den Themen

In diesem Abschnitt sollen nur die Antworten zum Fragebereich „Thema“ analysiert werden, d. h. der Mittelwert zu den sieben Items 12, 14, 21, 22, 23, 24 und 25 bezogen auf das letzte Thema.

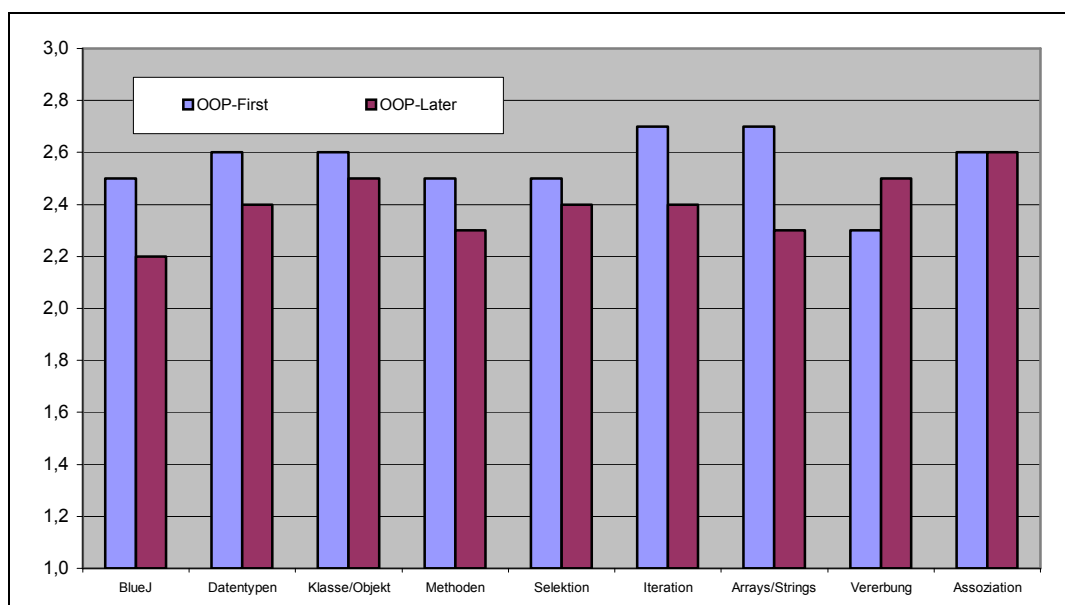


Abbildung 126: Kognitives Erleben (alle sieben Items) in Abhängigkeit von den Themen [ES10a, Figure 9, S. 111]

(vgl. emotionales Erleben in Abbildung 119 und motivationales Erleben in Abbildung 137)



In Abbildung 126 sind die Ergebnisse zum Bereich „Thema“ dieselben wie in Abbildung 124 bzw. Abbildung 125. Jetzt sind sie aber nicht nach der Zeit geordnet, sondern nach den Themen.

Folgende Aussagen können getroffen werden [ES10a, S. 111]:

- Die Differenz zwischen den Mittelwerten der beiden Klassen (OOP-First: 2,6 / OOP-Later: 2,4) ist signifikant ( $p = 0,016$ ). Die Irrtumswahrscheinlichkeit beträgt nur 1,6%. Die Effektstärke ( $d = 1,26$ ) liegt über dem Wert 0,8, der Mittelwertsunterschied stellt somit einen starken Effekt dar (vgl. [Coh88, S. 20-27]).
- Bei sieben von neun Themen ist das kognitive Erleben der OOP-Later-Klasse besser als das der OOP-First-Klasse.
- Nur ein Thema wird von der OOP-First-Klasse kognitiv besser als von der OOP-Later-Klasse erlebt.
- Thema 6 („Steuerstruktur Iteration“) und Thema 7 („Arrays und Strings“) schneiden bei der OOP-First-Klasse nicht ganz so gut ab, das beste Thema ist das Thema 8 („Vererbung“).
- Thema 9 („Assoziation“), Thema 8 („Vererbung“) und Thema 3 („Klasse und Objekt“) schneiden bei der OOP-Later-Klasse nicht ganz so gut ab, das beste Thema ist das Thema 1 („OOP-Einführung mit BlueJ“).
- Die größte Differenz ergibt sich bei Thema 7 („Arrays und Strings“), sie ist signifikant ( $p = 0,01$ ). Die Irrtumswahrscheinlichkeit beträgt nur 1%. Die Effektstärke ( $d = 0,85$ ) liegt über dem Wert 0,8, der Mittelwertsunterschied stellt somit einen starken Effekt dar (vgl. [Coh88, S. 20-27]).

#### c) Bestandteile des kognitiven Erlebens

Im Folgenden sollen die sieben Items des kognitiven Erlebens in drei Bereiche eingeteilt werden (siehe Abbildung 127 und Abbildung 128):

- Erbrachte *Leistung* der Schülerinnen und Schüler in Form von der empfundenen Themenschwere (Frage 12), dem Stoffumfang (Frage 14) und dem nötigen Arbeitsaufwand (Frage 23). Große Leistungswerte bedeuten, dass die Schüler ein Thema schwer empfunden haben bzw. der Stoffumfang des Themas und der nötige Arbeitsaufwand zum Durchdringen des Themas hoch waren. Erstrebenswert sind m. E. aus Schulsicht eher Werte, die im mittleren Bereich liegen, also nahe dem Wert 3,0.
- *Gewinn* in Form von Wissenszugewinn (Frage 22) durch das letzte Thema und Beherrschung des letzten Themas (Frage 21). Große Gewinnwerte bedeuten in den Abbildungen, dass der Wissenszugewinn der Schülerinnen und Schüler sehr niedrig war bzw. das letzte Thema schlecht beherrscht wird. Erstrebenswert sind hier also kleine Werte, die bestenfalls den Wert 1 annehmen.
- *Anwendung* bestehenden Wissens in Form von Wiedererkennungsgrad (Frage 24) und Nützlichkeit (Frage 25) schon behandelter Themen in Bezug auf das letzte unterrichtete Thema. Große „Anwendungswerte“ bedeuten in den Abbildungen, dass der Wiedererkennungsgrad bzw. die Nützlichkeit schon behandelter Themen in Bezug auf das letzte unterrichtete Thema niedrig ist. Erstrebenswert sind hier aus fachdidaktischer Sicht m. E. Werte zwischen 1 und 3, so dass das neue Thema mit mindestens 50% auf bekannten Themen aufbaut. Jede Abbildung hat für ein Thema keinen Wert. Dies ist das jeweils erste Thema der speziellen fachdidaktischen Vorgehensweise und kann somit noch keinen Bezug zu vorangegangenen Themen haben.

Die Werte wurden nach den Themen und nicht nach der zeitlichen Unterrichtsfolge geordnet, damit pro Thema sich die beiden Abbildungen leichter vergleichen lassen.

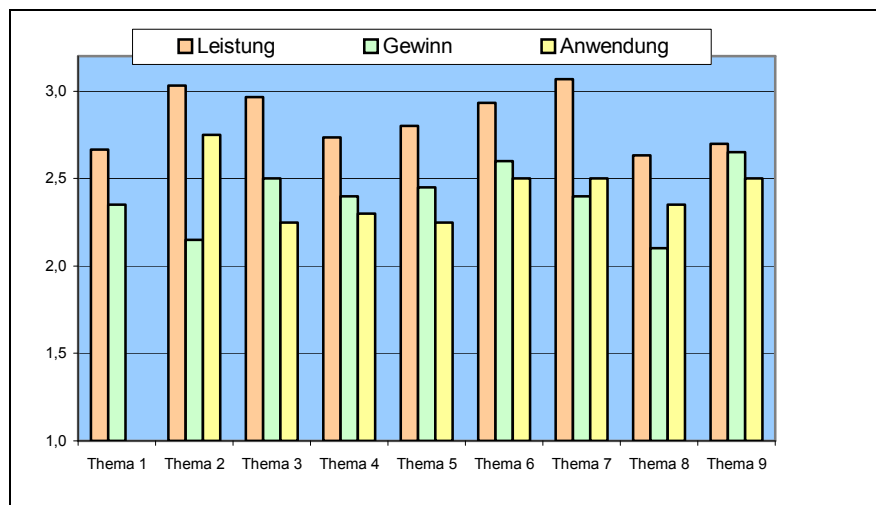


Abbildung 127: Kognitive Dimension (mit sieben Items) aufgliedert in drei Unterbereiche für OOP-First

Folgende Aussagen können für die OOP-First-Klasse getroffen werden (siehe Abbildung 127):

- Der Mittelwert für den „Leistungsaufwand“ ist 2,8, die Mittelwerte für den „Wissensgewinn“ und die „Nützlichkeit bzw. Anwendung bestehenden Wissens“ sind jeweils 2,4.
- Die größte Leistung mussten die Schüler bei den Themen 2 („Datentypen und Steuerstruktur Sequenz“), 3 („Klasse und Objekt“), 6 („Steuerstruktur Iteration“) und 7 („Arrays und Strings“) erbringen, Minimum (2,6) und Maximum (3,1) liegen beim „Leistungsaufwand“ relativ dicht beieinander.
- Das Thema 2 („Datentypen und Steuerstruktur Sequenz inkl. Texteditor“) hat verständlicherweise den höchsten (also schlechtesten) Wert beim Anwendungsbezug zu vorangegangenen Themen, da zum einen erst ein anderes Thema davor unterrichtet wurde, zum anderen die Schülerinnen und Schüler bei dem Thema „Einstieg in den Texteditor“ relativ wenig auf das gelernte Wissen der „OOP-Einführung mit BlueJ“ zurück greifen können.
- Das Thema 2 und das Thema 8 („Vererbung“) wurden laut Schüleraussagen am besten gelernt, das Thema 6 („Steuerstruktur Iteration“) und das Thema 9 („Assoziation“) am schlechtesten.

Folgende Aussagen können für die OOP-Later-Klasse getroffen werden (siehe Abbildung 128):

- Der Mittelwert für den „Leistungsaufwand“ ist 2,7, der Mittelwert für den „Wissensgewinn“ ist 2,2 und der Mittelwert für die „Nützlichkeit bzw. Anwendung bestehenden Wissens“ ist 2,1.
- Die größte Leistung mussten die Schüler beim Thema 3 („Klasse und Objekt“) und beim Thema 9 („Assoziation“) erbringen, Minimum (2,3) und Maximum (3,0) liegen beim „Leistungsaufwand“ weiter auseinander als in der OOP-First-Klasse.
- Das Thema 3 („Klasse und Objekt“) hat verständlicherweise den höchsten (also schlechtesten) Wert beim Anwendungsbezug zu den vorangegangenen Themen, da erst ein anderes objektorientiertes Thema davor unterrichtet wurde (Thema 1: „OOP-Einführung mit BlueJ“) und die Schülerinnen und Schüler bei dem Thema 3 mit der Implementierung von Klassen und der Erzeugung von Objekten (mit Hilfe des Texteditors)

relativ wenig auf das „BlueJ-Wissen“ zurück greifen können. Dort waren u. a. die Klassen vorgegeben und die Objekte konnten mittels Mausklick erzeugt werden.

- Das Thema 9 („Assoziation“) wurde laut Schüleraussagen mit Abstand am schlechtesten gelernt.

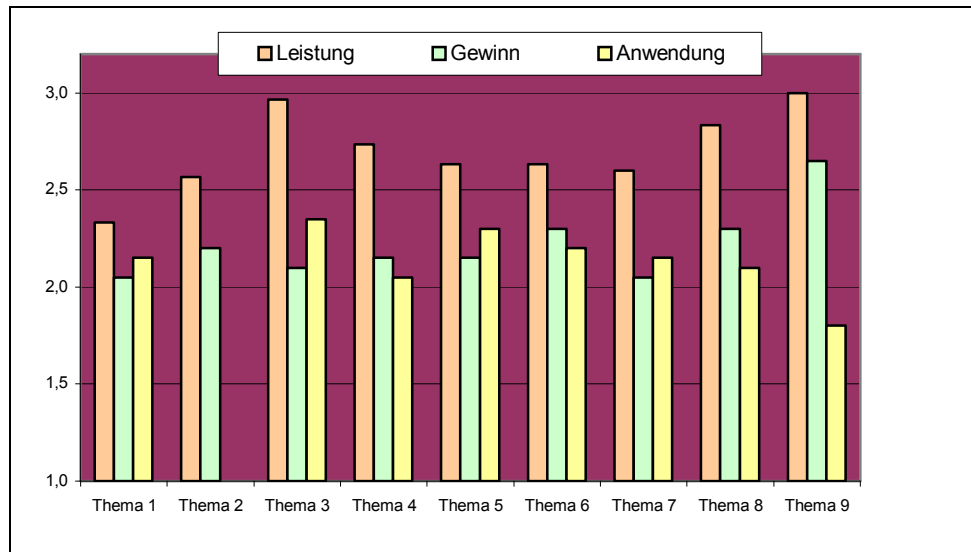


Abbildung 128: Kognitive Dimension (mit sieben Items) aufgegliedert in drei Unterbereiche für OOP-Later

Beim Vergleich der beiden Klassen fällt die Mittelwerts-Differenz ( $2,4 - 2,1 = 0,3$ ) beim Anwendungsbezug zu den vorherigen Themen auf. Die Differenzen bei den einzelnen Themen sind höchst signifikant ( $p = 0,005$ ). Die Irrtumswahrscheinlichkeit liegt nur bei 0,5%. Die Effektstärke ( $d = 1,66$ ) liegt über dem Wert 0,8, der Mittelwertsunterschied stellt somit einen starken Effekt dar (vgl. [Coh88, S. 20-27]). Die weiterführende Interpretation dieses Ergebnisses erfolgt in Kapitel 7.4.5: Warum sind die Themen bei OOP-Later besser vernetzt?

Die Mittelwerts-Differenz ( $2,4 - 2,2 = 0,2$ ) beim Wissenszugewinn ist dagegen knapp nicht signifikant ( $p = 0,06$ ).

#### d) Empfundene Schwere in Abhängigkeit von den Themen

Statt dem Mittelwert aller sieben Items bzw. den drei Bestandteilen des kognitiven Erlebens soll jetzt nur das Item (Frage 12) zum subjektiven Schwere-Empfinden der Schülerinnen und Schüler bezogen auf die einzelnen Themen untersucht werden (siehe Abbildung 129). Es sind die gleichen Messwerte wie die der Kurve „Schwere“ in der Abbildung 124 und Abbildung 125, nur diesmal sind diese nach den Themen statt nach der Zeit geordnet<sup>3</sup>.

Folgende Aussagen können getroffen werden:

- Die Differenz zwischen den Mittelwerten der beiden Klassen (OOP-First: 2,74 / OOP-Later: 2,54) ist nicht signifikant ( $p = 0,17$ ).
- Der augenscheinlich große Unterschied (0,6) zwischen den beiden Klassen bei Thema 1 („Einführung in die OOP mit BlueJ“) ist knapp nicht signifikant ( $p = 0,054$ ).

<sup>3</sup> Wichtiger Hinweis: Die Darstellung dieses Kapitels in zwei Veröffentlichungen ([ES09a: Section 5.5, S. 22-24] und [ES10a, Section 3.5, S. 111]) ist leider falsch, weil sie sich dort fälschlicherweise auf die Antworten zur Frage 8 statt zur Frage 12 bezieht!

- Der Unterschied ( $2,8 - 2,3 = 0,5$ ) beim Thema 2 („Texteditor, Datentypen und Steuerstruktur Sequenz“) ist auch nicht signifikant ( $p = 0,19$ )<sup>4</sup>.

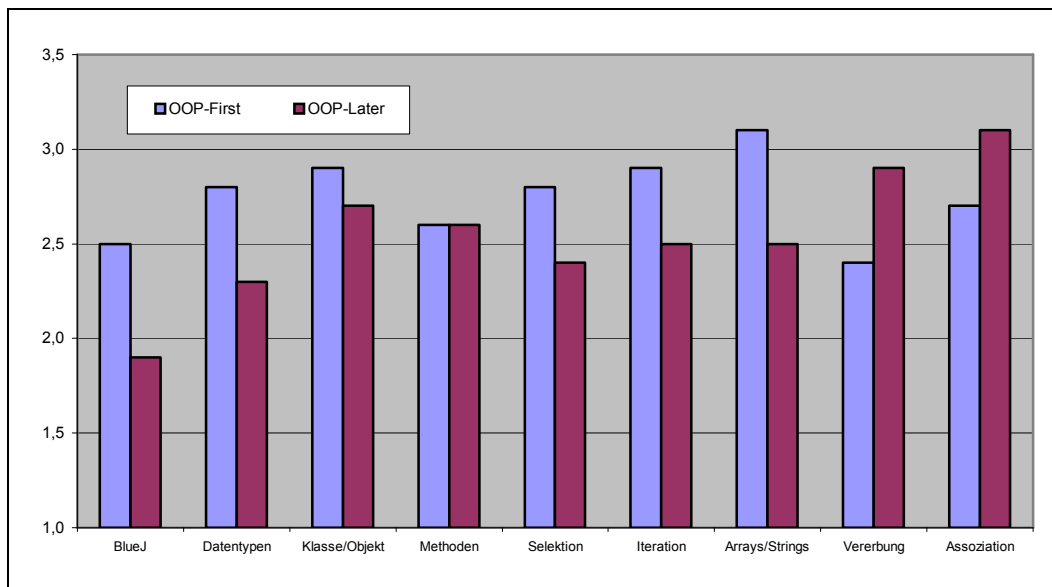


Abbildung 129: „Empfundene Schwere“ (mit nur einem Item: Frage 12) der einzelnen Themen<sup>5</sup>

- Der augenscheinlich große Unterschied (0,6) zwischen den beiden Klassen bei Thema 7 („Arrays und Strings“) ist signifikant ( $p = 0,03$ ). Dies war auch schon bei der kognitiven Dimension bezogen auf alle 7 Items der Fall. Die Effektstärke ( $d = 0,71$ ) liegt unter dem Wert 0,8, der Mittelwertsunterschied stellt somit nur einen mittelstarken Effekt dar (vgl. [Coh88, S. 20-27]).
- Das schwerste Thema für die OOP-First-Klasse ist das Thema 7 („Arrays und Strings“), das schwerste Thema für die OOP-Later-Klasse das Thema 9 („Assoziation“).
- Das leichteste Thema für die OOP-First-Klasse ist das Thema 8 („Vererbung“), dicht gefolgt vom Thema 1 („Einführung in die OOP mit BlueJ“). Das leichteste Thema für die OOP-Later-Klasse ist mit Abstand das Thema 1 („Einführung in die OOP mit BlueJ“).
- Die ersten sieben der neun Themen findet die OOP-Later-Klasse leichter. Nur bei den letzten beiden Themen dreht sich das Verhältnis um.

#### e) Das Schwere-Empfinden im Kontext zu der zeitlichen Abfolge der Unterrichtsthemen

Im Folgenden sollen die Messwerte für die „empfundene Schwere“ der Themen nach dem tatsächlichen Unterrichtsablauf dargestellt werden. Dafür sind in der Abbildung 130 und Abbildung 131 nicht nur die Themenreihenfolge unterschiedlich, sondern es wird sich zusätzlich noch auf den Mittelwert der jeweiligen Messwertreihe bezogen. Die Null-Linie ist somit bei der OOP-First-Klasse an der Stelle 2,74, bei der OOP-Later-Klasse dagegen an der Stelle 2,54. Diese Art der Darstellung eignet sich gut, um die „empfundene Schwere“ der einzelnen Themen miteinander vergleichen zu können. Positive Werte bedeuten dabei, dass das

<sup>4</sup> Im Gegensatz zu der Darstellung in den o. g. Veröffentlichungen: Dort waren die Unterschiede als signifikant benannt worden.

<sup>5</sup> Die Darstellung der Abbildung 129 hätte anstelle von [ES09a, Figure 18] und [ES10a, Figure 10] stehen müssen!

jeweilige Thema als überdurchschnittlich schwer empfunden wird.

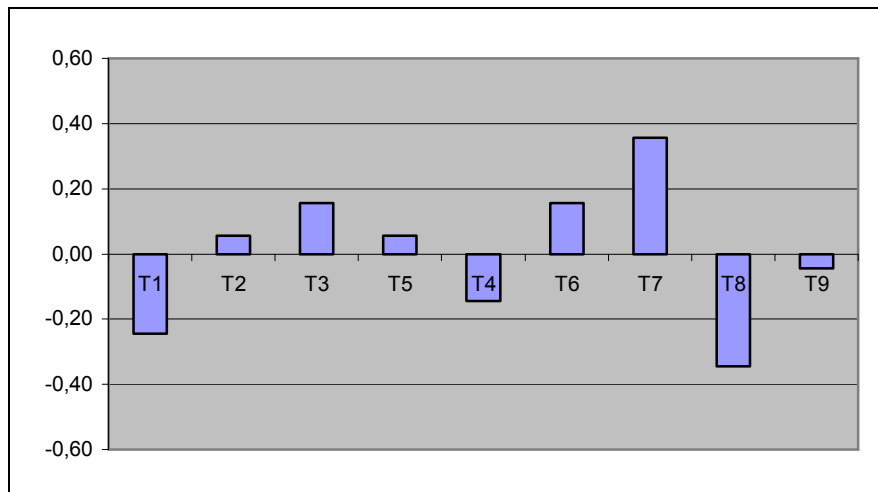


Abbildung 130: Schwere-Empfinden der OOP-First-Klasse in Abhängigkeit der Themenfolge (Null-Linie:  $\bar{\emptyset} = 2,74$ )

Für die OOP-First-Klasse können folgende Aussagen (siehe Abbildung 130) gemacht werden:

- Das Thema 1 („BlueJ-Einstieg“) wird als leicht empfunden.
- Der Wechsel vom ersten Thema (BlueJ: 2,5) auf das zweite Thema (Texteditor und Datentypen: 2,8) scheint den Schülerinnen und Schülern ein bisschen schwerzufallen, die Differenz beträgt 0,3.
- Erst das Thema 4 („Methoden“) wird wieder leichter als der Durchschnitt empfunden.
- Von T4 nach T6 („Iteration“) und von T6 nach T7 („Arrays und Strings“) kommt es zu kleinen Sprüngen (0,3 bzw. 0,2) in Richtung schwerere Themen.
- Der größte Sprung (0,7) findet allerdings in umgekehrter Richtung statt von Thema 7 zu Thema 8 („Vererbung“). Mit Thema 9 („Assoziation“) geht der Wert wieder in Richtung Durchschnitt.

Rein optisch betrachtet kann man von einer „relativ gleichmäßigen Wellenbewegung“ sprechen, der Extremwert (bezogen auf den Mittelwert) beträgt 0,36. Oder seriöser formuliert (siehe linken Box-Whisker-Plot in Abbildung 132): Alle Messwerte befinden sich innerhalb der Spannweite.

Für die OOP-Later-Klasse können folgende Aussagen (siehe Abbildung 131) getätigt werden:

- Das erste Thema, der „Einstieg in den Texteditor und Datentypen“, wird als überdurchschnittlich leicht empfunden (Wert: 2,3).
- Die nächsten drei Themen nähern sich immer mehr dem Durchschnitt.
- Die „Einführung in die OOP mit dem BlueJ-Tool“ wird mit Abstand als leichtestes Thema empfunden (1,9), die Differenz zum vorangegangenen Thema 7 („Arrays und Strings“) beträgt aber 0,6.
- Danach kommt es zu einem noch größeren Sprung (0,8) von Thema 1 zum Thema 3 („Klasse und Objekt“), diesmal aber in die andere Richtung.
- Zum Schuljahresende werden die Themen („Vererbung“ und „Assoziation“) für die Schülerinnen und Schüler immer schwerer.

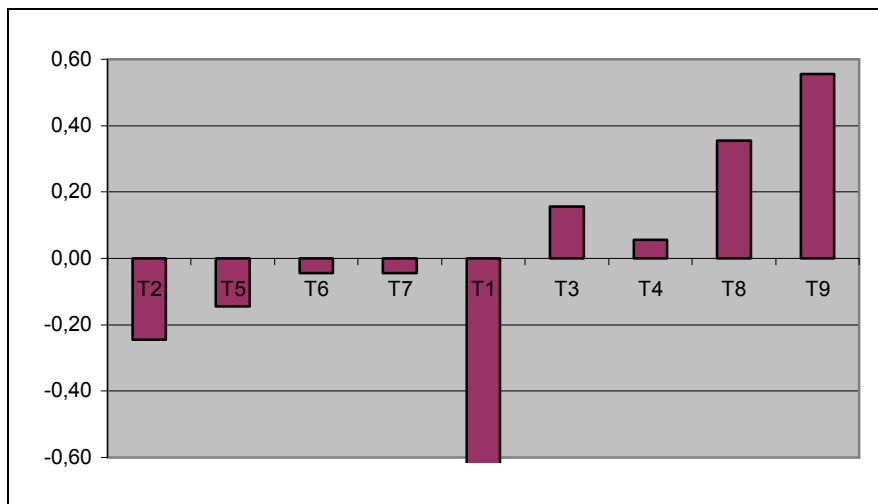


Abbildung 131: Schwere-Empfinden der OOP-Later-Klasse in Abhängigkeit der Themenfolge (Null-Linie:  $\emptyset = 2,54$ )

Rein optisch betrachtet wirkt das Diagramm der OOP-Later-Klasse (wie auch schon in der Vorstudie, siehe Abbildung 102) „unruhig“. Das Maximum ist mit 0,64 wesentlich größer als das beim Diagramm der OOP-First-Klasse. Oder seriöser formuliert (siehe rechten Box-Whisker-Plot in Abbildung 132): Ein Messwert befindet sich außerhalb der Spannweite.

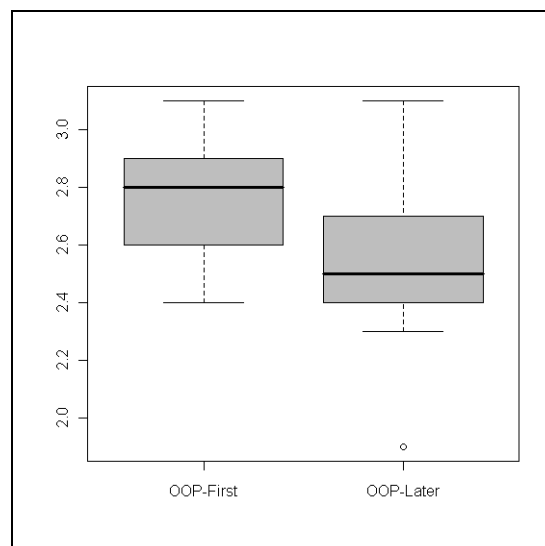


Abbildung 132: Vergleich des Schwere-Empfindens mit Hilfe eines Box-Whisker-Plots (vgl. [Fae07, S. 75-78])

Wenn man beide Box-Whisker-Plots in Abbildung 132 vergleicht, dann können die folgenden Aussagen formuliert werden:

- Der Plot der OOP-Later-Werte hat eine etwas größere Spannweite als der Plot der OOP-First-Werte, die Quartil-Abstände sind aber vergleichbar.
- Ein Messwert befindet sich außerhalb der Spannweite, er gilt also als Ausreißer [SH09, S. 69].
- Die beiden Darstellungen sind auffällig unterschiedlich, z. B. im Hinblick auf den Medianwert und den Ausreißer.

Folgende Aussagen (im Hinblick auf die Box-Whisker-Plots) können bei einem Vergleich mit den Ergebnissen der Vorstudie getroffen werden (siehe Abbildung 132 im Vergleich zu Abbildung 103):

- Die Box-Whisker-Plots unterscheiden sich im Hinblick auf die relative Position der Medianwerte zueinander. Bei der Vorstudie waren die Medianwerte fast gleich, in der Hauptstudie sind sie sehr unterschiedlich.
- Die Box-Whisker-Plots unterscheiden sich im Hinblick auf die Spannweite. Bei der Vorstudie hatte der Plot der OOP-Later-Klasse eine geringere Spannweite (dafür aber zwei Ausreißer) als der der OOP-First-Klasse, in der Hauptstudie dagegen hat er für die OOP-Later-Klasse eine etwas höhere Spannweite (bei nur noch einem Ausreißer).

In der Vorstudie hatte in der OOP-Later-Klasse das Thema „Methoden“ schlecht abgeschnitten. Daher wurde es in der Hauptstudie nicht vor, sondern erst nach dem Einstieg in die objektorientierte Programmierung unterrichtet.

	Thema: Methoden	
	Vorstudie	Hauptstudie
OOP-First	2,5	2,6
OOP-Later	3,2	2,6

Abbildung 133: Schwere-Empfinden des Themas „Methoden“ in der Vor- und Hauptstudie

Folgende Aussagen können (siehe Abbildung 133) bei einem Vergleich mit den Ergebnissen der Vorstudie getroffen werden:

- Gleiches fachdidaktisches Vorgehen (hier in der OOP-First-Klasse) führt zu gleichen Ergebnissen. Die Schwere des Themas „Methoden“ wird in der Vorstudie (mit dem Wert 2,5) von den Schülerinnen und Schülern ähnlich bewertet wie in der Hauptstudie (mit dem Wert 2,6).
- Didaktische Entscheidungen (hier in der OOP-Later-Klasse) können zu einem besseren Unterricht führen. Die Schwere des Themas „Methoden“ wird durch eine zeitliche Verschiebung des Themas (hinter den Einstieg in die objektorientierte Programmierung) für die Schülerinnen und Schüler deutlich gesenkt. Von dem Wert 3,2 (in der Vorstudie) auf den Wert 2,6 (in der Hauptstudie).

Weitere Ergebnisse zum Schwere-Empfinden finden sich im Kapitel 7.4.2 („Welche Themen werden als einfach, welche Themen als schwierig empfunden?“), Kapitel 7.4.3 („Hat das Schwierigkeits-Empfinden etwas mit dem tatsächlichen Schwierigkeitsgrad zu tun?“) und Kapitel 7.4.4 („Gibt es Berührungspunkte zu vergleichbaren Untersuchungen?“) wieder.

#### 6.4.4 Motivationale Dimension

Im Folgenden sollen einzelne Ergebnisse zu den Fragen zur motivationalen Dimension in Abhängigkeit von den drei Frage-Bereichen „Schule“, „Fach Informatik“ und „letztes Thema“ (siehe Abbildung 134) näher dargestellt und diskutiert werden.

	Fragen zur motivationalen Dimension	OOP-First	OOP-Later	Durchschnitt
Fragen zur Schule	4	2,6* p=0,0057	2,3* d=1,52	2,45
Fragen zum Fach Informatik	10	2,3* p=0,0002	1,9* d=2,34	2,10
Fragen zum jeweiligen Lehrer				
Fragen zum letzten Thema	13,16	2,0* p=0,0155	1,8* d=1,30	1,90
	<b>Durchschnitt</b>	2,30	2,00	2,15

Abbildung 134: Subjektives Erleben im Kontext zur motivationalen Dimension und zu den Frage-Bereichen

(\* signifikante Ergebnis-Unterschiede sind fett gedruckt)

Das motivationale Erleben wird in der Regel nur durch ein Item ermittelt. Einzige Ausnahme ist der Schwerpunkt-Frage-Bereich „Thema“, bei dem zwei Items vorgesehen sind. Die Fragen in den einzelnen Bereichen zur Ermittlung dieser Faktoren waren wie folgt formuliert (vgl. Kapitel 5.2.3):

- Frage 4: Meine Motivation in der **Schule** gute Leistungen zu erbringen ist zur Zeit eher ...  
(sehr hoch / sehr niedrig)
- Frage 10: Meine Motivation im **Fach Informatik** gute Leistungen zu erbringen ist zur Zeit ...  
(sehr hoch / sehr niedrig)
- Frage 13: Unabhängig von der Schwere des Stoffes fand ich das **letzte Thema** ...  
(sehr interessant / sehr langweilig)
- Frage 16: Das **letzte Thema** ist meiner Meinung nach für die Informatik ...  
(sehr wichtig / sehr unwichtig)

Die beiden Fragen 13 und 16 fragten nicht direkt nach der Motivation, sondern indirekt nach dem Interesse an den Themen bzw. nach der Wichtigkeit der Themen für die Schülerinnen und Schüler.

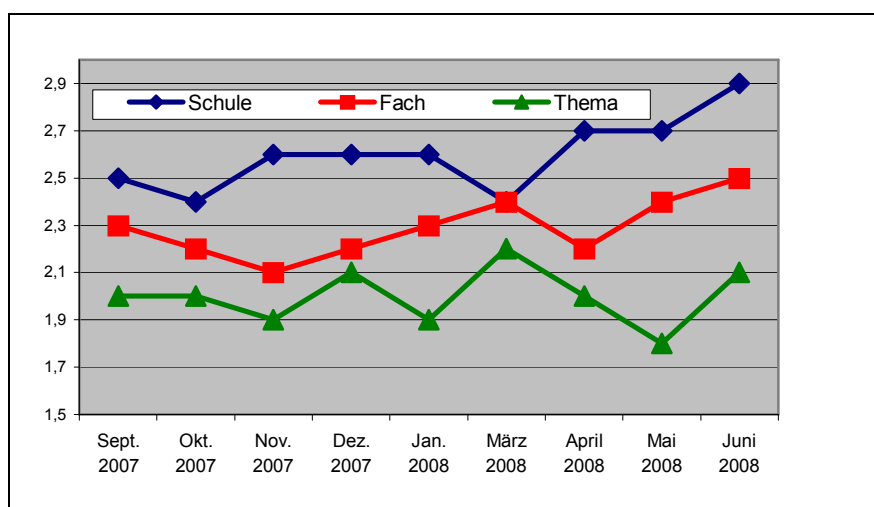


Abbildung 135: Das motivationale Erleben der OOP-First-Klasse



Folgende Aussagen können für die OOP-First-Klasse getroffen werden (siehe Abbildung 135):

- Die Reihenfolge in der Motivation für die verschiedenen Frage-Bereiche ist eindeutig: Die „Schul-Motivation“ ist nicht ganz so hoch wie die „Fach-Motivation“ (größere Skalenwerte  $\hat{=}$  geringere Motivation), die Werte dieser Motivation sind nicht ganz so gut wie die Werte für die Themen.
- Die Werte für die Themen pendeln konstant um einen Mittelwert von 2,0 herum, die Werte für die Schule verschlechtern sich ab dem 6. Messpunkt (März 2008) und die Werte für das Fach verschlechtern sich stetig (mit einer Ausnahme) schon ab dem 3. Messpunkt (November 2007).

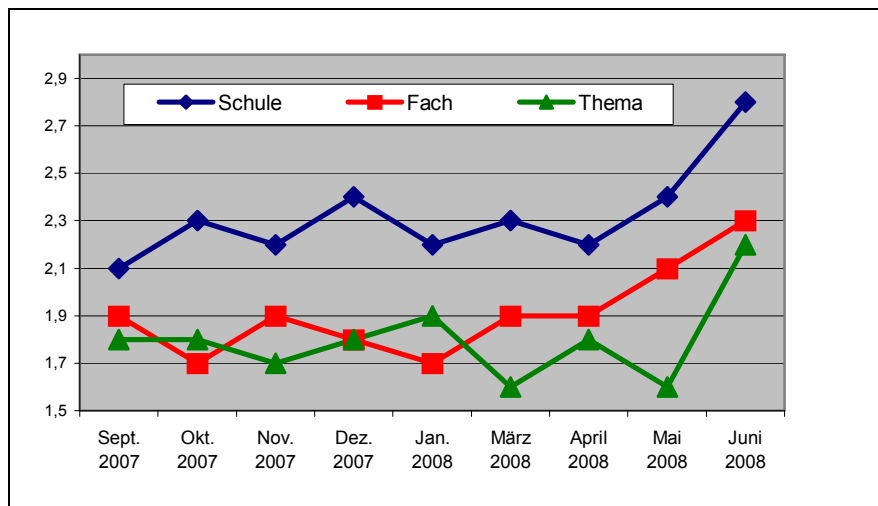


Abbildung 136: Das motivationale Erleben der OOP-Later-Klasse

Folgende Aussagen können für die OOP-Later-Klasse getroffen werden (siehe Abbildung 136):

- Die Reihenfolge ist im Gegensatz zur OOP-First-Klasse nur bedingt eindeutig: Die „Schul-Motivation“ ist zwar wieder nicht ganz so hoch wie die „Fach-Motivation“, die Werte dieser Motivation sind aber ähnlich wie die Themen-Werte.
- Die OOP-Later-Klasse hat am Anfang des Schuljahres eine wesentlich höhere Motivation im Hinblick auf die Schule als die OOP-First-Klasse, zum Schuljahres-Ende gleichen sich die Werte aber an.
- Auch die Werte für die Themen pendeln in der OOP-Later-Klasse konstant um einen Mittelwert herum, der aber mit 1,8 unter dem Mittelwert der OOP-First-Klasse liegt. Einzige Ausnahme ist das letzte Thema („Assoziation“), welches am wenigsten interessant und am wenigsten wichtig erlebt bzw. gesehen wird.
- Die Werte für die Schule verschlechtern sich stetig ab dem 7. Messpunkt (April 2008) und die Werte für das Fach verschlechtern sich stetig schon ab dem 5. Messpunkt (Januar 2008).

Wenn man die beiden Klassen vergleicht, können die folgenden Aussagen gemacht werden:

- In allen drei Frage-Bereichen ist die OOP-Later-Klasse höher motiviert als die OOP-First-Klasse. Diese Unterschiede sind signifikant. Alle Effektstärken liegt über dem Wert 0,8, die Mittelwertsunterschiede stellen somit einen starken Effekt dar (vgl. [Coh88, S. 20-27]).
- Bei beiden Klassen sinkt die Motivation zum Schuljahres-Ende erheblich. Dieser Effekt ist in der OOP-Later-Klasse noch stärker als in der OOP-First-Klasse, so dass am Schuljahres-Ende das Motivationsniveau beider Klassen fast identisch ist (OOP-First: 2,9/2,5/2,1 und OOP-Later: 2,8/2,3/2,2).

Wenn man die Motivation bezogen auf das Thema nicht im Kontext zu dem Unterrichtsverlauf betrachtet, sondern nur im Kontext zum Thema selber, dann ergibt sich die Darstellung in Abbildung 137.

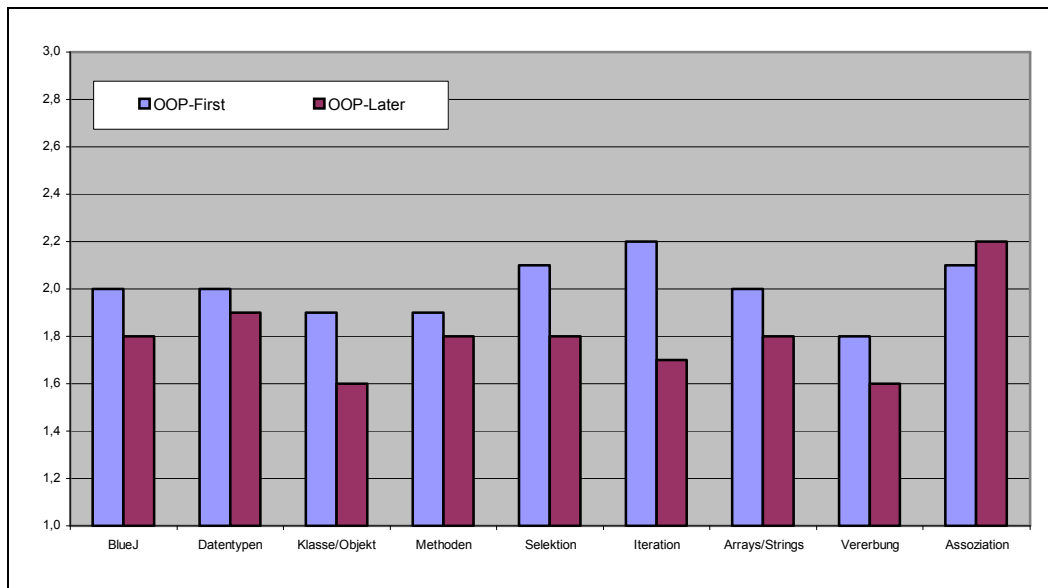


Abbildung 137: Motivationales Erleben in Abhängigkeit von den Themen [ES10a, Figure 8, S. 110]  
(vgl. emotionales Erleben in Abbildung 119 und kognitives Erleben in Abbildung 126)

Folgende Aussagen können getroffen werden [ES10a, S. 110-111]:

- Alle Werte sind gut, sie liegen deutlich unter dem Mittelwert 3,0 der Skala von 1 bis 5. Beide Klassen sind somit motiviert.
- Bis auf eine Ausnahme (Thema 9) ist die OOP-Later-Klasse immer motivierter als die OOP-First-Klasse.
- Der Unterschied in den Mittelwerten ist signifikant ( $p = 0,0155$ ,  $d = 1,30$ ).
- Die größten Unterschiede zwischen beiden Klassen sind bei dem Thema 6 („Iteration“, Differenz = 0,5), dem Thema 5 („Selektion“, Differenz = 0,3) und dem Thema 3 („Klasse und Objekt“, Differenz = 0,3).
- Thema 9 („Assoziation“) empfinden beide Klassen als nicht ganz so interessant und wichtig. Doch auch dieses Thema liegt deutlich unter dem Skalenmittelwert von 3.
- Während bei der OOP-Later-Klasse augenscheinlich nur ein Thema „heraus fällt“ („Assoziation“), sind es bei der OOP-First-Klasse gleich drei Themen („Selektion“, „Iteration“ und „Assoziation“).

Bei allen Vergleichen sollte aber der Maßstab der Diagramme beachtet werden. Würden die Y-Achse den gesamten möglichen Bereich von 1 bis 5 darstellen, dann würden die „augenscheinlichen Unterschiede“ deutlich reduzierter wahrgenommen werden.

### 6.4.5 Subjektives Erleben

Im Folgenden sollen alle drei Erlebnis-Dimensionen zusammengefasst werden zum Faktor „subjektives Erleben“ (evtl. kann man dieses subjektive Erleben als eine Art „Comfort Level“ betrachten). Hierfür werden die

entsprechenden Werte aus Abbildung 115 noch einmal angeführt und zusätzlich einer Signifikanz-Berechnung unterzogen (siehe Abbildung 138)<sup>6</sup>.

Subjektives Erleben	OOP-First	OOP-Later
Schule	<b>2,86*</b> p=0,00006	<b>2,60*</b> d=2,56
Fach Informatik	<b>2,56*</b> p=0,0003	<b>2,31*</b> d=2,52
Lehrer	<b>2,31*</b> p=0,0005	<b>1,94*</b> d=2,05
Thema	<b>2,40*</b> p=0,026	<b>2,22*</b> d=1,16

Abbildung 138: Subjektives Erleben in allen vier Frage-Bereichen  
(vgl. Abbildung 115, \* signifikante Ergebnis-Unterschiede sind fett gedruckt)

Folgende Aussage kann getroffen werden:

- Das subjektive Erleben der OOP-Later-Schüler ist in allen vier untersuchten Bereichen signifikant besser als das subjektive Erleben der OOP-First-Schüler.

Im weiteren Verlauf dieses Kapitels soll das subjektive Erleben fokussiert werden auf die beiden Frage-Bereiche „Fach Informatik“ und „Thema“ [ES10a, S. 111-112]. Dazu wird zuerst der zeitliche Verlauf des subjektiven Erlebens dargestellt (siehe Abbildung 139). Die Interpretation der Schuldaten findet sich in Kapitel 7.4.6 und die Interpretation der Lehrerdaten in Kapitel 7.2.1. Im Kapitel 7.4.1 wird das subjektive Erleben in den Frage-Bereichen „Fach Informatik“ und „Thema“ in einen Kontext zu den Ergebnissen des Nachhaltigkeitstests gestellt.

Folgende Aussagen können getroffen werden (siehe Abbildung 138 und Abbildung 139):

- Das subjektive Erleben bei beiden fachdidaktischen Vorgehensweisen ist immer im positiven Bereich, also unter dem Skalen-Mittelwert von 3.
- Das subjektive Erleben der OOP-Later-Klasse ist besser als das der OOP-First-Klasse.
- Das subjektive Erleben der Themen ist in beiden Klassen besser als das Erleben im Fach.
- Das subjektive Erleben der OOP-First-Klasse bezogen auf das Fach ist über das gesamte Schuljahr sehr konstant (Bereich zwischen 2,5 und 2,6).

---

<sup>6</sup> Die Werte in Abbildung 138 bzw. die Mittelwerte in Abbildung 139 scheinen in einem kleinen Widerspruch zu den Mittelwerten der Abbildung 115 zu stehen. Die beiden Mittelwerte der OOP-First-Klasse liegen jetzt bei 2,56 und 2,40 (statt 2,67 und 2,43) und der OOP-Later-Klasse bei 2,31 und 2,22 (statt 2,40 und 2,23). Die Mittelwerte der Abbildung 115 sind aus den gerundeten Werten zu den drei Erlebnis-Dimensionen berechnet worden (und somit nicht ganz exakt) und weichen ein wenig von den tatsächlichen Werten der Abbildung 138 und Abbildung 139 ab.

- Das subjektive Erleben der OOP-Later-Klasse bezogen auf das Fach ist über das gesamte Schuljahr ziemlich konstant (Bereich zwischen 2,2 und 2,4) wenn man das letzte Thema nicht berücksichtigt.
- Bei der OOP-First-Klasse gibt es eine starke Korrelation ( $r = 0,84$ ) zwischen den Themen-Werten und den Fachwerten. Die Klasse scheint beim subjektiven Erleben nicht zwischen Thema und Fach zu trennen.
- Bei der OOP-Later-Klasse gibt es eine mittlere Korrelation ( $r = 0,77$ ) zwischen den Themen-Werten und den Fachwerten. Auch diese Klasse scheint beim subjektiven Erleben nicht zwischen Thema und Fach zu trennen.
- Die Unterschiede im subjektiven Erleben bezogen auf das Fach (OOP-First: 2,56 / OOP-Later: 2,31) sind höchst signifikant ( $p = 0,0003$ ). Die Effektstärke liegt mit 2,52 deutlich über dem Wert 0,8, der Mittelwertsunterschied stellt somit einen sehr starken Effekt dar (vgl. [Coh88, S. 20-27]).
- Die Unterschiede im subjektiven Erleben bezogen auf die Themen (OOP-First: 2,40 / OOP-Later: 2,22) sind signifikant ( $p = 0,026$ ). Die Effektstärke liegt mit 1,16 über dem Wert 0,8, der Mittelwertsunterschied stellt somit einen relativ starken Effekt dar.
- Das subjektive Erleben der Themen korreliert wenig (OOP-First:  $r = 0,25$  / OOP-Later:  $r = 0,55$ ) mit den Ergebnissen des Nachhaltigkeitstests (vgl. Abbildung 169 und Abbildung 170).

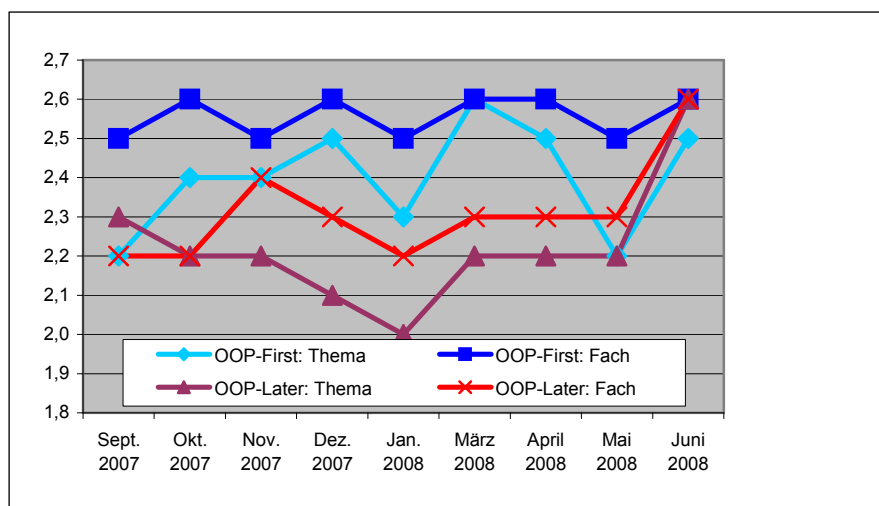


Abbildung 139: Subjektives Erleben in Abhängigkeit vom Fach und von den Themen ([vgl. [ES10a, Figure 12, S. 111])

Beide fachdidaktische Vorgehen erzeugen ein positives subjektives Erleben [ES10a, S. 111]. Die Werte der OOP-Later-Klasse sind dabei signifikant besser als die der OOP-First-Klasse. Sollte man wegen dieser Signifikanz ein OOP-Later-Vorgehen wählen? Diese Diskussion soll u. a. im Kapitel 7.4.6 („Erleben die OOP-Later-Schüler den Unterricht wirklich subjektiv angenehmer?“) und Kapitel 7.5.4 („Gibt es Schlussfolgerungen für den Informatik-Anfangsunterricht?“) geführt werden.

McKinney und Denton [MD04] beschreiben, dass das „Comfort Level“ mit der Zeit abnimmt. Dies kann man (wenn überhaupt) nur bei der OOP-Later-Klasse sehen. Nimmt man das letzte Thema aus der Betrachtung heraus, dann kann man eher wie folgt formulieren: „Das subjektive Erleben der Schülerinnen und Schüler ist für beide fachdidaktischen Vorgehensweisen im Verlauf eines Schuljahres relativ konstant.“ Im Kapitel 7.5.4 („Gibt es Schlussfolgerungen für den Informatik-Anfangsunterricht?“) wird vorgeschlagen, das Thema „Assoziation“

aus dem Informatik-Anfangsunterricht zu entfernen. Dann würden die Werte für das „subjektive Erleben“ am Anfang und am Ende des Schuljahres für beide fachdidaktischen Vorgehensweisen (fast) gleich sein.

## 6.4.6 Spezielle Lerngruppen

Im Folgenden sollen (wie schon bei dem objektiven Aspekt, vgl. Kapitel 6.3.5) drei spezielle Lerngruppen bzw. Stichproben untersucht werden, die (sehr) leistungsstarken Schülerinnen und Schüler, die (sehr) leistungsschwachen Schülerinnen und Schüler und nur die Schülerinnen.

### 1. Die Lerngruppen der leistungsstärksten und leistungsschwächsten Schülerinnen und Schüler

Zuerst sollen die leistungsstärksten Schülerinnen und Schüler untersucht werden. Dies sind in der OOP-First-Klasse die Personen ANPE, SIJO, SUNO und NUYU und in der OOP-Later-Klasse die Personen BIHO, INWO, LASA und SADI (vgl. Kapitel 6.3.5 und Anhang L). Für jedes der neun Themen werden die drei Erlebnis-Dimensionen mit je einem Drittel Anteil zusammengefasst zum Faktor „subjektives Erleben“ (siehe Abbildung 140).

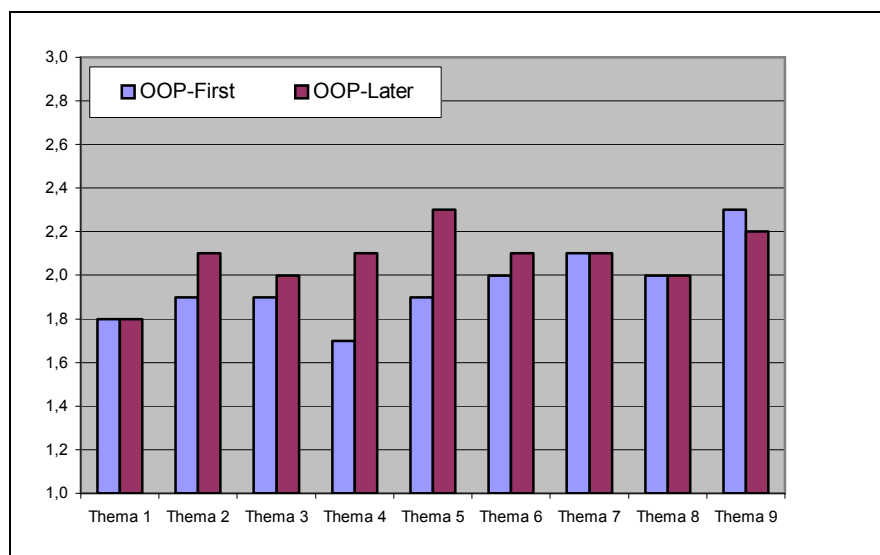


Abbildung 140: Subjektives Erleben der leistungsstärksten Schülerinnen und Schüler

Folgende Aussagen können getroffen werden:

- Der Mittelwerts-Unterschied (OOP-First: 1,96 / OOP-Later: 2,08) ist zwar nicht signifikant ( $p = 0,12$ ), aber trotzdem sehr interessant, weil bei der Gesamtheit aller Schülerinnen und Schüler (OOP-First: 2,40 / OOP-Later: 2,22) nicht nur die OOP-Later-Schüler die besseren (also niedrigeren) Werte erzielten, sondern dieser Unterschiede auch signifikant war ( $p = 0,026$ ).
- Das subjektive Erleben der OOP-Later-Schüler ist bis auf eine Ausnahme (Thema 9) schlechter als das subjektive Erleben der OOP-First-Schüler. Dies ist sehr interessant, weil in der Regel immer die OOP-Later-Werte besser sind als die OOP-First-Werte. Für leistungsstarke Schülerinnen und Schüler scheint im Hinblick auf das subjektive Erleben das OOP-First-Vorgehen von Vorteil zu sein.

Jetzt sollen die leistungsschwächsten Schülerinnen und Schüler untersucht werden. Dies sind in der OOP-First-Klasse die Personen CLUL, LINO, REDE und SIPE und in der OOP-Later-Klasse die Personen BETH, CHMI, JUKL und TAWA (vgl. Kapitel 6.3.5 und Anhang L). Auch hier wird das subjektive Erleben im Hinblick auf die einzelnen Themen dargestellt (siehe Abbildung 141).

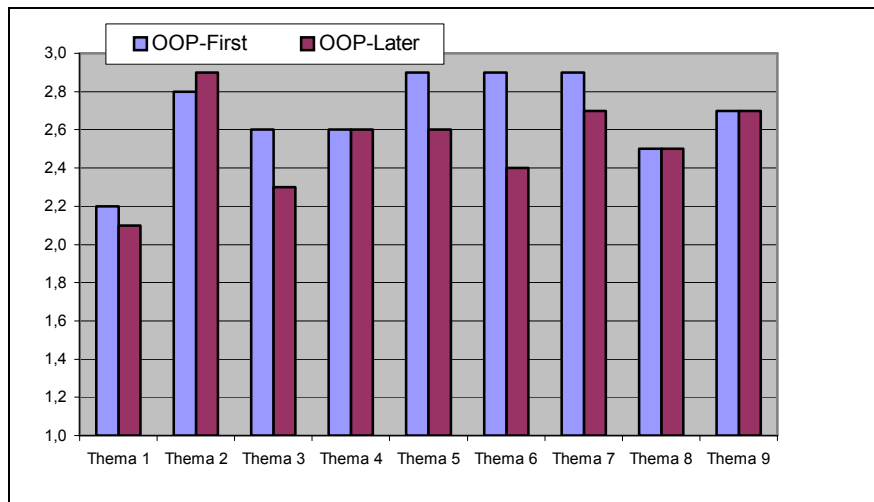


Abbildung 141: Subjektives Erleben der leistungsschwächsten Schülerinnen und Schüler

Folgende Aussagen können getroffen werden:

- Das subjektive Erleben der leistungsschwächsten OOP-First-Schüler ist im Durchschnitt (OOP-First: 2,66 / OOP-Later: 2,54) etwas schlechter als das der leistungsschwächsten OOP-Later-Schüler.
- Der Mittelwerts-Unterschied ( $0,12 = 2,66 - 2,54$ ) ist nicht signifikant ( $p = 0,31$ ) und entspricht ungefähr dem Mittelwerts-Unterschied der gesamten Stichprobe ( $0,18 = 2,40 - 2,22$ ). Dieser war aber signifikant ( $p = 0,026$ ).
- Das subjektive Erleben der OOP-First-Schüler ist bis auf eine Ausnahme (Thema 2) schlechter als das subjektive Erleben der OOP-Later-Schüler. Für leistungsschwache Schülerinnen und Schüler scheint im Hinblick auf das subjektive Erleben das OOP-Later-Vorgehen einen kleinen Vorteil zu haben.

## 2. Die Schülerinnen-Lerngruppe

Im Folgenden soll die spezielle Lerngruppe bzw. Stichprobe der Schülerinnen untersucht werden. In beiden Klassen waren von den 19 Personen zwei weiblich. Dies sind in der OOP-First-Klasse die Schülerinnen ANFR und LINO und in der OOP-Later-Klasse die Schülerinnen BERE und CARU (vgl. Kapitel 6.3.5 und Anhang L). Auch jetzt wird das subjektive Erleben der Schülerinnen im Hinblick auf die einzelnen Themen dargestellt (siehe Abbildung 142).

Für die Abbildung 142 stand in der OOP-First-Klasse beim Thema 6 und beim Thema 8 nur ein Datensatz zur Verfügung, genauso wie in der OOP-Later-Klasse beim Thema 1 und Thema 9 (Krankheit jeweils einer Schülerin). Der OOP-Later-Wert für das Thema 9 ist mit 3,4 außerhalb des Skalenbereichs.

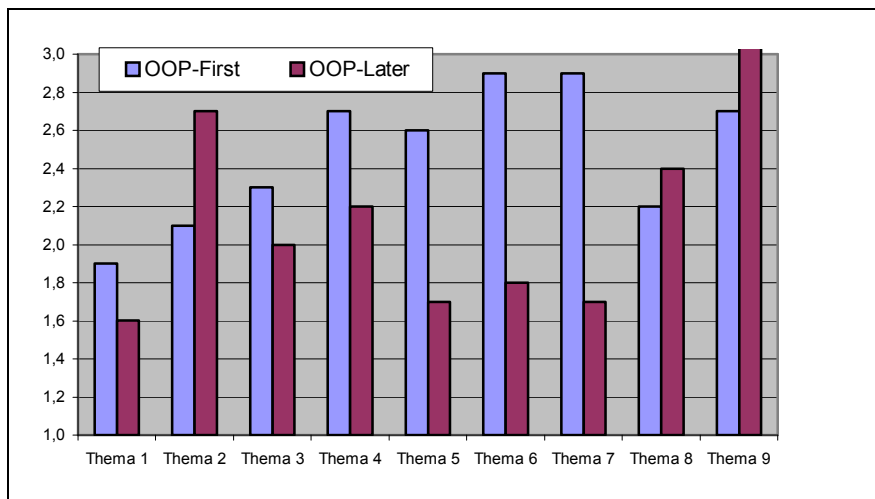


Abbildung 142: Subjektives Erleben der Schülerinnen

Folgende Aussagen können (mit der gebotenen Vorsicht aufgrund der kleinen Lerngruppe) getroffen werden:

- Der Mittelwerts-Unterschied (OOP-First: 2,48 / OOP-Later: 2,17) ist nicht signifikant ( $p = 0,2$ ), im Gegensatz zur gesamten Stichprobe (OOP-First: 2,40 / OOP-Later: 2,22), bei der der Unterschied signifikant war ( $p = 0,026$ ).
- Schülerinnen erleben die Themen sehr unterschiedlich. Die Abweichungen im subjektiven Erleben sind sehr hoch, mal mit einem großen „Vorteil“ für das OOP-First-Vorgehen (Thema 2 und Thema 9), mal mit einem großen „Vorteil“ für das OOP-Later-Vorgehen (Thema 4, Thema 5, Thema 6 und Thema 7).

Es scheint so zu sein, dass Schülerinnen wesentlich emotional beteiligter sind als Schüler, allerdings unabhängig von der fachdidaktischen Vorgehensweise. Im Durchschnitt des subjektiven Erlebens scheint aber das OOP-Later-Vorgehen für Schülerinnen (genauso wie für leistungsschwache Schüler) besser geeignet zu sein.

### 6.4.7 Zusammenfassung

Das Kapitel 6.4 erläutert die Ergebnisse der Hauptstudie bezogen auf das emotionale, kognitive und motivationale Erleben.

Im ersten Kapitel (siehe Kapitel 6.4.1) wurde ein Ergebnis-Überblick bezogen auf die 25 generierten Fragen in den drei Erlebnis-Dimensionen und den vier Frage-Bereichen gegeben. Speziell die Abbildung 115 zeigt alle ermittelten Werte.

Die nächsten Kapitel fokussierten die drei Erlebnis-Dimensionen in den drei Frage-Bereichen „Schule“, „Fach Informatik“ und „letztes Thema“. Der Fragebereich „Lehrer“ wird erst in Kapitel 7.2.1 behandelt.

Die Fragen zur emotionalen Dimension (siehe Kapitel 6.4.2) ermitteln in der Regel den so genannten Wohlfühlfaktor und den so genannten Stressfaktor. Das emotionale Erleben bzw. das emotionale Klima in den o. g. drei Frage-Kategorien in der OOP-First-Klasse ist nicht so gut wie in der OOP-Later-Klasse, in zwei der drei Frage-Bereichen („Schule“ und „Fach Informatik“) sind die Unterschiede sogar signifikant. Da aber der positive subjektive Bereich unter dem Durchschnitt von dem Wert 3 liegt, haben beide Klassen ein gutes emotionales Erle-

ben in allen drei Bereichen. Wenn man den Wohlfühl- und den Stressfaktor nur auf das „letzte Thema“ bezieht, dann fühlt sich bis auf eine Ausnahme (Thema 9: Assoziation) die OOP-Later-Klasse wohler und weniger gestresst als die OOP-First-Klasse. Bei beiden Klassen ist das Thema 1 („Einführung in die OOP mit BlueJ“) das „emotional beste Thema“. Bei beiden Klassen ist das jeweils „emotional schlechteste Thema“ mit jeweils einem Wert von 2,9 immer noch im positiven emotionalen Bereich. Zusätzlich zum Wohlfühl- und Stressfaktor wurde auch direkt danach gefragt, ob die Schülerinnen und Schüler das Fach Informatik gut oder schlecht finden. Die Schülerinnen und Schüler beider Klassen (OOP-First: 1,9 / OOP-Later: 1,7) finden das Fach Informatik gut. Die OOP-Later-Klasse zeigt wie so oft dabei die (noch) besseren Werte.

Die ermittelten Werte zu den Fragen zur kognitiven Dimension (siehe Kapitel 6.4.3) ergeben, dass in allen drei Frage-Kategorien die OOP-Later-Klasse bessere Werte erzielt, bei den Fragen zur „Schule“ und zum „letzten Thema“ sogar signifikant besser. Die Fragen zielen z. B. auf einen so genannten Anforderungsfaktor und einen so genannten Schwierigkeitsfaktor ab. Die Schülerinnen und Schüler erleben die Anforderungen, die die Schule und das Fach Informatik stellen, durchschnittlich als gleich hoch (OOP-First:  $\bar{X} = 3,1$  / OOP-Later:  $\bar{X} = 3,0$ ). Die Werte entsprechen (fast) dem Durchschnitt der Skala von 1 bis 5, also dem Wert 3. Dies ist für die Schule und das Fach Informatik erfreulich: Die Schüler waren durchschnittlich weder überfordert noch unterfordert. Wenn man die sieben Items bezogen auf den Fragebereich „letztes Thema“ betrachtet, dann fällt die Differenz zwischen den Mittelwerten der beiden Klassen (OOP-First: 2,6 / OOP-Later: 2,4) auf. Sie ist signifikant ( $p = 0,016$ ), die Irrtumswahrscheinlichkeit beträgt nur 1,6%, der Mittelwertsunterschied stellt einen starken Effekt dar ( $d = 1,26$ ). Nur ein Thema wird von der OOP-First-Klasse „kognitiv besser“ als von der OOP-Later-Klasse erlebt. Die größte Differenz ergibt sich bei Thema 7 („Arrays und Strings“), sie ist signifikant ( $p = 0,01$ ). Die Irrtumswahrscheinlichkeit beträgt nur 1%, der Mittelwertsunterschied stellt einen (noch) starken Effekt dar ( $d = 0,85$ ). Die sieben Items wurden in Kapitel 6.4.3 in die drei Bereiche „Leistung“, „Gewinn“ und „Anwendung“ eingeteilt. Beim Vergleich der beiden Klassen fällt die Mittelwerts-Differenz von 0,3 beim Anwendungsbezug zu den vorherigen Themen auf. Die Differenzen bei den einzelnen Themen sind höchst signifikant ( $p = 0,005$ ). Die Irrtumswahrscheinlichkeit liegt nur bei 0,5%, der Mittelwertsunterschied stellt einen starken Effekt dar ( $d = 1,66$ ). Es wurde daraus die Aussage abgeleitet: „Beim OOP-Later-Vorgehen wird mehr auf dem Vorwissen aufgebaut als beim OOP-First-Vorgehen!“. Wenn man sich nur auf die Schwere-Frage fokussiert (Item 12), dann findet die OOP-Later-Klasse die ersten sieben der neun Themen leichter (siehe Abbildung 129). Nur bei den letzten beiden Themen dreht sich das Verhältnis um. Der augenscheinlich große Unterschied (0,6) zwischen den beiden Klassen beim Thema 7 („Arrays und Strings“) ist signifikant ( $p = 0,03$ ). Dies war auch schon bei der kognitiven Dimension bezogen auf alle sieben Items der Fall. Der Mittelwertsunterschied stellt aber nur einen mittelstarken Effekt dar ( $d = 0,71$ ). Das schwerste Thema für die OOP-First-Klasse ist das Thema 7 („Arrays und Strings“), das schwerste Thema für die OOP-Later-Klasse das Thema 9 („Assoziation“). Das leichteste Thema für die OOP-First-Klasse (ganz im Gegensatz zur OOP-Later-Klasse) ist das Thema 8 („Vererbung“), dicht gefolgt vom Thema 1 („Einführung in die OOP mit BlueJ“). Das leichteste Thema für die OOP-Later-Klasse ist mit Abstand das Thema 1. Die nächste Fragestellung war, ob es eine Korrelation zwischen der „empfundenen Schwere“ und den Ergebnissen im Nachhaltigkeitstest gibt. Dies muss für die OOP-First-Klasse verneint werden (Korrelationskoeffizient = 0,23) und kann für die OOP-Later-Klasse bejaht werden (Korrelationskoeffizient = 0,61).



Die ermittelten Werte zu den Fragen zur motivationalen Dimension (siehe Kapitel 6.4.4) ergeben, dass in allen drei Frage-Bereichen die OOP-Later-Klasse höher motiviert ist als die OOP-First-Klasse. Diese Unterschiede sind signifikant. Alle Effektstärken liegt über dem Wert 0,8, die Mittelwertsunterschiede stellen somit einen starken Effekt dar. Bei beiden Klassen sinkt die Motivation zum Schuljahres-Ende erheblich. Dieser Effekt ist in der OOP-Later-Klasse noch stärker als in der OOP-First-Klasse, so dass am Schuljahres-Ende das Motivationsniveau (für die Bereiche: Schule/Fach/Thema) beider Klassen fast identisch ist (OOP-First: 2,9/2,5/2,1 und OOP-Later: 2,8/2,3/2,2). Wenn man die Motivation nur bezogen auf die Themen betrachtet (siehe Abbildung 137), dann sind alle Werte gut, sie liegen deutlich unter dem Mittelwert 3 der Skala von 1 bis 5. Beide Klassen sind somit motiviert. Bis auf eine Ausnahme (Thema 9) ist die OOP-Later-Klasse immer motivierter als die OOP-First-Klasse. Der Unterschied (0,2) in den Mittelwerten ist signifikant ( $p = 0,0155$ ,  $d = 1,30$ ). Das Thema 9 („Assoziation“) empfinden beide Klassen als nicht ganz so interessant und wichtig. Doch auch dieses Thema liegt deutlich unter dem Skalenmittelwert von 3.

Im Kapitel 6.4.5 wurden alle drei Erlebnis-Dimensionen gleichberechtigt zusammengefasst zum Faktor „subjektives Erleben“. Das subjektive Erleben wurde fokussiert auf die beiden Frage-Bereiche „Fach Informatik“ und „Thema“. Beide fachdidaktische Vorgehen erzeugen ein positives subjektives Erleben. Das subjektive Erleben der OOP-Later-Klasse ist dabei besser als das der OOP-First-Klasse. Das subjektive Erleben der Themen ist in beiden Klassen besser als das Erleben im Fach. Das subjektive Erleben in beiden Klassen bezogen auf das Fach ist über das gesamte Schuljahr relativ konstant, wenn man das letzte Thema nicht berücksichtigt. Bei beiden Klassen gibt es eine Korrelation (OOP-First:  $r = 0,84$  / OOP-Later:  $r = 0,77$ ) zwischen den Themen-Werten und den Fach-Werten. Die Klassen scheinen beim subjektiven Erleben nicht zwischen dem Thema und dem Fach zu trennen. Die Unterschiede im subjektiven Erleben bezogen auf das Fach (OOP-First: 2,56 / OOP-Later: 2,31) sind höchst signifikant ( $p = 0,0003$ ). Die Unterschiede im subjektiven Erleben bezogen auf die Themen (OOP-First: 2,40 / OOP-Later: 2,22) sind signifikant ( $p = 0,026$ ). Das subjektive Erleben der Themen korreliert wenig (OOP-First:  $r = 0,25$  / OOP-Later:  $r = 0,55$ ) mit den Ergebnissen des Nachhaltigkeitstests.

Im Kapitel 6.4.6 wurden drei spezielle Lerngruppen untersucht. Für leistungsstarke Schülerinnen und Schüler scheint im Hinblick auf das subjektive Erleben das OOP-First-Vorgehen von Vorteil zu sein. Für leistungsschwache Schülerinnen und Schüler scheint das OOP-Later-Vorgehen einen kleinen Vorteil zu haben. Schülerinnen sind wesentlich emotionaler beteiligt als Schüler, allerdings unabhängig von der fachdidaktischen Vorgehensweise. Im Durchschnitt des subjektiven Erlebens scheint aber das OOP-Later-Vorgehen für Schülerinnen (genauso wie für leistungsschwache Schüler) besser geeignet zu sein.

## 6.5 Zusammenfassung

Das Kapitel 6 hatte den Schwerpunkt auf der Darstellung der Ergebnisse der empirischen Studie. Es wurde dabei einerseits unterschieden in Ergebnisse der Vor- und Hauptstudie, andererseits in Ergebnisse zum objektiven und subjektiven Aspekt.

Hierfür wurden zuerst im Kapitel 6.1 die statistischen Grundlagen erläutert.

Das Kapitel 6.2 beschäftigte sich dann mit den Ergebnissen der Vorstudie und nahm einige Interpretationen der Ergebnisse vor.

In dem Vergleichstest der Vorstudie (siehe Kapitel 6.2.1) zum objektiven Aspekt ist bei vier von sechs Themen die OOP-Later-Klasse besser als die OOP-First-Klasse. Zwei von drei OOP-Themen werden sogar signifikant besser bearbeitet. Insgesamt hat die OOP-Later-Klasse beim Vergleichstest um 5% besser abgeschnitten als die OOP-First-Klasse. Als plausibelste Erklärung hierfür wurde die mangelhafte Synchronisation des Unterrichtsgeschehens der beiden fachdidaktischen Vorgehensweisen angeführt und die Theorie geäußert, dass vielleicht das OOP-First-Vorgehen zu einem zu schnellen Vorgehen „verführt“. Es wird deutlich, dass unterschiedlicher Unterricht unterschiedlichen Lernerfolg bewirken kann.

Der Fragebogen (siehe Kapitel 6.2.2) zum subjektiven Aspekt generierte Antworten zu acht Fragen. Bei den drei Fragen zum subjektiven Erleben des Unterrichts gefällt den Schülern das Fach Informatik unabhängig von der fachdidaktischen Vorgehensweise, sie fühlen sich wohl und motiviert. Bei den vier Fragen zum subjektiven Erleben der einzelnen Themen liegen alle Mittelwerte im positiven Bereich, die Mittelwerte der beiden Klassen sind entweder gleich oder unterscheiden sich nur um den Faktor 0,1. Beide fachdidaktischen Vorgehensweisen scheinen spezifische Vor- und Nachteile zu haben, gerade wenn man die Antworten bezogen auf die einzelnen Themen, die Themenreihenfolge oder die Themenübergänge betrachtet (mit dem Schwerpunkt „Schwere-Empfinden“). Auffällig ist z. B., dass das subjektive Schwere-Empfinden der einzelnen Themen nicht korreliert mit den objektiven Ergebnissen des Vergleichstests und dass das BlueJ-Tool (unabhängig von der fachdidaktischen Vorgehensweise) als sehr leicht und wirksam von den Schülerinnen und Schülern erlebt wird. Am Ende des Kapitels 6.2.2 wurden dann schon erste mögliche fachdidaktische Reaktionen auf die Ergebnisse genannt.

Das Kapitel 6.3 fokussierte den Lernerfolg in der Hauptstudie, aufgeschlüsselt in den Vortest (Pre-Test), den Vergleichstest (Post-Test) und den Nachhaltigkeitstest (Follow-Up-Test).

Die Ergebnisse des Vortests (siehe Kapitel 6.3.1) bestätigen die Vermutung, dass die Schülerinnen und Schüler nahezu ohne Programmier-Vorkenntnisse an das OSZ IMT kommen. Sie erzielten im Durchschnitt nur 3,9 von 40 Punkten. Dies sind weniger als 10% und würde der Schulnote 6 entsprechen.

Auf den Vergleichstest (siehe Kapitel 6.3.2) sind die beiden Klassen von den zwei beteiligten Informatik-Lehrern leider sehr unterschiedlich vorbereitet worden und es ergeben sich dadurch (teilweise signifikante) Ergebnisunterschiede. Gerade bei schweren Themen lassen sich durch eine spezielle Präparation einer Klasse deutlich bessere Ergebnisse erreichen.

Beim Nachhaltigkeitstest (siehe Kapitel 6.3.3) sind alle (signifikanten) Unterschiede des Vergleichstests verschwunden. Es werden im Nachhaltigkeitstest keine signifikanten Unterschiede zwischen dem OOP-First-Vorgehen und dem OOP-Later-Vorgehen im Hinblick auf den Lernerfolg festgestellt bzw. gemessen.

Betrachtet man den Lernverlauf mit Hilfe aller drei Tests (siehe Kapitel 6.3.4), ergeben sich fast identische Lern- bzw. Wissenskurven (bezogen auf die ersten acht Themen). Ausgehend von ca. 10% Vorkenntnissen erreichen die Schülerinnen und Schüler nach einem Schuljahr ca. 64,6% Kenntnisse der unterrichteten Informatik-Inhalte. Die Nachhaltigkeit des Wissens ist relativ groß. Der Kenntnisstand sinkt innerhalb von zwei Monaten nur um durchschnittlich 5,7% auf ca. 58,9%.

Das Kapitel 6.3.5 widmete sich dann drei speziellen Lerngruppen. Es werden bei der Lerngruppe der leistungsstärksten Schülerinnen und Schüler keine signifikanten Unterschiede im Lernerfolg in Abhängigkeit von der fachdidaktischen Vorgehensweise gemessen. Eine spezielle Präparation leistungsstarker Schüler auf einen Test

hin zeigt keine offensichtliche Auswirkung. Es gibt bei der Lerngruppe der leistungsschwächsten Schülerinnen und Schüler (teilweise signifikante) Unterschiede in den Ergebnissen des Vergleichstests im Verhältnis zu den Ergebnissen des Nachhaltigkeitstests. Eine spezielle Präparierung leistungsschwacher Schüler auf einen Test hin zeigt eine offensichtliche Auswirkung, die aber nicht nachhaltig ist. Es lassen sich bei der Lerngruppe der Schülerinnen aufgrund der kleinen Stichprobe so gut wie keine Aussagen treffen. Erfreulich ist aber, dass nur eine von vier Schülerinnen der Gruppe der leistungsschwächsten Schülerinnen und Schüler angehört und dass die Schülerinnen bessere Ergebnisse in der Informatik als in der Mathematik erzielen.

Das Kapitel 6.4 befasste sich mit den Fragen in der Hauptstudie zu den drei Erlebnis-Dimensionen.

Im Kapitel 6.4.1 wurde ein Ergebnis-Überblick bezogen auf die 25 generierten Fragen in den drei Erlebnis-Dimensionen und den vier Frage-Bereichen gegeben. In den weiteren Kapiteln wurde der Fragebereich „Lehrer“ ausgelagert in das Kapitel 7.2.1.

Im Kapitel 6.4.2 lag der Fokus auf dem so genannten Wohlfühlfaktor und dem so genannten Stressfaktor, abgeleitet von Fragen zur emotionalen Dimension. Das emotionale Erleben bzw. das emotionale Klima in den drei Frage-Kategorien in der OOP-First-Klasse ist nicht so gut wie in der OOP-Later-Klasse, in zwei der drei Frage-Bereichen („Schule“ und „Fach Informatik“) sind die Unterschiede sogar signifikant. Bei beiden Klassen ist das Thema 1 („Einführung in die OOP mit BlueJ“) das „emotional beste Thema“. Die Schülerinnen und Schüler beider Klassen (OOP-First: 1,9 / OOP-Later: 1,7) finden das Fach Informatik gut.

Im Kapitel 6.4.3 lag der Fokus auf den ermittelten Werten zu den Fragen zur kognitiven Dimension. In allen drei Frage-Kategorien erzielt die OOP-Later-Klasse bessere Werte, bei den Fragen zur Schule und zum Thema sogar signifikant bessere. Die Schüler waren in der Schule und im Fach Informatik durchschnittlich weder überfordert noch unterfordert. Das OOP-Later-Vorgehen scheint mehr auf dem Vorwissen aufzubauen als das OOP-First-Vorgehen. Die OOP-Later-Klasse findet die ersten sieben der neun Themen leichter als die OOP-First-Klasse. Die „empfundene Schwere“ korreliert bei der OOP-Later-Klasse (im Gegensatz zur OOP-First-Klasse) mit den Ergebnissen des Nachhaltigkeitstests (Korrelationskoeffizient = 0,61).

Im Kapitel 6.4.4 lag der Fokus auf den ermittelten Werten zu den Fragen zur motivationalen Dimension. In allen drei Frage-Bereichen ist die OOP-Later-Klasse höher motiviert als die OOP-First-Klasse. Diese Unterschiede sind signifikant, die Mittelwertsunterschiede stellen einen starken Effekt dar. Beide Klassen sind motiviert. Im Fragebereich „Thema“ ist bis auf eine Ausnahme (Thema 9) die OOP-Later-Klasse immer motivierter als die OOP-First-Klasse. Das Thema 9 („Assoziation“) empfinden beide Klassen als nicht ganz so interessant und wichtig.

Im Kapitel 6.4.5 wurden alle drei Erlebnis-Dimensionen gleichberechtigt zusammengefasst zum Faktor „subjektives Erleben“. Das subjektive Erleben wurde fokussiert auf die beiden Frage-Bereiche „Fach Informatik“ und „Thema“. Beide fachdidaktische Vorgehen erzeugen ein positives subjektives Erleben. Das subjektive Erleben der OOP-Later-Klasse ist dabei besser als das der OOP-First-Klasse. Das subjektive Erleben der Themen ist in beiden Klassen besser als das Erleben im Fach Informatik. Die Unterschiede im subjektiven Erleben bezogen auf das Fach sind höchst signifikant. Die Unterschiede im subjektiven Erleben bezogen auf die Themen sind signifikant. Das subjektive Erleben der Themen korreliert wenig mit den Ergebnissen des Nachhaltigkeitstests.

Im Kapitel 6.4.6 wurden drei spezielle Lerngruppen untersucht. Für leistungsstarke Schülerinnen und Schüler scheint im Hinblick auf das subjektive Erleben das OOP-First-Vorgehen von Vorteil zu sein. Für Schülerinnen (genauso wie für leistungsschwache Schüler) scheint dagegen das OOP-Later-Vorgehen besser geeignet zu sein.

Als Fazit bzw. Gesamtergebnis lässt sich folgendes festhalten:

Es wurden keine signifikanten Unterschiede zwischen dem OOP-First-Vorgehen und dem OOP-Later-Vorgehen im Hinblick auf den Lernerfolg der Schülerinnen und Schüler festgestellt bzw. gemessen (die p-Werte für die neun abgefragten Themen liegen zwischen 0,18 und 0,83). Die OOP-Later-Schüler hatten in vielen Bereichen ein (signifikant) besseres subjektives Erleben, sowohl in den Frage-Bereichen („Schule“, „Fach Informatik“ und „Thema“) als auch in den Erlebnis-Dimensionen (emotionales, kognitives und motivationales Erleben). Das emotionale Erleben (ermittelt durch einen „Wohlfühlfaktor“ und einen „Stressfaktor“) war in der OOP-First-Klasse nicht so gut wie in der OOP-Later-Klasse, die Unterschiede waren meist signifikant. Im Hinblick auf das motivationale Erleben war in allen drei Frage-Bereichen die OOP-Later-Klasse (signifikant) höher motiviert als die OOP-First-Klasse. In den beiden Klassen gab es teilweise große Unterschiede in der „empfundenen Schwere“ (ein Aspekt des kognitiven Erlebens) der einzelnen Themen, z. B. bei den Themen „Vererbung“, „Arrays und Strings“ und „Assoziation“. Die „tatsächliche Schwere“ der Themen war dagegen unabhängig von einem OOP-First- bzw. OOP-Later-Vorgehen. Die schwierigsten Themen im Test waren in beiden Klassen die Themen „Assoziation“ und „Arrays“. Bei der Lerngruppe der leistungsschwächsten Schülerinnen und Schüler zeigten die OOP-Later-Schüler (im Nachhaltigkeitstest) einen besseren Lernerfolg als die OOP-First-Schüler. Die spezielle Präparierung leistungsschwacher Schüler auf den Vergleichstest hin zeigte eine (im Gegensatz zu den leistungsstarken Schülern) offensichtliche Auswirkung, die aber nicht nachhaltig war. Es gab bei der Lerngruppe der leistungsstärksten Schülerinnen und Schüler keine signifikanten Unterschiede im Lernerfolg in Abhängigkeit von der fachdidaktischen Vorgehensweise (der p-Wert für das Gesamtergebnis im Nachhaltigkeitstest beträgt 0,90). Für sie scheint aber im Hinblick auf das subjektive Erleben das OOP-First-Vorgehen von Vorteil zu sein. Für Schülerinnen scheint dagegen die OOP-Later-Vorgehensweise besser geeignet zu sein.

Im gesamten Kapitel 6 lag der Schwerpunkt auf der Aufarbeitung der gesammelten Daten und der Darstellung der Ergebnisse. Nur am Rande wurden Ergebnis-Interpretationen und Schlussfolgerungen für den Informatik-Anfangsunterricht vorgenommen. Auch die Vertiefung und Vernetzung der einzelnen Ergebnisse kamen zu kurz. Dies soll sich im nächsten bzw. letzten Kapitel ändern: Dort werden ausgehend von dem o. g. Fazit weiterführende Fragestellungen abgeleitet.

## 7 Fachdidaktische Diskussion: Spezielle Fragestellungen, Auswertungen und Interpretationen

Das Kapitel 7 gliedert sich in sechs Bereiche auf und soll spezielle Fragestellungen beantworten bzw. spezielle Auswertungen und Interpretationen vornehmen.

In den ersten beiden Kapiteln wird das Studiendesign auf das sequenzielle Unterrichten der Themen bezogen (siehe Kapitel 7.1) und auf die im Kapitel 3.3.15 genannten Problemfelder (siehe Kapitel 7.2) kritisch reflektiert. Die Probleme bestanden hierbei bei den unterschiedlichen Lehrern (siehe Kapitel 7.2.1), dem Schreiben eines Vortests (siehe Kapitel 7.2.2) und dem Umgang mit fehlenden Schülern (siehe Kapitel 7.2.3).

Im Hinblick auf den Lernerfolg (siehe Kapitel 7.3) wird analysiert, welche Themen (bei dem Nachhaltigkeitstest) schwer bzw. leicht sind (siehe Kapitel 7.3.1), ob objektorientierte Themen durchschnittlich schwerer oder leichter als prozedurale Themen sind (siehe Kapitel 7.3.2), warum das Thema „Assoziation“ und das Thema „Arrays“ die schwierigsten Themen sind (siehe Kapitel 7.3.3) und ob die objektorientierte Modellierung (OOM) in OOP-Kursen vermittelt werden kann (siehe Kapitel 7.3.4).

Im Hinblick auf das subjektive Erleben des Informatikunterrichts (siehe Kapitel 7.4) wird analysiert, ob das subjektive Erleben Einfluss auf den Lernerfolg hat (siehe Kapitel 7.4.1) und welche Themen als leicht bzw. schwer empfunden werden (siehe Kapitel 7.4.2). Die nächste Fragestellung ist, ob die „empfundene Schwere“ der Themen etwas mit der „tatsächlichen Schwere“ zu tun hat bzw. ob sie Auswirkungen auf den Lernerfolg hat (siehe Kapitel 7.4.3). Gibt es in diesem Zusammenhang Berührungspunkte zu anderen empirischen Untersuchungen (siehe Kapitel 7.4.4)? Ein weitere Fragestellung ist, warum beim OOP-Later-Vorgehen die Themen besser vernetzt sind (siehe Kapitel 7.4.5). Die letzte Frage ist, ob OOP-Later-Schüler den Unterricht wirklich subjektiv angenehmer erleben (siehe Kapitel 7.4.6).

Das letzte Kapitel (siehe Kapitel 7.5) schließt den Kreis von den Ergebnissen der Studie zu den ursprünglichen Thesen und Forschungsfragen und beantwortet die folgenden Fragen: Können die ursprünglichen Thesen (siehe Kapitel 7.5.1), die beiden Forschungsfragen (siehe Kapitel 7.5.2) und die Forschungshypothesen (siehe Kapitel 7.5.3) beantwortet bzw. bewiesen werden? Gibt es Schlussfolgerungen für den Informatik-Anfangsunterricht (siehe Kapitel 7.5.4) und interessante Nebenaspekte (siehe Kapitel 7.5.5)? Gibt es Einschränkungen bei der Gültigkeit (siehe Kapitel 7.5.6) und weiterführende Forschungsfragen (siehe Kapitel 7.5.7)?

Letztlich geht es in weiten Teilen des Kapitels 7 um die Beantwortung der Fragestellung, die vom Fachbereich Mathematik und Informatik der Freien Universität Berlin im Hinblick auf eine Promotion gefordert wird [Pre09]: Welches (fachdidaktische) Problem kann (mit diesem Beitrag) um wie viel besser gelöst werden?

### 7.1 Die Themen

Bei der Überprüfung des Studiendesigns im Hinblick auf die Themen sollen zwei spezielle Fragestellungen beantwortet werden:

- Konnten die Themen in der geplanten Reihenfolge auch tatsächlich unterrichtet werden?
- Konnten die Befragungen zum subjektiven Erleben der einzelnen Themen zeitlich wie geplant erfolgen?

### 7.1.1 Konnte die Themenreihenfolge wie geplant unterrichtet werden?

Im Kapitel 3.3.8 (siehe auch Abbildung 31) wurden die insgesamt neun Themen in Abhängigkeit von der fachdidaktischen Vorgehensweise in einer speziellen Reihenfolge geplant. Dabei flossen schon die Ergebnisse der Vorstudie ein, so dass z. B. das Thema Methoden bei der OOP-Later-Klasse erst nach dem Einstieg in die objektorientierte Programmierung unterrichtet werden sollte.

Die OOP-Later-Klasse wurde in der Hauptstudie exakt nach dieser geplanten Themenreihenfolge unterrichtet (siehe Abbildung 143).

Zeitliche Reihenfolge	1	2	3	4	5	6	7	8	9
OOP-First (geplant)	T1	T2	T3	T4	T5	T6	T7	T8	T9
OOP-First (real)	T1	T2	T3	T5*	T4*	T6	T7	T8	T9
OOP-Later (geplant)	T2	T5	T6	T7	T1	T3	T4	T8	T9
OOP-Later (real)	T2	T5	T6	T7	T1	T3	T4	T8	T9

Abbildung 143: Geplante und tatsächliche Reihenfolge der Themen (\*geänderte Reihenfolge, vgl. Abbildung 31)

In der OOP-First-Klasse wurde dagegen die Reihenfolge der Themen T4 (Methoden) und T5 (Steuerstruktur Selektion) vertauscht. Der unterrichtende Kollege hat sich aufgrund des realen Unterrichtsgeschehens dafür entschieden (vgl. Kapitel 3.3.8), wogegen m. E. aus fachdidaktischer Sicht keine Einwände bestehen. Es geht ja um das Modell, dass beide Vorgehensweisen in der Summe die gleichen Themen unterrichten bzw. dass die Themen in beiden Klassen sequenziell unterrichtet werden können und nicht um eine exakte Festschreibung der Themenfolge. So kann z. B. auch die Reihenfolge der Themen T5 (Steuerstruktur Selektion) und T6 (Steuerstruktur Iteration) umgedreht werden.

### 7.1.2 Konnten die Befragungen zu den Themen zeitlich wie geplant erfolgen?

In Kapitel 3.3.11 wurden die Kurzarbeitspläne (siehe Abbildung 35 und Abbildung 36) zu den beiden Schulhalbjahren und in Kapitel 3.3.8 die geplanten Befragungstermine zum subjektiven Erleben der Schülerinnen und Schüler (siehe Abbildung 31) dargestellt. Da jedes der neun Themen ungefähr die gleiche Priorität besitzen und damit auch die gleiche Unterrichtszeit in Anspruch nehmen sollte, wurden die Befragungen im gleichen monatlichen Abstand geplant.

Geplanter Befragungstermin	Sept. 2007	Okt. 2007	Nov. 2007	Dez. 2007	Febr. 2008	März 2008	April 2008	Mai 2008	Juni 2008
Thema (OOP-First)	T1	T2	T3	T5	T4	T6	T7	T8	T9
Befragungstermin (OOP-First)	27.09. 2007	01.11. 2007	15.11. 2007	13.12. 2007	07.02. 2008	07.03. 2008	24.04. 2008	22.05. 2008	24.06. 2008
Thema (OOP-Later)	T2	T5	T6	T7	T1	T3	T4	T8	T9
Befragungstermin (OOP-Later)	27.09. 2007	30.10. 2007	22.11. 2007	20.12. 2007	31.01. 2008	11.03. 2008	24.04. 2008	29.05. 2008	19.06. 2008

Abbildung 144: Geplante und tatsächliche Termine der Befragungen zum subjektiven Erleben (vgl. Abbildung 31)

In Abbildung 144 sieht man nun bei den Datumsangaben der tatsächlichen Befragungstermine im Vergleich zu den geplanten Terminen, dass die geplanten Befragungstermine auch realisiert werden konnten und dass es einen hohen Grad an (gewünschter) zeitlicher Synchronisation zwischen den beiden Klassen gab.

## 7.2 Die Problemfelder

Bei der Implementierung der empirischen Studie wurden auch drei Probleme (siehe Kapitel 3.3.15) benannt:

- Das Problem, dass die beiden Klassen von unterschiedlichen Lehrern unterrichtet werden.
- Das Problem, dass ein Vortest evtl. die weiteren Ergebnisse beeinflusst.
- Das Problem, dass bei den Umfragen und bei den Tests evtl. einige Schülerinnen und Schüler fehlen.

Die Lösungsansätze sahen wie folgt aus:

- Die Lehrer erhielten exakte Vorgaben zum Lernstoff und zur Methodik, sprachen sich ständig ab und ihr Unterricht wurde dahingehend kontrolliert, ob eine hohe Synchronisation des Unterrichtsgeschehens in beiden Klassen vorliegt (siehe Kapitel 3.3.8, 3.3.9, 3.3.10, 3.3.11 und 3.3.15). Und es wurden verschiedene Messungen vorgenommen, die die Frage beantworten sollten, ob die Lehrer wirklich gleichartig waren (siehe Kapitel 7.2.1).
- Der Vortest wurde nicht mit der Gesamtheit der Versuchspersonen geschrieben, sondern mit den Schülerinnen und Schülern eines Jahrgangs später, um die weiteren Ergebnisse nicht zu beeinflussen (siehe Kapitel 3.3.12 und Kapitel 3.3.15). Dabei ging es nur um den Nachweis bzw. die Widerlegung einer sehr früh geäußerten These, dass nämlich die Schülerinnen und Schüler (fast) ohne Programmierkenntnisse an das OSZ IMT kommen. Es wird geprüft, ob die Ergebnisse des Vortests diese These untermauert haben (siehe Kapitel 7.2.2).
- Für das Fehlen von Schülerinnen und Schülern wurden eindeutige Cut-off-Kriterien (siehe Kapitel 3.3.15) festgelegt, sowohl für die Umfragen (siehe Abbildung 41) als auch für die Tests (siehe Abbildung 42). Nun muss analysiert werden, inwiefern diese Kriterien Einfluss auf die Gesamtheit der gewerteten Versuchspersonen genommen haben (siehe Kapitel 7.2.3).

In den nächsten drei Unterkapiteln sollen somit Antworten auf die folgenden Fragestellungen gegeben werden:

- Waren die beiden an der Studie beteiligten Lehrer wirklich gleichartig?
- Hat der Vortest die These zum Schüler-Vorwissen untermauert?
- Gab es Probleme mit fehlenden Schülern bei den Umfragen und bei den Tests?

### 7.2.1 Waren die Lehrer wirklich gleichartig?

Bei der Analyse des emotionalen, kognitiven und motivationalen Erlebens der Schülerinnen und Schüler (siehe Kapitel 6.4) wurden bewusst die Antworten zum Fragebereich „Lehrer“ ausgeklammert. In Abbildung 145 wird der entsprechende Ausschnitt aus der Abbildung 115 noch einmal abgebildet. Die Analyse dieser Daten soll jetzt

mit der Fragestellung erfolgen, inwiefern die Lehrer gleichartig oder unterschiedlich waren. Die Interpretation der restlichen Daten ist nämlich umso glaubwürdiger, je gleichartiger die beiden beteiligten Lehrer waren.

Dimension	Emotionale Dimension		Kognitive Dimension		Motivationale Dimension		Durchschnitt Subjektives Erleben	
Frage-Nr.	Fragen 7, 18, 19		Frage 17		Keine Fragen			
Klasse	OOP- First	OOP- Later	OOP- First	OOP- Later	OOP- First	OOP- Later	OOP- First	OOP- Later
Frage zum jeweiligen Lehrer	<b>2,2*</b> p=0,0016	<b>1,9*</b> d=1,90	<b>2,4*</b> p=0,0025	<b>2,0*</b> d=1,69			<b>2,31*</b> p=0,0005	<b>1,94*</b> d=2,05

Abbildung 145: Beurteilung der beiden Lehrer

(vgl. Abbildung 115 und Abbildung 138, \* signifikante Ergebnis-Unterschiede sind fett gedruckt)

Die einzelnen Fragen lauteten:

- Frage 7: Den unterrichtenden **Lehrer** finde ich ...  
(sehr gut / sehr schlecht) bzw. keine Aussage
- Frage 17: Die Darstellung und **Veranschaulichung** der Lerninhalte des letzten Themas waren ...  
(sehr verständlich / sehr unverständlich)
- Frage 18: Die **Arbeitsatmosphäre** in der Klasse beim letzten Thema war eher ...  
(sehr gut / sehr schlecht)
- Frage 19: Die **Unterstützung** des Lehrers bei meinen Lernversuchen beim letzten Thema war ...  
(sehr gut / sehr schlecht) bzw. keine Aussage

Die fett hervorgehobenen Wörter werden in anderen Darstellungen (siehe Abbildung 147 und Abbildung 148) als Kennzeichnung der Frage benutzt.

Folgende Aussagen können getroffen werden (siehe Abbildung 145):

- In beiden Frage-Bereichen (emotionale und kognitive Dimension) erzielt der OOP-Later-Lehrer bessere Werte als der OOP-First-Lehrer. Diese Unterschiede sind signifikant. Beide Effektstärken liegen deutlich über dem Wert 0,8, die Mittelwertsunterschiede stellen somit einen starken Effekt dar (vgl. [Coh88, S. 20-27]).
- Trotzdem befinden sich auch die Werte des OOP-First-Lehrers im guten Bereich.
- Vergleicht man nur die Frage 7 (siehe Abbildung 146) der Hauptstudie mit der gleichen Frage der Vorstudie (Frage 3, siehe Abbildung 96), dann verbleibt der OOP-First-Lehrer mit dem Durchschnittswert 1,8 (in der Vorstudie) auf seinem Wert. Dies spricht für die Aussagekraft der Daten der OOP-First-Klasse.
- Der OOP-Later-Lehrer verbessert sich aber von dem Wert 2,2 (in der Vorstudie) auf den Wert 1,8 (in der Hauptstudie). Dies ist sehr erfreulich. Dieser Lehrer hat erfolgreich die Lehren aus der Vorstudie gezogen (siehe Kapitel 3.3.14) und erfolgreich sein Verhalten verbessert. Nun werden die beiden Lehrer im Hinblick auf die Frage 7 in der Hauptstudie gleichartig von den Schülern erlebt.



Betrachtet man die Antworten zu den Fragen 7, 17, 18 und 19 im Kontext zu den Themen (siehe Anhang L), so ergeben sich die Daten aus der Abbildung 146.

OOP-First-Lehrer					OOP-Later-Lehrer				
Frage	7	17	18	19	Frage	7	17	18	19
Thema 1	1,5	2,1	2,1	1,9	Thema 1	1,7	1,6	1,5	2,1
Thema 2	1,9	2,5	2,3	2,4	Thema 2	1,9	2,1	1,6	2,0
Thema 3	1,9	2,4	2,4	2,0	Thema 3	1,8	1,9	2,1	1,9
Thema 4	2,0	2,2	2,5	2,3	Thema 4	1,9	1,7	1,9	2,1
Thema 5	1,8	2,4	2,7	2,4	Thema 5	1,9	2,1	1,8	1,7
Thema 6	1,9	2,7	2,9	2,4	Thema 6	1,7	2,1	1,8	2,2
Thema 7	1,8	2,7	2,8	2,4	Thema 7	1,7	1,8	2,1	1,9
Thema 8	1,7	2,0	2,5	2,1	Thema 8	1,6	2,0	2,3	2,1
Thema 9	1,9	2,6	3,0	2,7	Thema 9	2,1	2,3	2,3	2,1
<b>Durchschnitt</b>	<b>1,8</b>	<b>2,4</b>	<b>2,6</b>	<b>2,3</b>	<b>Durchschnitt</b>	<b>1,8</b>	<b>2,0</b>	<b>1,9</b>	<b>2,0</b>

Abbildung 146: Beurteilung der Lehrer in Abhängigkeit von den Themen (vgl. Anhang L)

Folgende Aussagen können im Hinblick auf die Mittelwerte zu den einzelnen Fragen getroffen werden (siehe Abbildung 146):

- Die beiden Lehrer werden von den beiden Klassen als gleich gut beurteilt (Frage 7).
- Der OOP-Later-Lehrer kann anscheinend die Lerninhalte etwas besser veranschaulichen als der OOP-First-Lehrer (Frage 17).
- Die Arbeitsatmosphäre war beim OOP-Later-Lehrer anscheinend wesentlich besser als beim OOP-First-Lehrer (Frage 18).
- Der OOP-Later-Lehrer hat anscheinend die Schülerinnen und Schüler bei ihren Lernversuchen etwas besser unterstützt als der OOP-First-Lehrer (Frage 19).

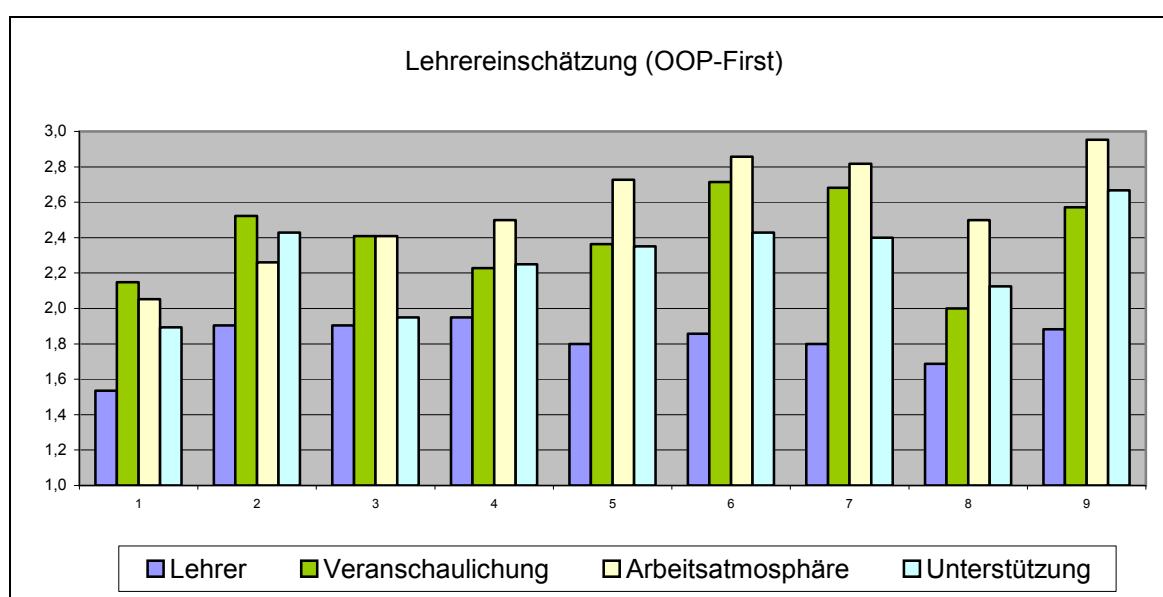


Abbildung 147: Beurteilung des OOP-First-Lehrers (mit vier Indikatoren) in Abhängigkeit von den Themen

Wenn man jetzt diese Daten anders darstellt (siehe Abbildung 147 und Abbildung 148), dann können die Unterschiede bei den Antworten zu den vier Fragen besser visualisiert werden und die folgenden Aussagen getroffen werden:

- Der OOP-First-Lehrer wird von den Schülerinnen und Schülern bei der direkten Frage zu seiner Person (Frage 7) deutlich besser beurteilt als bei den Antworten zu den drei anderen Fragen.
- Beim OOP-Later-Lehrer liegen die Werte der Antworten zu den vier Fragen näher zusammen. Sein Bild ist „in sich stimmiger“.

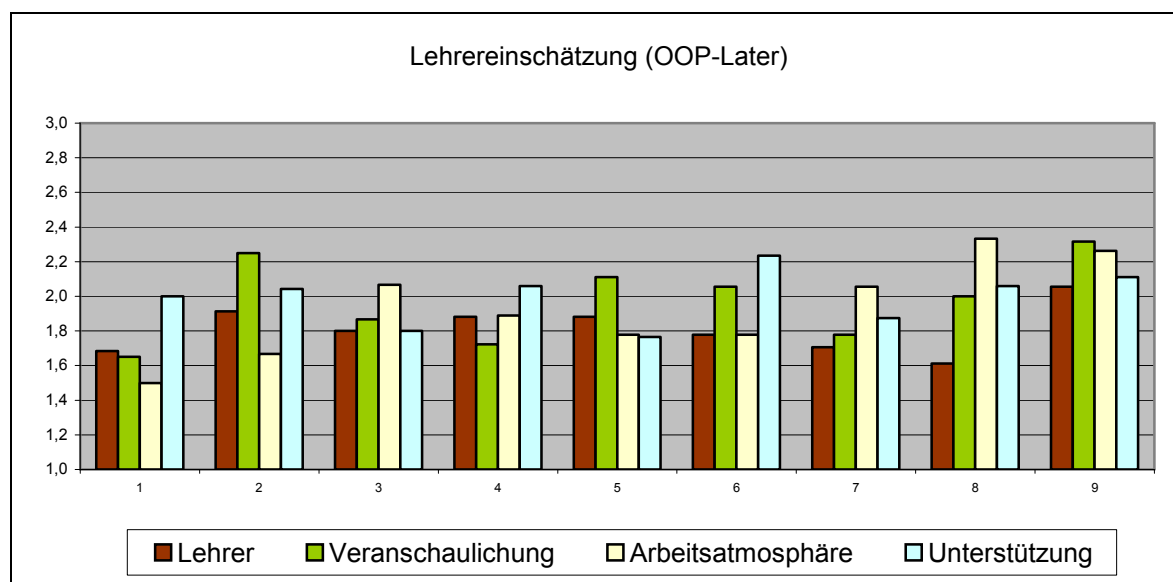


Abbildung 148: Beurteilung des OOP-Later-Lehrers (mit vier Indikatoren) in Abhängigkeit von den Themen

Die Schülerinnen und Schüler wurden am Ende des Schuljahres 2007/2008 zusätzlich gebeten, einen Online-Evaluations-Fragebogen des OSZ IMT, welcher dort schon seit langem als Qualitätskontrolle für den Unterricht eingesetzt wird, auszufüllen. Dieser unterteilt sich in vier Bereiche:

- Rahmenbedingungen des Unterrichts (Fragen 1 bis 7, siehe Abbildung 149)
- Zufriedenheit mit dem Lehrer (Fragen 8 bis 13, siehe Abbildung 150)
- Individuelle Komponenten (Fragen 14 bis 16, siehe Abbildung 151)
- Allgemeine Einschätzung (Frage 17, siehe Abbildung 152)

Es gibt fünf Antwortmöglichkeiten zum Ankreuzen (von „sehr positiv“ über „neutral“ bis „sehr negativ“). Den fünf Ankreuzmöglichkeiten werden die Zahlen von 1 bis 5 zugeordnet.

Folgende Aussagen können getroffen werden:

- Die Rahmenbedingungen wurden vom OOP-Later-Lehrer ( $\bar{x} = 1,9$ ) etwas besser gestaltet als vom OOP-First-Lehrer ( $\bar{x} = 2,1$ , siehe Abbildung 149).
- Beide Klassen waren gleich zufrieden ( $\bar{x} = 1,8$ , siehe Abbildung 150) mit dem Lehrer. Dieser Wert deckt sich mit dem Wert aus Frage 7 (auch der Wert 1,8 für beide Lehrer, siehe Abbildung 146).
- Die Werte der individuellen Komponente der Schülerinnen und Schüler sind in der OOP-First-Klasse ( $\bar{x} = 2,3$ ) deutlich schlechter als in der OOP-Later-Klasse ( $\bar{x} = 1,8$ , siehe Abbildung 151). Dies könnte z. B. an

der unterschiedlichen fachdidaktischen Vorgehensweise liegen.

- Bei der zusammenfassenden Frage nach der allgemeinen Einschätzung des Unterrichts sind die Durchschnittswerte (mit 2,3 und 1,75, siehe Abbildung 152) vergleichbar mit den gemittelten Gesamtwerten (mit 2,31 und 1,94, siehe Abbildung 145) aus den Antworten zu den Fragebögen zum subjektiven Erleben.

OOP-First-Lehrer	OOP-Later-Lehrer
1. Informationen im Zusammenhang mit dem Unterricht (Termine, Organisatorisches) erfolgten rechtzeitig und umfassend. („Stimmt völlig“ bis „stimmt überhaupt nicht“)	
2,15	1,65
2. Das Ausmaß an Arbeitsunterlagen und Informationen für den Unterricht war ... („völlig ausreichend“ bis „völlig unzureichend“)	
1,75	1,85
3. Die Sozialform des Unterrichts (Frontalunterricht, Gruppenunterricht, Partnerarbeit, Einzelarbeit) wurde ... („ausreichend variiert“ bis „überhaupt nicht variiert“)	
2,05	1,80
4. Die Aktionsform des Unterrichts (Lehrervortrag, Lehrer-Schüler-Gespräch, Schülervortrag, offene Diskussion, Rollenspiele etc.) wurde ... („ausreichend variiert“ bis „überhaupt nicht variiert“)	
2,25	1,80
5. Der Medieneinsatz (Videos, Animationen, OH-Folien, Versuche, Demonstrationen, Tafelarbeit, etc.) wurde ... („ausreichend variiert“ bis „überhaupt nicht variiert“)	
2,10	1,85
6. Die Bedeutung der Lerninhalte des Unterrichts für mein späteres Leben ist meiner Meinung nach ... („sehr groß“ bis „sehr gering“)	
2,15	1,90
7. Der Bezug des Unterrichts zu aktuellen Ereignissen und alltäglichen Fragestellungen war ... („sehr hoch“ bis „sehr gering“)	
2,35	2,65

Abbildung 149: Beurteilung der Rahmenbedingungen des Unterrichts

OOP-First-Lehrer	OOP-Later-Lehrer
8. Die fachliche Kompetenz und die Kenntnisse der/des Lehrerin/Lehrers sind ... („sehr hoch“ bis „sehr gering“)	
1,55	1,25
9. Der Unterrichtsstoff wurde klar und übersichtlich vermittelt. („Trifft voll zu“ bis „Trifft überhaupt nicht zu“)	
2,00	2,05
10. Mit der sprachlichen Ausdrucksweise und der Verständlichkeit der Erklärungen der/des Lehrerin/Lehrers war ich ... („sehr zufrieden“ bis „sehr unzufrieden“)	
1,95	1,85
11. Mit dem persönlichen Auftreten (Engagement, Freude, Spaß am Lehren) der/des Lehrerin/Lehrers war ich ... („sehr zufrieden“ bis „sehr unzufrieden“)	
1,75	1,55
12. Die soziale Kompetenz der/des Lehrerin/Lehrers (Hilfsbereitschaft, Verhalten gegenüber Kritik, etc.) ist ... („sehr hoch“ bis „sehr gering“)	
1,65	1,90
13. Das Eingehen der/des Lehrerin/Lehrers auf Beiträge und Fragen der Schülerinnen/Schüler war für mich ... („völlig ausreichend“ bis „völlig unzureichend“)	
1,90	2,00

Abbildung 150: Beurteilung der Zufriedenheit mit dem Lehrer

OOP-First-Lehrer	OOP-Later-Lehrer
14. Mein persönliches Interesse an den im Unterricht behandelten Themen war ... („sehr groß“ bis „sehr gering“)	
2,30	1,80
15. Mit dem Stoff und den Inhalten des Unterrichts kam ich ... („sehr gut zurecht“ bis „überhaupt nicht zurecht“)	
2,20	1,95
16. Das Unterrichtsklima war ... („sehr angenehm“ bis „sehr unangenehm“)	
2,30	1,70

Abbildung 151: Beurteilung der individuellen Komponenten

OOP-First-Lehrer						OOP-Later-Lehrer					
17. Insgesamt erhält der Unterricht von mir die Note											
1	2	3	4	5	6	1	2	3	4	5	6
4	11	2	2	0	1	6	13	1	0	0	0
<b>Durchschnitt: 2,3</b>						<b>Durchschnitt: 1,75</b>					

Abbildung 152: Allgemeine Einschätzung des Unterrichts

Folgende verschiedene Interpretationen sind denkbar:

- Die beiden Lehrer sind von ihren individuellen Werten im Großen und Ganzen gleichartig. Die Störvariable „unterschiedliche Lehrer“ hat somit (fast) keinen Einfluss auf die Unterschiede in den Ergebnissen. Die Unterschiede in den Ergebnissen sind das Ergebnis der unterschiedlichen fachdidaktischen Vorgehensweisen.
- Die besseren Werte des OOP-Later-Lehrers haben zu den (teilweise signifikant) besseren subjektiven Erlebniswerten der Schülerinnen und Schüler in den drei Erlebnis-Dimensionen geführt.
- Die besseren Werte des OOP-Later-Lehrers haben dafür gesorgt, dass trotz mathematischer Vorteile der OOP-First-Klasse (um ca. 3,3%, vgl. Kapitel 6.3.2) die OOP-Later-Klasse bei dem Nachhaltigkeitstest (mit 54,9%, vgl. Kapitel 6.3.3) genauso gut war wie die OOP-First-Klasse (mit 54,4%, siehe Abbildung 107).

## 7.2.2 Hat der Vortest die Vermutung über die Vorkenntnisse der Schüler bestätigt?

In Kapitel 4.3.3 wurde entschieden, nicht die Schülerinnen und Schüler der Hauptuntersuchung, sondern die des darauf folgenden Jahrgangs mit einem Vortest zu untersuchen (unter der Annahme, dass sich dieser Jahrgang nicht signifikant von dem Jahrgang davor unterscheidet).

Die <b>Arbeiten in der E-Phase</b> (11. Schuljahr) werden mit Punkten und Noten nach folgender Skala bewertet:																
Prozent	100	≥95	≥90	≥85	≥80	≥75	≥70	≥65	≥60	≥55	≥50	≥45	≥35	≥20	≥10	<10
Note	1+	1	1-	2+	2	2-	3+	3	3-	4+	4	4-	5+	5	5-	6
Punkte	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Abbildung 153: Bewertungsschlüssel der Einführungsphase (Berufliches Gymnasium)

[[http://www.oszint.de/4-service/download/unterricht/noten\\_abt4.pdf](http://www.oszint.de/4-service/download/unterricht/noten_abt4.pdf), geprüft 05.05.2011]

Als Begründung wurde genannt, dass die Schülerinnen und Schüler nicht zu stark mit ihrem „Nichtwissen“ konfrontiert werden sollen, damit ihre Motivation für eine Studienteilnahme aufrechterhalten wird. Es ging bei dem Vortest um die Bestätigung oder Widerlegung der Vermutung, dass die Schülerinnen und Schüler nahezu ohne Programmier-Vorkenntnisse an das OSZ IMT kommen.

Die Ergebnisse (siehe Kapitel 6.3.1) bestätigen die Vermutung: die Schülerinnen und Schüler erzielten im Durchschnitt nur 3,9 von 40 Punkten. Dies sind weniger als 10% und würde der Schulnote 6 (siehe Bewertungsschlüssel in Abbildung 153) entsprechen.

### 7.2.3 Gab es Probleme mit fehlenden Schülern?

In der OOP-Later-Klasse (OG72) gab es verstärkt Probleme mit fehlenden Schülern (38 fehlende Umfragen, siehe Abbildung 154). Die Situation in der OOP-First-Klasse (OG71) war deutlich besser (mit nur elf fehlenden Umfragen).

Thema	OOP- First	OOP- Later
T1	2	3
T2	1	1
T3	1	7
T4	0	5
T5	1	6
T6	1	5
T7	0	4
T8	4	4
T9	1	3
<b>Summe</b>	<b>11</b>	<b>38</b>
<b>Durchschnitt</b>	<b>1,2</b>	<b>4,2</b>

Abbildung 154: Anzahl der fehlenden Schüler pro Befragungstermin

In Kapitel 3.3.15 wurden Vorgaben für die Wertung der ermittelten Daten gemacht:

- Die Cut-off-Kriterien für die Umfragen (vgl. Abbildung 41) legen fest, ab welcher Anzahl an teilgenommenen Umfragen die Umfragedaten und die Testdaten gewertet werden.
- Die Cut-off-Kriterien für die Tests (vgl. Abbildung 42) legen fest, dass nur die Schüler gewertet werden, die sowohl am Vergleichstest als auch am Nachhaltigkeitstest teilgenommen haben.
- Schüler, die als Wiederholer in den beiden Klassen sind, werden zwar bei den Umfragen gewertet, nicht aber bei den Tests.

In der OOP-Later-Klasse nahmen zwei Schüler (CHGE und REYV) nur an vier Umfragen teil, ein Schüler (VEVA) nur an fünf Umfragen. Diese drei Schüler werden also bei der Testauswertung nicht berücksichtigt (siehe Abbildung 155), wobei der eine Schüler (VEVA) ohnehin vorher von der Schule abgegangen ist. Zwei Schüler (CHGE und REYV) haben den Vergleichstest mitgeschrieben. Nur einer von beiden (CHGE) absolvierte auch den Nachhaltigkeitstest. Er wird aber sowohl wegen der Cut-off-Kriterien für die Umfragen aus der

Wertung herausgenommen als auch wegen seines Wiederholerstatus (vgl. Kriterien für die Tests in Kapitel 3.3.15).

Anzahl der Umfragen	Schüleranzahl OOP-First	Schüleranzahl OOP-Later	Werden die Daten der Umfragen gewertet?	Werden die Daten der Tests gewertet?
6-9	22	19	ja	ja
4-5	0	3	ja	nein
1-3	0	0	nein	nein

Abbildung 155: Cut-off-Kriterien für die Umfragen (vgl. Abbildung 41)

In der OOP-First-Klasse gab es einen Schüler (MARE), der nur an sieben Umfragen teilgenommen hat, alle anderen haben an acht bzw. neun Umfragen teilgenommen. Somit können alle Schüler der OOP-First-Klasse nach den Umfrage-Kriterien zu den Tests zugelassen werden (siehe Abbildung 155). Ein Schüler (NAGO) fehlte aber dann dauerkrank beim Vergleichstest, so dass sein Nachhaltigkeitstest aus der Wertung fällt. Zusätzlich werden zwei Schüler (PETH und ANDA) als Wiederholer aus der Wertung herausgenommen (siehe Abbildung 156).

Klasse	Anzahl der Schüler mit ausreichenden Umfrage-Ergebnissen	Anzahl der Schüler, die gegen Wertungskriterien verstoßen	Anzahl der gewerteten Schüler
OOP-First	22	3	19
OOP-Later	19	0	19

Abbildung 156: Gewertete Testdaten

Am Ende führen die Vorgaben aus Kapitel 3.3.15 dazu, dass von den 22 Schülerinnen und Schülern in den beiden Klassen jeweils drei Personen nicht in den Vergleichstest-Tabellen bzw. Nachhaltigkeitstest-Tabellen mit Ergebnisdaten vertreten sind (siehe Anhang N und Anhang P).

## 7.3 Der Lernerfolg

In Kapitel 6.3 wurde ausführlich der Lernerfolg der Schülerinnen und Schüler (in der Hauptstudie) nach einem Schuljahr in Abhängigkeit vom fachdidaktischen Vorgehen dokumentiert und analysiert. In diesem Kapitel erfolgt jetzt überwiegend eine Betrachtung aller Schülerinnen und Schüler unabhängig von der fachdidaktischen Vorgehensweise. Von den Daten der beiden Klassen wird der Durchschnitt gebildet.

Es sollen dabei die folgenden Fragen beantwortet werden:

- Welche Themen sind schwer, welche leicht (siehe Kapitel 7.3.1)?
- Sind prozedurale oder objektorientierte Themen schwerer (siehe Kapitel 7.3.2)?
- Warum sind Assoziationen und Felder die schwierigsten Themen (siehe Kapitel 7.3.3)?
- Kann OOM in OOP-Kursen vermittelt werden (siehe Kapitel 7.3.4)?

### 7.3.1 Welche Themen sind schwierig, welche einfach?

In diesem Unterkapitel soll ermittelt werden, welche Unterrichtsthemen (unabhängig von der fachdidaktischen Vorgehensweise) für die Schüler schwierig und welche einfach gewesen sind [ES09a, S. 20] [ES09c, S. 129]. Dazu werden zuerst die Resultate des Nachhaltigkeitstest (siehe Abbildung 107) für die beiden Klassen gemittelt und nach Schwierigkeitsgrad geordnet (siehe Abbildung 157).

Nr.	Thema	Anforderungsbereich II in %	Max. Punkte	Alle Schüler Punkte
T5	Steuerstruktur Selektion	90	10	7,0
T2	Datentypen und Steuerstruktur Sequenz	70	10	6,8
T1 / T3	Einführung in die OOP Klassenimplementierung und Objekterzeugung	70	20 (10)	13,2 (6,6)
T8	Vererbung	70	10	5,9
Z2	Zusatz: Objektorientierte Modellierung (OOM)	100	10	5,8
T6	Steuerstruktur Iteration	80	10	5,6
Z1	Zusatz: Dynamische OOP	80	10	5,2
T4	Methoden	70	10	5,0
T7	Arrays und Strings	90	10	3,7
T9	Assoziation	80	10	2,1
	<b>Summe bzw. Durchschnitt</b>	<b>79,1</b>	<b>110</b>	<b>60,1 (54,6%)</b>

Abbildung 157: Geordnete Ergebnisse des Nachhaltigkeitstests in der Hauptstudie 2007/2008 (alle Schüler zusammen, vgl. [ES09a, Fig. 9, S. 20], [ES09c, Abb. 9, S. 129], Abbildung 61 und Abbildung 107)

Zusätzlich werden drei Leistungsbereiche unterschieden, die sich nach dem Notenschlüssel am Beruflichen Gymnasium (siehe Abbildung 153) richten:

- Mangelhafte Leistungen (10 – 44 Prozent, in Abbildung 157: 1,0 – 4,4 Punkte, rot gekennzeichnet)
- Ausreichende Leistungen (45 – 59 Prozent, in Abbildung 157: 4,5 – 5,9 Punkte, gelb gekennzeichnet)
- Befriedigende Leistungen (60 – 74 Prozent, in Abbildung 157: 6,0 – 7,4 Punkte, grün gekennzeichnet)

Betrachtet man die Ergebnisse unabhängig von den Anforderungsbereichen (siehe Abbildung 60), dann sind die vier einfachsten Themen das Thema „Steuerstruktur Selektion“, das Thema „Datentypen und Steuerstruktur Sequenz“, das Thema „Einführung in die OOP“ und das Thema „Klasse und Objekt“. Die zwei schwierigsten Themen sind das Thema „Arrays und Strings“ und das Thema „Assoziation“.

Man muss m. E. aber das Ergebnis im Verhältnis zum Schwierigkeitsgrad der Aufgaben betrachten. Dazu wird der prozentuale Anteil des Anforderungsbereichs II mit berücksichtigt (indem das Produkt aus Punkten und AF II gebildet wird). Zusätzlich werden die Ergebnisse des Themas 7 („Array und Strings“) aufgeschlüsselt in das Thema 7a („Arrays“) und das Thema 7b („Strings“) (siehe Abbildung 158)<sup>7</sup>.

<sup>7</sup> Die Schülerinnen und Schüler erzielten im Mittel 2,4 Punkte von möglichen 7 Punkten (also 34%) beim Thema „Arrays“ und 1,3 Punkte von möglichen 3 Punkten (also 43%) beim Thema „Strings“. Zusammen also 3,7 Punkte von 10 möglichen Punkten. Der Anforderungsbereich II war beim Thema „Arrays“ 86% (6 Punkte von 7 Punkten) und beim Thema „Strings“ 100%.

Nr.	Thema	Punkte * AF II
T5	Steuerstruktur Selektion	6,3
Z2	Zusatz: Objektorientierte Modellierung (OOM)	5,8
T2	Datentypen und Steuerstruktur Sequenz	4,8
T1 / T3	Einführung in die OOP Klassenimplementierung und Objekterzeugung	4,6
T6	Steuerstruktur Iteration	4,5
T7b	Strings	4,3*
Z1	Zusatz: Dynamische OOP	4,2
T8	Vererbung	4,1
T4	Methoden	3,5
T7a	Arrays	2,9*
T9	Assoziation	1,7
	<b>Durchschnitt</b>	<b>4,3</b>

Abbildung 158: Ergebnisse des Nachhaltigkeitstests unter Berücksichtigung des Anforderungsniveaus  
(alle Schüler zusammen, mit Zusatzthemen,\* berechnete Werte)

Die Einordnung in die drei Leistungsbereiche erfolgt nach dem gleichen Schema wie oben (ausgehend von dem Maximalwert  $87 = 0,791 * 110$ ), also nach dem Notenschlüssel am Beruflichen Gymnasium (siehe Abbildung 153):

- Mangelhafte Leistungen (10 – 44 %, in Abb. 158 und 160: 0,9 – 3,8 Punkte \* AF II, rot gekennzeichnet)
- Ausreichende Leistungen (45 – 59 %, in Abb. 158 und 160: 3,9 – 5,1 Punkte \* AF II, gelb gekennzeichnet)
- Befriedigende Leistungen (60 – 74 %, in Abb. 158 und 160: 5,2 – 6,4 Punkte \* AF II, grün gekennzeichnet)

Jetzt können die folgenden Aussagen getroffen werden (siehe Abbildung 158):

- Das Thema „OOM“ klettert in der Tabelle nach oben und gilt jetzt als einfaches Thema (grüner Bereich).
- Das Thema „Arrays“ ist das schwierigere Teilthema (roter Bereich) beim Thema „Arrays und Strings“. Dort erbringen die Schüler nur mangelhafte Leistungen. Beim Teilthema „Strings“ werden dagegen ausreichende Leistungen (gelber Bereich) erbracht.
- Das Thema „Vererbung“ rutscht in der Tabelle nach unten und ist knapp im gelben Bereich.
- Das Thema „Methoden“ behält zwar seinen drittletzten Platz, wird aber jetzt statt dem „ausreichenden“ (also gelben) Bereich dem „mangelhaften“ (also roten) Bereich zugeordnet.
- Die Themen „Datentypen und Steuerstruktur Sequenz“, „Einführung in die OOP“ und „Klasse und Objekt“ rutschen zwar nur um einen Platz zurück, allerdings fallen sie damit vom „befriedigenden“ (also grünen) Bereich in den „ausreichenden“ (also gelben) Bereich.

Im Folgenden soll die Schwere bzw. der Schwierigkeitsgrad der Themen im Kontext zu der sequenziellen Abfolge der Themen im konkreten Unterricht fokussiert werden (siehe Abbildung 159).



Monat (ca.)	OOP-First (vgl. Abbildung 157)	OOP-Later (vgl. Abbildung 157)	OOP-First (vgl. Abbildung 158)	OOP-Later (vgl. Abbildung 158)
Sept. 2007	T1	T2	T1	T2
Okt. 2007	T2	T5	T2	T5
Nov. 2007	T3	T6	T3	T6
Dez. 2007	T5	T7	T5	T7
Febr. 2008	T4	T1	T4	T1
März 2008	T6	T3	T6	T3
April 2008	T7	T4	T7	T4
Mai 2008	T8	T8	T8	T8
Juni 2008	T9	T9	T9	T9

Abbildung 159: Schwierigkeit der Themen, geordnet nach der sequenziellen Abfolge der Themen

(links ohne bzw. rechts mit Berücksichtigung des Anforderungsniveaus, vgl. [ES09a, Fig. 10, S. 20], Abbildung 157 und Abbildung 158)

In der linken Tabelle ist die farbliche Kennzeichnung nach der Abbildung 157 vorgenommen worden, also ohne die Berücksichtigung des Anforderungsniveaus der Aufgaben zu den einzelnen Themen. Nach dieser Darstellung spricht einiges für die OOP-First-Vorgehensweise, weil sie in der Regel (Ausnahme: T7 bzw. T8) das fachdidaktische Prinzip berücksichtigt, dass im Unterricht „vom Leichten zum Schweren“ vorgegangen werden soll [ES09a, S. 20]. Dies ist bei der OOP-Later-Vorgehensweise nicht der Fall.

Bei der rechten Tabelle mit den farblichen Kennzeichnungen nach der Abbildung 158, also mit der Berücksichtigung des Anforderungsniveaus der Aufgaben zu den einzelnen Themen, fallen keine großen Abweichungen zwischen der OOP-First- und der OOP-Later-Vorgehensweise auf. Das Prinzip „vom Leichten zum Schweren“ ist in beiden Vorgehensweisen nur rudimentär zu erkennen. Die einfachen (grünen) Themen liegen zwar vor den schwierigen (roten) Themen, aber die „mittelschwierigen“ (gelben) Themen sind „bunt dazwischen gewürfelt“.

Zusammenfassend kann man m. E. sagen, dass man aus der Darstellung des Schwierigkeitsgrads der Themen im Kontext zu der sequenziellen Abfolge der Themen keine Empfehlung für die eine oder andere fachdidaktische Vorgehensweise ableiten kann.

### 7.3.2 Sind prozedurale oder objektorientierte Themen schwieriger?

Im letzten Unterkapitel wurden die Ergebnisse der beiden Klassen im Nachhaltigkeitstest gemittelt, quantitativ geordnet und in drei Leistungsbereiche aufgeteilt. Jetzt soll zusätzlich eine Aufteilung in so genannte prozedurale und objektorientierte Themen erfolgen [ES09a, S. 20].

Damit ergibt sich die Darstellung in Abbildung 160. Die so genannten prozeduralen Themen (auf den Plätzen 1, 2, 5, 8 und 9) und die objektorientierten Themen (auf den Plätzen 3, 4, 6, 7 und 10) sind „unregelmäßig im Mix“ über alle Leistungsbereiche angeordnet und haben „augenscheinlich“ die gleiche durchschnittliche Schwierigkeit. Genau berechnet erzielen die prozeduralen Themen (Durchschnitt (reale Punkte) = 5,6 / Durchschnitt (Punkte \* AFII) = 4,40) etwas bessere Resultate als die objektorientierten Themen (Durchschnitt (reale Punkte) = 5,1 / Durchschnitt (Punkte \* AFII) = 3,86). Diese Unterschiede sind aber nicht signifikant. Auch sind zwei der drei schwierigsten Themen nicht objektorientierte Themen, sondern prozedurale Themen.

Platz	Nr.	Thema	Punkte * AF II
1	T5	Steuerstruktur Selektion	6,3
2	T2	Datentypen und Steuerstruktur Sequenz	4,8
3 / 4	T1 / T3	Einführung in die OOP Klassenimplementierung und Objekterzeugung	4,6
5	T6	Steuerstruktur Iteration	4,5
6	T7b	Strings	4,3
7	T8	Vererbung	4,1
8	T4	Methoden	3,5
9	T7a	Arrays	2,9
10	T9	Assoziation	1,7
		<b>Durchschnitt</b>	<b>4,13</b>

Abbildung 160: Prozedurale und objektorientierte Themen, geordnet nach der Schwierigkeit  
(alle Schüler zusammen, ohne Zusatzthemen, vgl. [ES09a, Fig. 10, S. 20] und Abbildung 158)

Somit kann nicht mit der Schwierigkeit dieser beiden Themenbereiche argumentiert werden, wenn es darum geht, sich zwischen den beiden fachdidaktischen Vorgehensweisen entscheiden zu müssen mit dem Argument, dass schwierige Themen später unterrichtet werden sollten [ES09a, S. 20]. Beachtet werden muss auch, dass die diskutierten Werte die Durchschnittswerte aus beiden Klassen sind. Sie sind somit unabhängig von der fachdidaktischen Vorgehensweise.

Zusammenfassend kann die folgende Aussage getroffen werden [ES09c, S. 131 und ES09a, S. 20]:

- Die Themen der OOP sind bei dieser Studie nicht (signifikant) schwieriger als die prozeduralen Themen. In jeder Themenart finden sich schwierige und einfache Themen wieder.

### 7.3.3 Warum sind Assoziationen und Felder die schwierigsten Themen?

Hinweis: Dieses Unterkapitel ist fast identisch mit dem Kapitel 5.3 in dem Papier zur ICER'09 [ES09a, S. 20-21].

Beim Thema „Assoziation“ erreichten alle Schüler zusammen nur 2,1 von 10 möglichen Punkten und beim Thema „Arrays“ nur 2,4 von 7 möglichen Punkten (quasi 3,4 von 10 Punkten). Dies entspricht nach der Notentabelle des Beruflichen Gymnasiums (siehe Abbildung 153) der Note 5. Warum hatten die Schülerinnen und Schüler so große Schwierigkeiten bei diesen beiden Themen?

Für die Beantwortung der Frage sollen die Fragen zu dem Thema „Arrays“ betrachtet werden. Eine typische Multiple-Choice-Frage ist in Abbildung 161 zu sehen. Um diese Frage beantworten zu können müssen die Schüler Kenntnisse darüber besitzen, wie in Java Felder bzw. Arrays konstruiert werden, wie die einzelnen Elemente nummeriert werden, dass der Feldindex bei 0 beginnt und dass Feldelemente vom Datentyp „int“ vom System mit dem Wert 0 initialisiert werden.

Nr.	Aufgabe	Antworten zum Ankreuzen inkl. Lösung
4.2	<pre>int[] x = new int[25];</pre> <p>Welche Aussage(n) trifft/treffen nach der Ausführung der Programmzeile zu?</p>	<p><b>X</b> x.length hat den Wert 25 <b>1P</b></p> <p><input type="radio"/> x[0] hat den Wert null</p> <p><b>X</b> x[24] hat den Wert 0</p> <p><b>X</b> x[25] ist nicht definiert</p>

Abbildung 161: Beispielhafte Multiple-Choice-Frage zum Thema „Arrays“  
(vgl. [ES09a, Figure 11 bezogen auf den Vergleichstest] und Nachhaltigkeitstest in Anhang O)

Bei einer anderen Frage (siehe Abbildung 162) müssen die Schüler wissen, wie man Felder mit Hilfe von Schleifen (Steuerstruktur Iteration) durchläuft.

Nr.	Aufgabe	Lösung
4.4	<p>Initialisieren Sie das folgende Feld mit Hilfe einer Zählschleife mit den Werten ‚a‘, ‚b‘ und ‚c‘!</p> <pre>char[] zeichenfeld = new char[3];</pre>	<pre>zeichen = 'a'; for (int i = 0; i &lt; 3; i++)     zeichenfeld[i] = zeichen++;</pre> <p><b>3P</b></p>

Abbildung 162: Beispielhafte Frage und Antwort zum Thema „Arrays“  
(vgl. [ES09a, Figure 12 bezogen auf den Vergleichstest] und Nachhaltigkeitstest in Anhang O)

Bei den Fragen zum Thema „Assoziation“ müssen die Schüler z. B. Assoziationen implementieren, wobei Teile des Quelltexts vorgegeben werden (siehe Abbildung 163).

Die Schüler müssen hierfür eine Referenz (Attribut: hauptwohnsitz) von einem Menschen auf einen Wohnsitz (1:1-Assoziation) und eine Referenz (Attribut: nebenwohnsitze) von einem Menschen auf ein Array von Wohnsitzen (1:n-Assoziation) vorsehen. Dies ist für die meisten Schüler eine schwierige Aufgabe.

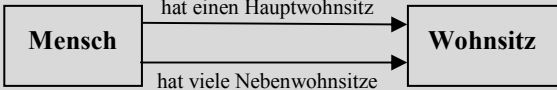
Nr.	Aufgabe	Quelltext mit Lücken inkl. Lösung
9.3	<p>Gegeben ist das folgende Szenario:</p>  <p><b>Ergänzen Sie den folgenden Quelltext!</b></p>	<pre>public class Mensch {     private String name;     // Realisierung der Assoziationen     <u>Wohnsitz hauptwohnsitz;</u>     <u>Wohnsitz[] nebenwohnsitze =</u>         <u>new Wohnsitz[10];</u> } // Klasse Mensch</pre> <p><b>3P</b></p>

Abbildung 163: Beispielhafte Frage und Antwort zum Thema „Assoziation“  
(vgl. [ES09a, Figure 13 und Figure 14 bezogen auf den Vergleichstest] und Nachhaltigkeitstest in Anhang O)

Evtl. können diese Schwierigkeiten bei den Themen „Arrays“ und „Assoziation“ wie folgt erklärt werden:

- Arrays sind deshalb schwer, weil nicht nur eine einzelne Variable von einem bestimmten Datentyp vorliegt, sondern gleich ein ganzes Feld von Variablen existiert (der Datentyp somit wesentlich komplexer ist) und

zusätzlich die einzelnen Feldelemente durch einen Index angesprochen werden müssen. Zusätzlich braucht man auch oft noch die Steuerstruktur Iteration, um auf den Feldern spezielle Algorithmen auszuführen.

- Assoziationen sind deshalb schwer, weil sie das Thema Attribute erweitern und z. B. eine 1:n-Assoziation mit Hilfe eines Feldes implementiert werden muss, bei dem sich ja die Schüler schwer tun.

Die Schwierigkeiten scheinen unabhängig davon zu sein, ob es um objektorientierte oder prozedurale Themen geht, da die beiden schwierigsten Themen aus beiden Themenbereichen kommen. Und da beide Klassen Schwierigkeiten mit den beiden Themen haben, sind diese auch außerhalb des Bereichs OOP-First bzw. OOP-Later, also unabhängig von der fachdidaktischen Vorgehensweise.

So scheint wirklich die Begründung wie folgt zu sein: Arrays werden unterrichtet in Kombination mit Schleifen um mit dem „Konzept Feld“ sinnvoll arbeiten zu können. Die häufig vorkommenden 1:n-Assoziationen können durch Felder implementiert werden. Auch dort muss das Feld oft mit Hilfe von Schleifen (Steuerstruktur Iteration) durchlaufen werden um mit dem „Konzept 1-n-Assoziation“ bzw. „Konzept Feld“ arbeiten zu können. Die Schwierigkeit ist also, nicht nur Programmier-Konstrukte, Daten- und Steuerstrukturen zu verstehen, sondern auch die dahinter stehenden Konzepte zu integrieren:

„The difficulty is to integrate concepts, not only language constructs but also data structure and algorithms. Hence the need arises

- to comprehend larger entities instead of details [MR02];
- to understand the whole instead of parts, which is more difficult according to the SOLS-taxonomy [LAua04], [LSua06];
- or to relate different parts of the program, which is more difficult than understanding entities only [Sch08].

In summary, the most difficult topics are not affected by changing from OOP-later to OOP-first or vice versa.” [ES09a, S.21, Quellenangaben wurden angepasst].

Es spricht vieles dafür (siehe Kapitel 7.5.4: „Gibt es Schlussfolgerungen für den Informatik-Anfangsunterricht?“), den Schülern mehr (Übungs-) Zeit beim Thema „Arrays“ zu geben und evtl. dafür das Thema „Assoziation“ aus dem Informatik-Anfangsunterricht herauszunehmen.

### 7.3.4 Kann OOM in OOP-Kursen vermittelt werden?

Hinweis: Dieses Unterkapitel ist fast identisch mit dem Kapitel 5.4 in dem Papier zur ICER’09 [ES09a, S. 21-22].

In einigen fachdidaktischen Diskussionen wird die Fähigkeit, ein Problem in ein objektorientiertes Design zu übersetzen bzw. die Fähigkeit ein objektorientiertes Modell zu entwerfen, als grundlegendes Teil der Einführung in das objektorientierte Paradigma gesehen (vgl. z. B.: [MWua05], [Chr05], [ZZ03] und [BS07]).

Ogleich die objektorientierte Modellierung (OOM) kein eigenständiges Thema bei den beiden fachdidaktischen Vorgehensweisen gewesen ist, fand es trotzdem Anwendung. Von Beginn der Einführung in die objektorientierte Programmierung wurden im Unterricht auch UML-Klassendiagramme und UML-Objektdiagramme verwendet, die immer mehr vertieft wurden. Z. B. wurden beim Thema „Vererbung“ Beispiele diskutiert, Vererbungsstrukturen mit Hilfe von UML-Klassendiagrammen entworfen und am Ende implementiert. Oder beim Thema „Assoziation“ wurden die Beziehungen zwischen Objekten (z. B. mit Hilfe von UML-Objektdiagrammen) diskutiert, auf Klassenebene abstrahiert bzw. generalisiert und am Ende implementiert.

Da die Einführung in die OOP bei der OOP-Later-Klasse fast ein halbes Jahr später als in der OOP-First-Klasse erfolgte, wurde die OOM in der OOP-Later-Klasse nur das letzte halbe Jahr unterrichtet. Die Summe der Modellierungen war aber in etwa gleich.

Bei den Modellierungen mussten die Schülerinnen und Schüler beider Schulklassen u. a. die folgenden Fragen beantworten:

- Welche Attribute und Methoden sollte die Klasse enthalten?
- Welche Beziehungen gibt es zwischen den Objekten und können diese generalisiert werden?
- Gibt es Generalisierungsmöglichkeiten von vergleichbaren Klassen im Hinblick auf eine Vererbungsstruktur?

Somit wurde die objektorientierte Modellierung „nebenbei“ in der OOP vermittelt. Um dies zu belegen bzw. nachzuprüfen wurde im Vergleichstest und im Nachhaltigkeitstest das Zusatzthema „OOM“ eingefügt (siehe Abbildung 164).

Nr.	Aufgabe
11.1	<p>Gegeben ist das folgende Szenario: <span style="float: right;"><b>10P</b></span></p> <p>Menschen können Fahrzeuge besitzen und Ehepartner haben, z. B.:</p> <p>Maria Müller besitzt ein konkretes Boot (Hersteller: C&amp;C), welches den Heimathafen in Berlin hat. Peter Meyer besitzt einen konkreten PKW (Hersteller: VW) mit 4 Sitzen und 95PS Leistung. Rainer Müller besitzt einen LKW (Hersteller: Volvo) mit 850PS und einer Ladekapazität von 40 Tonnen. Maria Müller ist mit Rainer Müller verheiratet.</p> <p><b>Modellieren</b> Sie mit Hilfe eines <b>UML-Klassendiagramms</b> (inkl. Vererbungen und Assoziationen) dieses Szenario, indem Sie die konkreten Daten versuchen zu generalisieren!</p>

Abbildung 164: Beispielhafte Frage zum Thema „OOM“

(vgl. [ES09a, Figure 15] und Nachhaltigkeitstest in Anhang O)

Die Lösung (siehe Abbildung 165) enthält sowohl Vererbungs- als auch Assoziationsstrukturen. Im Durchschnitt erreichten die Schülerinnen und Schüler in beiden Klassen bei dieser Aufgabe 5,8 von 10 möglichen Punkten. Dieses Thema belegt damit Platz 6 (von elf möglichen Plätzen) bei dem Schwere-Ranking der Themen (siehe Abbildung 157). Berücksichtigt man den Schwierigkeitsgrad der Aufgaben zu den einzelnen Themen, dann erreicht die OOM sogar den 2. Platz (von zwölf möglichen Plätzen, siehe Abbildung 158). Und die OOP-Later-Klasse schnitt (mit durchschnittlich 6,3 Punkten) deutlich besser ab als die OOP-First-Klasse (mit 5,3 Punkten).

Dies ist der größte Unterschied bei allen elf abgefragten Themen (siehe Abbildung 107). Der Unterschied ist aber nicht signifikant.

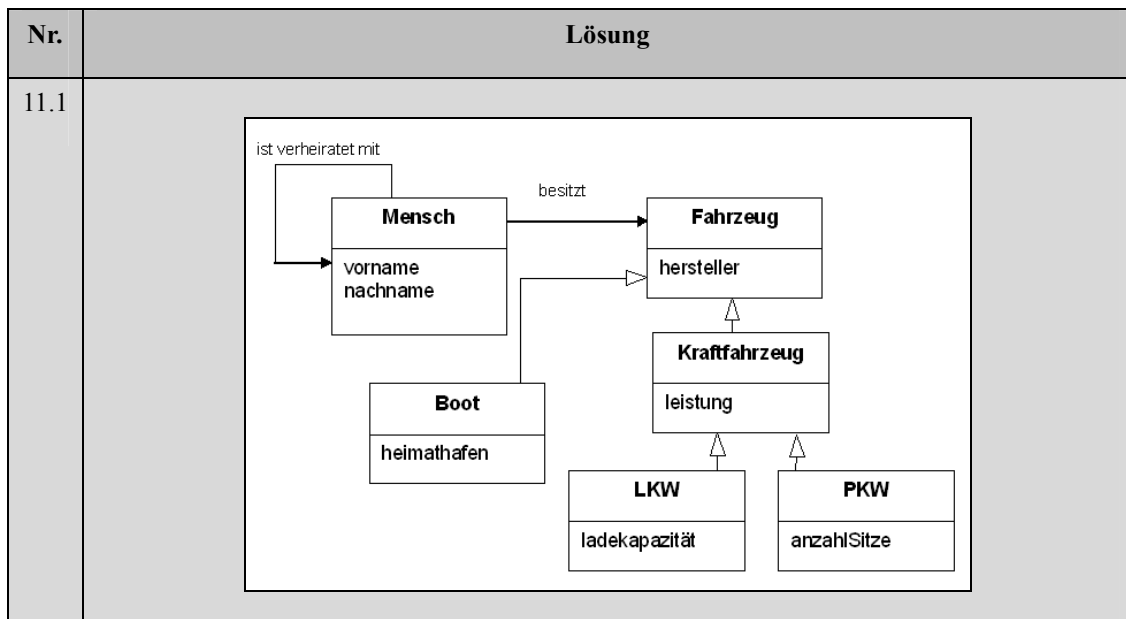


Abbildung 165: Beispielhafte Antwort zum Thema „OOM“

(vgl. [ES09a, Figure 16] und Nachhaltigkeitstest in Anhang O)

Dieses Resultat ist aus zweierlei Gründen interessant:

1. Es scheint so, dass die OOM in Kursen mit dem Schwerpunkt OOP vermittelt werden kann. Oder vorsichtiger ausgedrückt: Das Lernen der objektorientierten Modellierung (OOM) scheint nicht durch die objektorientierte Programmierung (OOP) behindert zu werden, weder von einem OOP-First-Vorgehen noch von einem OOP-Later-Vorgehen.
2. Es wird ein neues Licht auf die Debatte um den Umstieg von einem prozeduralen Design auf ein objektorientiertes Design geworfen. Die Ergebnisse legen nahe, dass OOM auch in OOP-Later-Kursen gut vermittelt werden kann (mit der Tendenz sogar zu einer besseren Nachhaltigkeit als in OOP-First-Kursen).

Damit fällt m. E. ein Argument aus der fachdidaktischen Diskussion heraus (oder bekommt weniger Gewicht), warum ein „objects-first-Vorgehen“ gewählt werden sollte.

## 7.4 Das subjektive Erleben

In Kapitel 6.4 wurde ausführlich das emotionale, kognitive und motivationale Erleben der Schülerinnen und Schüler innerhalb des Schuljahres in Abhängigkeit vom fachdidaktischen Vorgehen dokumentiert und analysiert.

Es sollen nun die folgenden zusammenfassenden bzw. vertiefenden fachdidaktischen Fragestellungen beantwortet werden:

- Hat das subjektive Erleben Einfluss auf den Lernerfolg (siehe Kapitel 7.4.1)?
- Welche Themen werden als leicht, welchen Themen als schwer empfunden (siehe Kapitel 7.4.2)?

- Hat das Schwere-Empfinden etwas mit der tatsächlichen Schwere zu tun bzw. Auswirkungen auf den Lernerfolg (siehe Kapitel 7.4.3)?
- Gibt es bei der Untersuchung des Schwere-Empfindens Berührungspunkte zu anderen empirischen Untersuchungen (siehe Kapitel 7.4.4)?
- Erleben die OOP-Later-Schüler den Unterricht wirklich subjektiv angenehmer (siehe Kapitel 7.4.6)?

### 7.4.1 Hat das subjektive Erleben Einfluss auf den Lernerfolg?

Im Kapitel 6.4.5 wurde der Verlauf des subjektiven Erlebens dargestellt und erläutert. Beide fachdidaktische Vorgehen erzeugen danach ein positives subjektives Erleben (siehe Abbildung 166).

	OOP-First	OOP-Later
<b>Fach Informatik</b>	<b>2,56*</b> p=0,0003	<b>2,31*</b> d=2,52
<b>Thema</b>	<b>2,40*</b> p=0,026	<b>2,22*</b> d=1,16

Abbildung 166: Subjektives Erleben im Kontext zu zwei speziellen Frage-Bereichen

(vgl. Abbildung 115 und Abbildung 138, \* signifikante Ergebnisse sind fett gedruckt)

Die Werte der OOP-Later-Klasse sind dabei in allen Bereichen besser als die der OOP-First-Klasse. Bezogen auf das Fach und das Thema lassen sich die folgenden Aussagen treffen (vgl. Kapitel 6.4.5):

- Die Unterschiede im subjektiven Erleben bezogen auf das Fach (OOP-First: 2,56 / OOP-Later: 2,31) sind höchst signifikant ( $p = 0,0003$ ). Die Effektstärke ( $d = 2,52$ ) liegt deutlich über dem Wert 0,8, der Mittelwertsunterschied stellt somit einen sehr starken Effekt dar (vgl. [Coh88, S. 20-27]).
- Die Unterschiede im subjektiven Erleben bezogen auf die Themen (OOP-First: 2,40 / OOP-Later: 2,22) sind signifikant ( $p = 0,026$ ). Die Effektstärke ( $d = 1,16$ ) liegt über dem Wert 0,8, der Mittelwertsunterschied stellt somit einen relativ starken Effekt dar.

Das subjektive Erleben der Themen korreliert allerdings wenig (OOP-First:  $r = 0,25$  / OOP-Later:  $r = 0,55$ ) mit den Ergebnissen des Nachhaltigkeitstests (vgl. Abbildung 169 und Abbildung 170). Dies ist die (kurze) Antwort auf die fachdidaktische Fragestellung, ob das subjektive Erleben Einfluss auf den Lernerfolg hat.

Das bedeutet aber nicht, dass das subjektive Erleben für den Unterricht bzw. für den Lernerfolg nicht wichtig ist. Vielleicht wäre der Lernerfolg ja schlechter gewesen, wenn das subjektive Erleben schlechter gewesen wäre. In Kapitel 2.3 wurden einige Quellen (z. B. [EH08], [Mül08], [Sei04], [Sem92], [Sem05], [Ven04] und [WS01]) genannt, die einen Zusammenhang zwischen „Emotion“ bzw. „emotionaler Befindlichkeit“ bzw. „individuellem Wohlbefinden“ bzw. „Comfort Level“ und dem Unterrichtserfolg herstellen. Das subjektive Erleben hat sicher auch etwas mit den Interessen der beteiligten Schüler und ihren Grundeinstellungen zum Themengebiet Informatik zu tun. In der Studie konnte nicht herausgearbeitet werden, welche Schüler (z. B. aus mangelndem Interesse oder Überforderung) den Informatik-Anfangskurs freiwillig verlassen hätten, da sie das Fach Informatik am OSZ IMT nicht abwählen können ohne auch gleichzeitig die Schule verlassen zu müssen [ES10a, S. 112].

## 7.4.2 Welche Themen werden als einfach, welche Themen als schwierig empfunden?

In Kapitel 6.4.3 wurden schon die Antworten auf die Frage nach der „subjektiven Schwierigkeit“ der einzelnen Themen näher untersucht. Die Abbildung 167 zeigt die Ergebnisse, geordnet nach den einzelnen Themen.

Nr.	Thema	Schwere OOP-First	Schwere OOP-Later
T1	Einführung in die OOP	2,5	1,9
T2	Datentypen / Steuerstruktur Sequenz	2,8	2,3
T3	Klasse und Objekt	2,9	2,7
T4	Methoden	2,6	2,6
T5	Steuerstruktur Selektion	2,8	2,4
T6	Steuerstruktur Iteration	2,9	2,5
T7	Arrays und Strings	<b>3,1*</b>	<b>2,5*</b>
T8	Vererbung	2,4	2,9
T9	Assoziation	2,7	3,1
	<b>Durchschnitt</b>	2,74	2,54

Abbildung 167: Schwierigkeits-Empfinden in Abhängigkeit vom Thema

(vgl. Abbildung 129, \* signifikante Ergebnis-Unterschiede sind fett gedruckt)

Es wurde festgestellt, dass die Differenz zwischen den Mittelwerten der beiden Klassen (0,2) nicht signifikant ist ( $p = 0,17$ ). Der große Unterschied (0,6) zwischen den beiden Klassen beim Thema 1 ist knapp nicht signifikant ( $p = 0,054$ ). Der Unterschied (0,5) beim Thema 2 ist auch nicht signifikant ( $p = 0,19$ ). Der große Unterschied (0,6) zwischen den beiden Klassen beim Thema 7 ist dagegen signifikant ( $p = 0,03$ ). Die Effektstärke ( $d = 0,71$ ) liegt aber unter dem Wert 0,8, der Mittelwertsunterschied stellt somit nur einen mittelstarken Effekt dar (vgl. [Coh88, S. 20-27]).

Im Folgenden soll nun die „empfundene Schwierigkeit“ pro Thema in einer Rangfolge dargestellt werden (siehe Abbildung 168).

Nr.	OOP-First	Schwere
T8	Vererbung	2,4
T1	Einführung in die OOP	2,5
T4	Methoden	2,6
T9	Assoziation	2,7
T2	Datentypen / Steuerstruktur Sequenz	2,8
T5	Steuerstruktur Selektion	2,8
T3	Klasse und Objekt	2,9
T6	Steuerstruktur Iteration	2,9
T7	Arrays und Strings	3,1
	<b>Durchschnitt</b>	<b>2,74</b>

Nr.	OOP-Later	Schwere
T1	Einführung in die OOP	1,9
T2	Datentypen / Steuerstruktur Sequenz	2,3
T5	Steuerstruktur Selektion	2,4
T7	Arrays und Strings	2,5
T6	Steuerstruktur Iteration	2,5
T4	Methoden	2,6
T3	Klasse und Objekt	2,7
T8	Vererbung	2,9
T9	Assoziation	3,1
	<b>Durchschnitt</b>	<b>2,54</b>

Abbildung 168: Rangfolge der „empfundene Schwierigkeit“ der Themen



Folgende Aussagen können dann getroffen werden (siehe Abbildung 168)<sup>8</sup>:

- Beide Klassen empfinden die „Einführung in die OOP“ mit dem BlueJ-Tool als einfach.
- Für die OOP-Later-Klasse ist das „zweiteinfachste Thema“ das Thema 2 („IDE und Datentypen“), mit diesem wurde der Unterricht im Schuljahr begonnen.
- Die OOP-Later-Klasse empfindet die beiden Themen 8 („Vererbung“) und 9 („Assoziation“) als schwierigste Themen. Diese wurden zuletzt unterrichtet und sind beide objektorientierte Themen.
- Die OOP-First-Klasse empfindet die Themen 7, 6 und 3 als schwierigste Themen. Für einen Teil des Themas 7 („Arrays“) benötigen die Schülerinnen und Schüler das Thema 6 („Steuerstruktur Iteration“). Das Thema 3 („Klasse und Objekt“) ist auch bei der OOP-Later-Klasse auf dem dritthöchsten „Schwere-Platz“.
- Für die OOP-First-Klasse ist das einfachste Thema das Thema 8 („Vererbung“), ganz im Gegensatz zum Empfinden der OOP-Later-Klasse.
- Drei Themen werden sehr unterschiedlich erlebt. Das Thema 8 („Vererbung“) ist um sieben Rangplätze und die Themen T7 („Arrays und Strings“) und T9 („Assoziation“) sind um fünf Rangplätze verschoben.
- Interessant ist, dass beide Klassen das Thema „Klasse und Objekt“ als relativ schwierig erleben. Die Schüler scheinen sich schwer zu tun mit der Unterscheidung von einer Idee bzw. Modellierung einer Klasse (als Bauplan für spätere Objekte) und der Erzeugung von Objekten.
- Vergleicht man auch hier die Werte von so genannten objektorientierten Themen (OOP-First / OOP-Later = 2,63 / 2,65) mit denen von so genannten prozeduralen Themen (OOP-First / OOP-Later = 2,78 / 2,45), dann erleben die beiden Klassen die objektorientierten Themen gleich schwierig, aber die OOP-Later-Klasse empfindet die prozeduralen Themen als deutlich einfacher.<sup>9</sup>

### 7.4.3 Hat das Schwierigkeits-Empfinden etwas mit dem tatsächlichen Schwierigkeitsgrad zu tun?

Im Folgenden soll die „empfundene Schwierigkeit“ der einzelnen Themen mit den Ergebnissen des Nachhaltigkeitstests verglichen werden, z. B. ob es eine Korrelation zwischen beiden Ergebnissen gibt.

Für die Tabelle in Abbildung 169 werden die Ergebnisse des Nachhaltigkeitstests (in Prozent) in die Item-Skala (Werte von 1 bis 5) transformiert. Dies geschieht durch die Formel:

$$W_{\text{Itemskala}} = 5 - (4 * W_{\text{Prozent}} / 100)$$

Daraus ergibt sich der folgende Sachverhalt: 100% Leistung ergibt die Zahl 1 auf der Itemskala, 50% Leistung ergibt den Wert 3 und 0% Leistung die Zahl 5. Der Wert 50% war im Sommer 2008 (siehe Notenschlüssel in

---

<sup>8</sup> Die farbliche Kennzeichnung in Abbildung 168 ist wie folgt: Wenn man die Extremgruppen-Definition (jeweils 25% der Untersuchungsteilnehmer) von Bortz und Döring [BD02, S. 219] zur Grundlage einer Entscheidung für eine Extremwert-Definition (jeweils 25% der Themen) nimmt, können z. B. die Themen mit den zwei größten Werten und die Themen mit den zwei kleinsten Werten (25% \* 9 Themen = 2,25 Themen) als schwerste (rote) Themen und leichteste (grüne) Themen dargestellt werden.

<sup>9</sup> Das Thema „Arrays und Strings“ wurde bei dem Vergleich nicht berücksichtigt, da es sowohl eine prozedurale als auch eine objektorientierte Komponente enthält.

Abbildung 153) der Grenzwert für die Aussage, ob ein Test bestanden wurde oder nicht. Ab 50% Leistung sind die Schulleistungen ausreichend (Note 4). Dies bedeutet für die Transformation in die Itemskala, dass Werte größer 3 mangelhafte Leistungen sind, also die entsprechende Aufgabe nicht bestanden wurde [ES10a, S. 111].

Nr.	Thema	Schwere OOP-First	Schwere OOP-Later	Schwere Ø	N-Test OOP-First	N-Test OOP-Later	N-Test Ø
T1	Einführung in die OOP	2,5	1,9	2,20	2,3**	2,4**	2,35**
T3	Klasse und Objekt	2,9	2,7	2,80			
T2	Datentypen / Steuerstruktur Sequenz	2,8	2,3	2,55	2,4	2,1	2,25
T4	Methoden	2,6	2,6	2,60	2,9	3,2	3,05
T5	Steuerstruktur Selektion	2,8	2,4	2,60	2,4	2,0	2,20
T6	Steuerstruktur Iteration	2,9	2,5	2,70	2,7	2,8	2,75
T7	Arrays und Strings	<b>3,1*</b>	<b>2,5*</b>	2,80	3,6	3,5	3,55
T8	Vererbung	2,4	2,9	2,65	2,7	2,6	2,65
T9	Assoziation	2,7	3,1	2,90	4,0	4,4	4,20
	<b>Durchschnitt</b>	2,74	2,54	2,64	2,81	2,82	2,815

Abbildung 169: Ergebnisse des Nachhaltigkeitstests und Werte der „empfundenen Schwierigkeit“

(\* signifikante Ergebnis-Unterschiede sind fett gedruckt, \*\* Thema T1/T3 wird doppelt gewertet)

Folgende Aussagen können getroffen werden (siehe Abbildung 169):

- Die Mittelwerte für den Nachhaltigkeitstest sind in den beiden Klassen (fast) gleich.
- Die Mittelwerte für die „empfundene Schwierigkeit“ in den beiden Klassen unterscheiden sich um den Wert 0,2. Diese Differenz ist nicht signifikant (siehe Kapitel 6.4.3).
- Der Mittelwert der „empfundenen Schwierigkeit“ für beide Klassen zusammen (Wert: 2,64) und der Mittelwert beider Klassen bezogen auf den Nachhaltigkeitstest (Wert: 2,815) bewegen sich auf ähnlichem Niveau. Das Schwierigkeits-Empfinden könnte somit ein guter Indikator sein für das Ergebnis einer späteren Leistungs-Überprüfung. Wenn man aber nicht den Mittelwert, sondern alle Ergebnisse zu den einzelnen Themen betrachtet, dann gilt dies nur für die OOP-Later-Klasse (s. u.).
- Bei den Themen 7 („Arrays und Strings“) und 9 („Assoziation“) ist die Abweichung zwischen der „empfundenen Schwierigkeit“ und den späteren Test-Ergebnissen am größten (Thema 7:  $3,55 - 2,80 = 0,75$  / Thema 9:  $4,20 - 2,90 = 1,3$ ). Immerhin sind für die OOP-First-Klasse das Thema 7 und für die OOP-Later-Klasse das Thema 9 die Themen mit der höchsten „empfundenen Schwierigkeit“. Sie liegen in diesem Fall mit ihrer Einschätzung richtig.
- Die OOP-Later-Klasse schätzt insbesondere das Thema 7 („Arrays und Strings“) als viel zu einfach ein (Wert: 2,5). Sie unterschätzt das Thema noch viel mehr als die OOP-First-Klasse (Wert: 3,1). Dieser Unterschied ist der einzige Unterschied bei allen Themen, der signifikant ist ( $p = 0,03$ ), wobei die Effektstärke ( $d = 0,71$ ) unter dem Wert 0,8 liegt, der Mittelwertsunterschied somit nur einen mittelstarken Effekt darstellt (vgl. [Coh88, S. 20-27]).
- Es gibt keine (gute) Korrelation bzw. nur eine kleine Korrelation zwischen der „empfundenen Schwierigkeit“ in der OOP-First-Klasse und den Ergebnissen im Nachhaltigkeitstest (Korrelationskoeffizient = 0,23).

- Es gibt eine (gute) Korrelation zwischen der „empfundenen Schwierigkeit“ in der OOP-Later-Klasse und den Ergebnissen im Nachhaltigkeitstest (Korrelationskoeffizient = 0,61).

Die letzten beiden Punkte sind eine Antwort auf die Frage, ob das subjektive Schwierigkeits-Empfinden etwas mit dem tatsächlichen Schwierigkeitsgrad zu tun hat. Die (kurze) Antwort richtet sich nach der fachdidaktischen Vorgehensweise: In der OOP-First-Klasse lautet sie eher „Nein“ und in der OOP-Later-Klasse eher „Ja“. Vielleicht hatte der Nachhaltigkeitstest mehr mit der Unterrichtswirklichkeit der OOP-Later-Klasse als mit der Unterrichtswirklichkeit der OOP-First-Klasse zu tun?

Durch die Abbildung 170 sollen die Ergebnisse des Nachhaltigkeitstests besser visualisiert werden. Für das Thema 1 mit der Anwendung des BlueJ-Tools gab es keinen speziellen Aufgabenbereich. Die „Einführung in die OOP“ war beim Nachhaltigkeitstest dem Thema 3 („Klasse und Objekt“) hinzugefügt, so dass bei diesem Thema auch die doppelte Punktzahl im Vergleich zu den anderen Themen erzielt werden konnte. Dies ist der Grund für die zweifache Wertung des Ergebnisses für Thema 1 und Thema 3 in Abbildung 169.

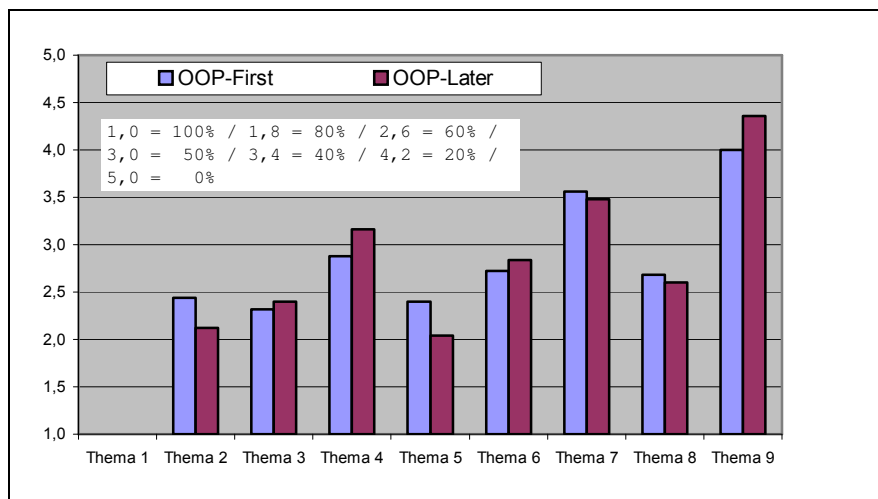


Abbildung 170: Ergebnisse des Nachhaltigkeitstests dargestellt mit Hilfe der „Item-Skala“ [ES10a, Figure 11, S. 111]

Folgende Aussagen (siehe Abbildung 170) können getroffen werden [ES10a, S. 111]:

- Die Mittelwerte für den Nachhaltigkeitstest sind fast gleich (OOP-First:  $2,81 \pm 54,67\%$  / OOP-Later:  $2,82 \pm 54,44\%$ ). Das Testergebnis insgesamt war im Hinblick auf die Noten somit ausreichend.
- Zwei Themen sind nicht mit Erfolg bearbeitet worden: Thema 7 („Arrays und Strings“) und Thema 9 („Assoziation“). Eine Interpretation dieses Sachverhalts wurde in Kapitel 7.3.1 gegeben.
- Das Thema 4 („Methoden“) ist ein „Grenzthema“: Die OOP-First-Klasse hat die Aufgaben zu diesem Thema knapp bestanden (Wert: 2,9), während die OOP-Later-Klasse knapp durchgefallen ist (Wert: 3,2).

#### 7.4.4 Gibt es Berührungspunkte zu vergleichbaren Untersuchungen?

In diesem Unterkapitel sollen die Ergebnisse zur „empfundenen und tatsächlichen Schwere“ der einzelnen Themen dieser Studie in einen Kontext zu anderen empirischen Untersuchungen gestellt werden, wobei sich diese oft auf Studenten (und nicht auf Schüler) beziehen. Der Abschnitt stammt dabei überwiegend aus einem Papier zur „International Computing Education Research 2009 (ICER’09)“ [ES09a].

Lahtinen, Ala-Mutka und Järvinen [LAJ05] baten Studenten, welche einen prozeduralen Anfänger-Programmierkurs (CS1) in Java oder C++ belegt hatten, ihre gefühlten Schwierigkeiten bei den einzelnen Programm-Konstrukten zu beschreiben. Themen, die auch in dieser Studie unterrichtet wurden, wurden (von leicht zu schwer) wie folgt genannt (siehe Abbildung 171 und [LAJ05, S. 16, Table 1]): Selektion, Iteration, Variablen, Parameter und Felder. Sie [LAJ05, S. 15] benutzten dabei die gleichen Skalenwerte von 1 („very easy to learn“) bis 5 („very difficult“) wie die in dieser Studie (siehe Kapitel 5.1.4), so dass in der Abbildung 171 einerseits die Rangfolge der Themen der „LAJ-Untersuchung“ als auch die entsprechenden „gefühlten Werte“ und realen Werte des Nachhaltigkeitstests nebeneinander aufgeführt werden können. Die Rangfolgen unterscheiden sich dabei in der Regel, aber für die letzten beiden Themen („Parameter (Methoden)“ und „Felder“) ergibt sich (mit der Ausnahme der „empfundenen Schwere“ der OOP-Later-Klasse) ein fast einheitliches Bild: Das Thema „Parameter (Methoden)“ ist das zweitschwerste Thema, das Thema „Felder“ das schwerste Thema.

Thema	[LAJ05] 1..5	Schwere OOP-First	Schwere OOP-Later	N-Test OOP-First	N-Test OOP-Later
Selektion	1,98	2,8	2,4	2,4	2,0
Iteration	2,09	2,9	2,5	2,7	2,8
Variablen (Datentypen)	2,10	2,8	2,3	2,4	2,1
Parameter (Methoden)	2,60	2,6	2,6	2,9	3,2
Felder	2,79	3,1	2,5	3,6	3,5
<b>Durchschnitt</b>	2,31	2,84	2,46	2,80	2,72

Abbildung 171: „Empfundene und tatsächliche Schwere“ der Themen im Kontext zur „LAJ-Untersuchung“ (vgl. [LAJ05, S. 16, Table 1], Abbildung 168 und Abbildung 169)

Milne und Rowe [MR02] fragten ihre Studenten auch nach der „empfundenen Schwere“ der Themen, allerdings erst nach zwei Jahren Informatik-Anfangsunterricht: einem Jahr mit prozeduraler Programmierung in C und einem weiteren Jahr mit objektorientierter Programmierung in C++. Die Umrechnung der Werte ihrer Werteskala von 1 bis 7 auf die Werteskala von 1 bis 5 erfolgt nach der folgenden Formel:

$$W_5 = (((W_7 - 1) / 6) * 4) + 1$$

Die Rangfolge der Themen der „MR-Untersuchung“ ist fast gleich mit den beiden Rangfolgen der „empfundenen Schwere“ dieser Studie: Bei der OOP-First-Klasse gibt es nur den Ausreißer „Vererbung“ und bei der OOP-Later-Klasse nur den Ausreißer „Arrays und Strings“.

Thema	[MR02] 1..7	[MR02] 1..5	Schwere OOP-First	Schwere OOP-Later	N-Test OOP-First	N-Test OOP-Later
Variable	1,885	1,59	2,8	2,3	2,4	2,1
Selektion	2,040	1,69	2,8	2,4	2,4	2,0
Iteration	2,346	1,90	2,9	2,5	2,7	2,8
Klasse und Objekt	3,000	2,33	2,9	2,7	2,3	2,4
Arrays und Strings	3,308*	2,54	3,1	2,5	3,6	3,5
Vererbung	3,423	2,62	2,4	2,9	2,7	2,6
<b>Durchschnitt</b>	2,67	2,11	2,82	2,55	2,68	2,57

Abbildung 172: „Empfundene und tatsächliche Schwere“ der Themen im Kontext zur „MR-Untersuchung“ (vgl. [MR02, S. 58, Table 2], Abbildung 168 und Abbildung 169; \* Zahl wurde gemittelt aus 2 Themen)

Die Werte der „MR-Untersuchung“ liegen deutlich unter den Werten dieser Studie. Die Studenten hatten ein geringeres Schwere-Empfinden im Hinblick auf die einzelnen Themen als die Schüler des OSZ IMT.

Allgemein kann man sagen, dass es schwer ist, die „empfundene Schwere“ über verschiedene Studien hinweg zu vergleichen. In vielen anderen Studien wird die Meinung der Studenten mit der Meinung der Professoren verglichen. In dieser Studie wird hingegen die Meinung der Schülerinnen und Schüler mit den Ergebnissen des Post- und Follow-Up-Tests verglichen. So oder so treten Unterschiede zu Tage. Die OOP-Later-Klasse fand z. B. das Thema 7 („Arrays und Strings“) nicht besonderes schwer (2,5), aber der Nachhaltigkeitstest ergab einen relativ schlechten Wert (3,5). Die OOP-First-Klasse unterschätzte die Schwere des Themas 9 („Assoziation“) genauso wie die OOP-Later-Klasse. Der Wert des Nachhaltigkeitstests liegt mit 4,0 deutlich über dem Wert der „empfundenen Schwere“ (2,7). Vielleicht ist die „empfundene Schwere“ mehr fokussiert auf den Lern-Prozess, während die Testergebnisse mehr auf das Prozess-Resultat fokussiert sind. Und evtl. sind auch bei den anderen Studien die Meinung der lehrenden Professoren mehr bezogen auf das Lern-Ergebnis als auf den Lern-Prozess.

Die „empfundene Schwere“ darf nicht verwechselt werden mit dem „Comfort Level“ der Untersuchung von Wilson und Shrock [WS01]. Die beiden Autoren fanden heraus, dass nur drei von zwölf ausgewählten Faktoren signifikant Einfluss nehmen auf den Erfolg der Teilnehmer eines Informatik-Anfangskurses, davon ist der Faktor „Comfort Level“ der wichtigste Faktor. Diesen Faktor ermittelten sie durch insgesamt sieben Fragen (siehe Originalzitat in Kapitel 2.3), z. B. Angstgefühle während der Computerarbeit („anxiety level“), gefühlte Schwere („perceived difficulty“) des Kurses und der Programmieraufgaben und ihr Verständnis („perceived understanding“) der Themen bzw. Konzepte auch im Vergleich zu ihren Mitstudenten.

Es ist schwer, die Ergebnisse dieser Studie mit der „WS-Untersuchung“ zu vergleichen, obwohl einige Fragen bestimmt in die Richtung „Comfort Level“ gehen, z. B. die Frage 11, die nach den Anforderungen des Faches Informatik fragt und die Frage 12, die nach der Schwere des Stoffs des letzten Themas fragt.

#### 7.4.5 Warum sind die Themen bei OOP-Later besser vernetzt?

In Kapitel 6.4.3c wurden u. a. die folgenden zwei Fragen bzw. Items betrachtet:

Frage 24: Der Wiedererkennungswert zu bereits behandelten **Themen** war ...  
(sehr hoch / sehr niedrig)

Frage 25: Die Nützlichkeit des Wissens bereits behandelter Themen für das **letzte Thema** war ...  
(sehr hoch / sehr niedrig)

Beide zusammen kann man als eine Art Vernetzungsfaktor der einzelnen Themen betrachten, also die Anwendbarkeit bestehenden Wissens (in Form von Wiedererkennungsgrad und Nützlichkeit schon behandelter Themen) auf das aktuell unterrichtete Thema.

Für die einzelnen Themen ergeben sich die Werte (vgl. Abbildung 127 und Abbildung 128) aus Abbildung 173. Die Differenzen bei den einzelnen Themen sind höchst signifikant ( $p = 0,005$ ), die Irrtumswahrscheinlichkeit liegt nur bei 0,5%. Die Effektstärke ( $d = 1,66$ ) liegt über dem Wert 0,8. Der Mittelwertsunterschied stellt somit einen starken Effekt dar (vgl. Kapitel 6.4.3c).

Nr.	Thema	Vernetzung	
		OOP-First	OOP-Later
T1	Einführung in die OOP	*	2,2
T2	Datentypen / Steuerstruktur Sequenz	2,8	*
T3	Klasse und Objekt	2,3	2,4
T4	Methoden	2,3	2,1
T5	Steuerstruktur Selektion	2,3	2,3
T6	Steuerstruktur Iteration	2,5	2,2
T7	Arrays und Strings	2,5	2,2
T8	Vererbung	2,4	2,1
T9	Assoziation	2,5	1,8
	<b>Durchschnitt</b>	2,4	2,1

Abbildung 173: Vernetzung der einzelnen Themen mit vorher schon unterrichteten Themen  
(vgl. Abbildung 127 und Abbildung 128, \* das erste Thema hat keine vorherigen Themen)

Somit kann die folgende Aussage formuliert werden:

- Beim OOP-Later-Vorgehen wird mehr auf dem Vorwissen aufgebaut als beim OOP-First-Vorgehen; die Vernetzung der Themen ist größer.

Diese Erkenntnis ist wie folgt erklärbar:

- Gerade das OOP-Later-Vorgehen orientiert sich am Anfang an der klassischen prozeduralen Vorgehensweise, bei der ein Thema auf dem anderen aufbaut.
- Evtl. ist das OOP-First-Vorgehen nicht ganz so sequenziell, wie durch das Modell dargestellt, sondern mehr integrativ bzw. spiralförmig. Gerade am Anfang der objektorientierten Programmierung (und damit auch am Anfang des Informatik-Anfangskurses) werden innerhalb der „Einführung in die OOP“ Themen angewendet (z. B. Parameter inkl. Datentypen, Methodenaufrufe etc.), die erst später explizit unterrichtet werden.

#### 7.4.6 Erleben die OOP-Later-Schüler den Unterricht wirklich subjektiv angenehmer?

Bei der Ergebnisübersicht zum subjektiven Erleben (siehe Kapitel 6.4.1 bzw. Abbildung 115 und Abbildung 138) wurde deutlich, dass das subjektive Erleben der OOP-Later-Klasse (oft signifikant) besser ist als das der OOP-First-Klasse. Die Ergebnisse aus der Abbildung 115 werden nun in der Abbildung 174 verkürzt dargestellt bzw. die Signifikanzberechnungen aus Abbildung 138 hinzugenommen.

Folgende Aussagen können getroffen werden (siehe Abbildung 174):

- Beide fachdidaktische Vorgehen erzeugen ein positives subjektives Erleben, da alle Zahlenwerte kleiner als die Zahl 3 sind.
- In allen sieben Teilkategorien (aufgeschlüsselt in vier Frage-Bereiche und drei Erlebnis-Dimensionen) erzielt die OOP-Later-Klasse bessere Ergebnisse, durchschnittlich um den Faktor 0,26.

- Das durchschnittliche subjektive Erleben bezogen auf die vier Frage-Bereiche ist immer signifikant unterschiedlich. Bei den Durchschnittsberechnungen bezogen auf die drei Erlebnis-Dimensionen wurde auf eine Signifikanzberechnung verzichtet.

Sollte man wegen dieser Unterschiede nun ein OOP-Later-Vorgehen wählen? Eine einfache Erklärung für die Unterschiede im Bereich „Thema“ und im Bereich „Fach Informatik“ könnte sein, dass die beiden unterschiedlich erlebten Lehrer die Ursache hierfür sind, die Unterschiede also nichts mit dem fachdidaktischen Vorgehen zu tun haben. Dann stellt sich aber die Frage, warum ein einzelner Informatik-Lehrer gleich das subjektive Erleben im ganzen Bereich „Schule“ beeinflussen sollte. Vielleicht hat die OOP-First-Klasse andere negative Einflussfaktoren, z. B. in den anderen Fächern schlechtere Lehrer oder einen schlechteren Stundenplan oder eine schlechtere „Klassenatmosphäre“: Dies könnten die Ursachen für die Werte-Differenz der beiden Klassen im Bereich „Schule“ sein [ES10a, S.112].

Subjektives Erleben	Durchschnitt über alle 3 Erlebnis-Dimensionen		Subjektives Erleben	Durchschnitt*** über alle 4 Frage-Bereiche	
	OOP-First	OOP-Later		OOP-First	OOP-Later
Schule	2,86* p=0,00006	2,60* d=2,56	Emotionales Erleben	2,6	2,3
Fach Informatik	2,56* p=0,0003	2,31* d=2,52	Kognitives Erleben	2,8	2,6
Lehrer	2,31* p=0,0005	1,94* d=2,05	Motivationales Erleben	2,3	2,0
Thema	2,40* p=0,026	2,22* d=1,16	Durchschnitt	2,57 (2,53**)	2,30 (2,27**)
Durchschnitt***	2,53	2,27			

Abbildung 174: Subjektives Erleben im Kontext zu den Erlebnis-Dimensionen und zu den Frage-Bereichen (vgl. Abbildung 115 und Abbildung 138, \* signifikante Ergebnis-Unterschiede sind fett gedruckt, \*\* tatsächlicher Durchschnitt, \*\*\* ohne Signifikanzberechnung)

Um diese ganzen Widersprüche besser analysieren zu können, soll jetzt der Fokus auf die Antworten zur Frage 1 gelenkt werden. Diese Frage war bisher ausgeklammert worden. Sie lautete:

Frage 1: Mir geht es zur Zeit in meinem **privaten Umfeld** eher ...  
(sehr gut / sehr schlecht)

Diese Frage sollte als Kontrolle für die Güte der ermittelten Daten dienen. Erwartet wurde für beide Klassen ein konstanter Verlauf (auf gleicher Höhe), der evtl. kleine Senken enthält, z. B. durch gerade beendete Ferien, in denen evtl. das private Umfeld verbessert werden konnte. Den zeitlichen Verlauf zum subjektiven Erleben des privaten Umfelds zeigt die Abbildung 175.

Folgende Aussagen können getroffen werden:

- Das private Umfeld wird von beiden Klassen immer im positiven Bereich gesehen.

- Der Durchschnittswert der OOP-First-Klasse liegt mit 2,73 höher als der Durchschnittswert der OOP-Later-Klasse mit 2,16. Die Differenz von 0,57 ist höchst signifikant ( $p = 0,00000056$ ). Die Effektstärke ( $d = 3,95$ ) liegt deutlich über dem Wert 0,8, der Mittelwertsunterschied stellt somit einen sehr starken Effekt dar (vgl. [Coh88, S. 20-27]).

Man sollte somit vorsichtig sein mit den Werten des subjektiven Erlebens in den Bereichen „Fach Informatik“, „Lehrer“ und „Themen“, da die OOP-Later-Klasse (auch) im Bereich „Schule“ und im Bereich „privates Umfeld“ deutlich bessere Werte besitzt. In diesen beiden Bereichen hätte man vergleichbare Werte in den beiden Klassen erwartet. Ein spezielles fachdidaktisches Vorgehen in einem speziellen Schulfach wird m. E. nicht gleich die Schule oder das private Umfeld „subjektiv besser“ machen [ES10a, S. 112].

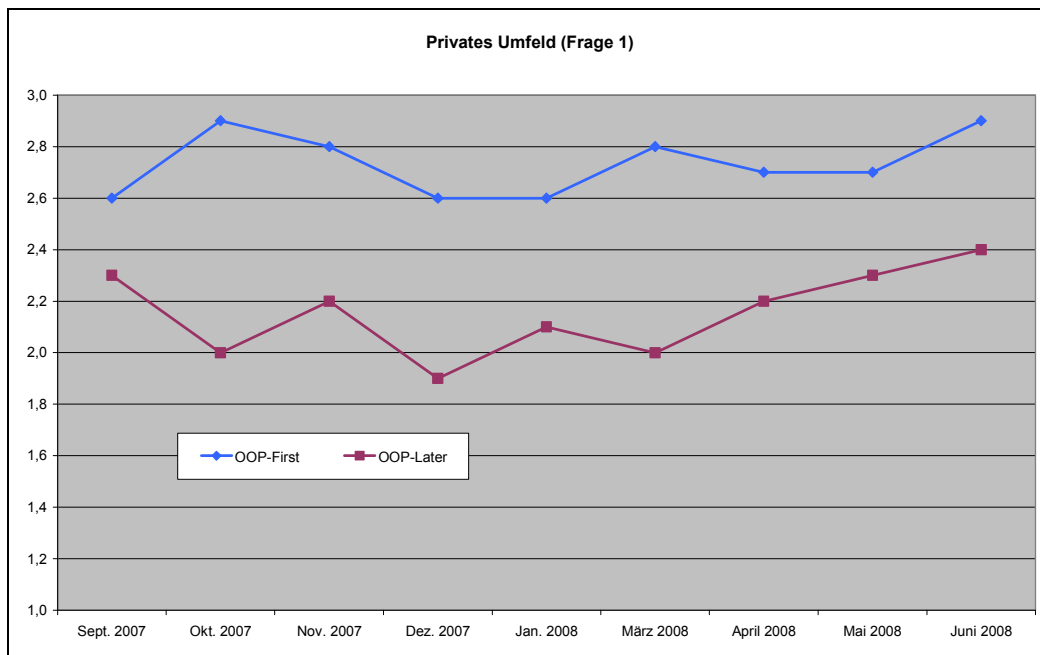


Abbildung 175: Subjektives Erleben des privaten Umfelds

Folgende Interpretationen für diese Widersprüche sind denkbar:

- Die OOP-First-Schüler neigen dazu, die Skalenwerte anders zu betrachten als die OOP-Later-Schüler und eher etwas schlechtere Werte anzukreuzen. Oder sie sind generell pessimistischer [ES10a, S. 112]. Oder die OOP-Later-Schüler neigen dazu, die eher etwas besseren Werte anzukreuzen. Oder sie sind generell optimistischer. Dann müssten die Werte mit Hilfe eines Korrekturfaktors vergleichbar gemacht werden, der irgendwo zwischen 0,57 (Differenz im Bereich „privates Umfeld“) und 0,26 (Differenz im Bereich „Schule“) liegen müsste. Allein der Faktor 0,26 würde ausreichen, die subjektiven Erlebnisunterschiede im Mittelwert zu Null zu machen (siehe Abbildung 174). Bei höheren Korrekturwerten würden plötzlich die OOP-First-Schüler das bessere subjektive Erleben haben.
- Die Schüler projizieren ihre Gefühle für das Fach, die Themen und den Lehrer (bei einem Fragebogen, welcher im Unterrichtsfach Informatik ausgeteilt wird) auf die Schule und das private Umfeld. Oder sie verstehen das Wort „privates Umfeld“ nicht. In beiden Fällen würden die Daten zum Fach, zum Lehrer und zu den Themen stimmig sein und könnten so interpretiert werden, wie in den letzten Kapiteln geschehen.



- Der statistische Zufall hat dafür gesorgt, dass in der OOP-First-Klasse deutlich mehr Schüler Probleme im privaten Umfeld haben als in der OOP-Later-Klasse. Jetzt muss spekuliert werden, wie stark die Gefühle im privaten Umfeld Einfluss nehmen auf die einzelnen Schul-Bereiche. Ist der Einfluss groß, dann könnte dies die Ursache dafür sein, dass alle Schul-Bereiche von den OOP-First-Schülern negativer gesehen werden als von den OOP-Later-Schülern. Ist der Einfluss klein, dann könnten die Daten so interpretiert werden, wie bisher geschehen. Allerdings mit der dann weiterhin offenen Frage, warum der Bereich „Schule“ von der OOP-First-Klasse negativer erlebt wird als von der OOP-Later-Klasse.

Zusammenfassend lässt sich sagen, dass manche Ergebnisse verschieden interpretiert werden können, sich also nicht eindeutig erklären lassen und somit sehr vorsichtig in die fachdidaktische Diskussion eingebracht werden sollten.

## 7.5 Ergebnisse im Kontext zu den Forschungsfragen und Forschungshypothesen

Im letzten Unterkapitel dieser Arbeit soll der Kreis geschlossen werden, also die bisher diskutierten Ergebnisse in den Kontext zu den ursprünglichen Ideen, Thesen, Forschungsfragen und Forschungshypothesen gestellt werden. Hierfür sollen die folgenden Fragen beantwortet werden:

- Können die ursprünglichen Thesen bestätigt werden (siehe Kapitel 7.5.1)?
- Können die Forschungsfragen beantwortet werden (siehe Kapitel 7.5.2)?
- Können die Forschungshypothesen bestätigt werden (siehe Kapitel 7.5.3)?
- Gibt es Schlussfolgerungen für den Informatik-Anfangsunterricht (siehe Kapitel 7.5.4)?
- Gibt es interessante Nebenaspekte (siehe Kapitel 7.5.5)?
- Gibt es Einschränkungen bei der Gültigkeit (siehe Kapitel 7.5.6)?
- Gibt es weiterführende Forschungsfragen (siehe Kapitel 7.5.7)?

### 7.5.1 Können die ursprünglichen Thesen bestätigt werden?

Im Kapitel 2.4.5 wurden die ursprünglichen Thesen aus dem Promotionsvorschlag aufgeführt.

Die erste These lautete:

„Das Thema OOP gleich zu Beginn der ersten Informatik-Stunden wird von den Schülerinnen und Schülern schwerer erlebt, als wenn es erst nach einer längeren „Einführung in die prozedurale Programmierung“ erfolgt.“

Das Thema 1 („Einführung in die OOP mit BlueJ“) wird von den beiden Klassen unterschiedlich schwer erlebt (siehe Abbildung 129 und Abbildung 167): Die OOP-First-Schüler liegen mit einem „Schwere-Wert“ von 2,5 deutlich über dem Wert von 1,9 der OOP-Later-Schüler. Diese große Mittelwerts-Differenz (0,6) zwischen den beiden Klassen ist aber knapp nicht signifikant ( $p = 0,054$ ).

Die zweite These lautete:

„Ein Unterrichts-Einstieg mit „Programmieren im Kleinen“ ohne OOP-Bezug steigert den „Wohlfühlfaktor“ im Unterricht und somit auch die Motivation bzw. die Erfolgserlebnisse des einzelnen (leistungsschwachen) Schülers und verbessert somit auch die Lernergebnisse. Parallel dazu wird der Schwierigkeitsgrad und die Komplexität einiger Themen verringert.“

In dieser These sind viele Teilaspekte enthalten, die jetzt einzeln analysiert werden sollen:

- Wohlfühlfaktor bezogen auf die Themen: Bei sechs von neun Themen (ein Thema ist ausgeglichen) fühlt sich die OOP-Later-Klasse weniger gestresst und wohler als die OOP-First-Klasse (siehe Kapitel 6.4.2b bzw. 6.4.2c und Abbildung 119 bzw. Abbildung 120). Die OOP-First-Schüler liegen mit dem Wohlfühlfaktor 2,66 über dem Wert 2,49 der OOP-Later-Schüler. Höhere Werte bedeuten schlechtere Ergebnisse. Diese Mittelwerts-Differenz (0,17) zwischen den beiden Klassen (siehe Abbildung 116) ist aber nicht signifikant ( $p = 0,08$ ).
- Wohlfühlfaktor bezogen auf das Fach Informatik: Die Werte der OOP-Later-Klasse sind bei sechs von neun zeitlichen Messpunkten für das Fach Informatik (bei drei ausgeglichenen Werten) besser als die der OOP-First-Klasse (siehe Kapitel 6.4.2a und Abbildung 117 bzw. Abbildung 118). Die OOP-First-Schüler liegen mit dem Wohlfühlfaktor 2,6 über dem Wert 2,3 der OOP-Later-Schüler. Diese Mittelwerts-Differenz (0,3) zwischen den beiden Klassen (siehe Abbildung 116) ist signifikant ( $p = 0,012$  mit  $d = 1,36$ ).
- Motivation bezogen auf das Fach Informatik: Die Motivationswerte der OOP-Later-Klasse sind bei allen zeitlichen Messpunkten für das Fach Informatik besser als die der OOP-First-Klasse (siehe Kapitel 6.4.4 und Abbildung 135 bzw. Abbildung 136). Die OOP-First-Schüler liegen mit dem „Motivationsfaktor“ 2,3 deutlich über dem Wert 1,9 der OOP-Later-Schüler. Höhere Werte bedeuten schlechtere Ergebnisse. Diese Mittelwerts-Differenz (0,4) zwischen den beiden Klassen (siehe Abbildung 134) ist höchst signifikant ( $p = 0,0002$  mit  $d = 2,34$ ).
- Lernergebnisse: Die Ergebnisse der OOP-First-Klasse und der OOP-Later-Klasse im Nachhaltigkeitstest sind fast gleich (siehe Abbildung 107). Es gibt keine (signifikanten) Unterschiede, weder im Gesamtergebnis (OOP-First: 54,4% / OOP-Later: 54,9%), noch in den Teilresultaten.
- Schwierigkeitsgrad der Themen: Die „empfundene Schwere“ der einzelnen Themen (siehe Kapitel 6.4.3d, Abbildung 129 und Abbildung 167) liegt in der OOP-First-Klasse durchschnittlich bei 2,74 und in der OOP-Later-Klasse durchschnittlich bei 2,54. Diese Mittelwerts-Differenz (0,2) ist nicht signifikant ( $p = 0,17$ ).

Die zweite These kann somit nur teilweise belegt werden. Ein Unterrichts-Einstieg mit „Programmieren im Kleinen“ ohne OOP-Bezug steigert zwar (signifikant) den Wohlfühlfaktor und die Motivation bezogen auf das Fach Informatik. Dies hat aber keine Auswirkung auf den objektiven Lernerfolg. Durchschnittlich werden die einzelnen Themen von den OOP-Later-Schülern etwas leichter erlebt, dieser Unterschied ist aber nicht signifikant.

Die dritte These lautete:

„Der OOP-First-Ansatz unterscheidet sich von dem OOP-Later-Ansatz überwiegend nur in der unterschiedlichen Reihenfolge der Themen. Daher können für beide Ansätze gleichartige Themen benannt werden.“

Die dritte These wurde durch die empirische Studie bestätigt. Es konnten neun Themengebiete generiert werden, die sowohl beim OOP-First-Ansatz als auch beim OOP-Later-Ansatz zur Geltung kamen. Nach einem Jahr sind alle neun Themengebiete unterrichtet worden, nur die zeitliche Reihenfolge der Themen wurde variiert.

## 7.5.2 Können die Forschungsfragen beantwortet werden?

Die ursprüngliche Forschungsfrage, die indirekt im Titel dieser Dissertationsschrift angeführt ist, wurde in Kapitel 2.4.1 in zwei Teile aufgeteilt.

Die Forschungsfrage zum objektiven Aspekt lautete:

„Gibt es Unterschiede im Lernerfolg von Schülerinnen und Schülern im Informatik-Anfangsunterricht in Abhängigkeit eines OOP-First- bzw. OOP-Later-Vorgehens? Wenn ja, welche?“

Die Forschungsfrage zum subjektiven Aspekt lautete:

„Gibt es Unterschiede im subjektiven Erleben des Unterrichts von Schülerinnen und Schülern im Informatik-Anfangsunterricht in Abhängigkeit eines OOP-First- bzw. OOP-Later-Vorgehens? Wenn ja, welche?“

Die Antworten zum objektiven Aspekt (siehe Kapitel 6.3, speziell auch Unterkapitel 6.3.3 und Unterkapitel 6.3.5) sind wie folgt:

- Es wurden keine signifikanten Unterschiede zwischen dem OOP-First-Vorgehen und dem OOP-Later-Vorgehen (bezogen auf alle Schülerinnen und Schüler) im Hinblick auf den Lernerfolg festgestellt bzw. gemessen. Sowohl was das Gesamtergebnis als auch was die einzelnen Themenresultate anbelangt, ergaben sich keine signifikanten Unterschiede [ES09c, S. 129]. Beide Vorgehensweisen produzierten vergleichbare Lernerfolge im Hinblick auf den verwendeten Nachhaltigkeitstest.
- Es gibt bei der Lerngruppe der leistungsstärksten Schüler keine signifikanten Unterschiede im Lernerfolg in Abhängigkeit von der fachdidaktischen Vorgehensweise.
- Es gibt bei der Lerngruppe der leistungsschwächsten Schüler offensichtliche Unterschiede im Lernerfolg in Abhängigkeit von der fachdidaktischen Vorgehensweise. Die leistungsschwächsten OOP-Later-Schüler erzielen mit 36% der Punkte im Nachhaltigkeitstest deutlich bessere Ergebnisse als die leistungsschwächsten OOP-First-Schüler mit 28,2% der maximal möglichen Punkten. Dieser Ergebnis-Unterschied (7,8%) ist aber bei dem vorgegebenen Signifikanzniveau nicht signifikant ( $p = 0,1542$ , dies entspricht einer Irrtumswahrscheinlichkeit von 15,42%).
- Bei der Lerngruppe der Schülerinnen wird aufgrund der kleinen Stichprobe (jeweils nur zwei Schülerinnen) auf eine konkrete Aussage verzichtet.

Die Ergebnisse befinden sich in Einklang mit den Untersuchungen von Vilner, Zur und Gal-Ezer [VZG07] bzw. Tew, McCracken und Guzdial [TMG05], bei denen auch keine Unterschiede gefunden wurden bzw. vorhandene Unterschiede nach einiger Zeit verschwanden [ES09a, S. 19].

Trotzdem sollte man diese Ergebnisse mit Vorsicht betrachten. Nur weil in dieser Studie keine Unterschiede gefunden wurden, bedeutet dies nicht unbedingt, dass es keine Unterschiede gibt. Nur bezogen auf das spezielle Studiendesign, die speziellen Schüler, die speziellen Themen, die speziellen Lehrer, die speziellen Unterrichtsmaterialien, die spezielle Programmiersprache, die speziellen Tests usw. wurden keine Unterschiede gefunden.

Die Antworten zum subjektiven Aspekt (siehe Kapitel 6.4, Abbildung 115 und [ES10a, S. 109]) sind wie folgt:

- 18 von 22 gemessene Teil-Werte zum subjektiven Erleben liegen unter dem Mittelwert von 3,0, spiegeln also positives subjektives Erleben wider, zwei Werte treffen genau den Durchschnitt und nur zwei Werte spiegeln geringfügig negatives subjektives Erleben wider.
- Der Durchschnitt der elf Teil-Werte der OOP-First-Klasse ( $\bar{x} = 2,59$ ) ist höher (also schlechter) als der Durchschnitt der elf Werte der OOP-Later-Klasse ( $\bar{x} = 2,33$ ).
- Alle elf Teil-Werte der OOP-First-Klasse sind höher (also schlechter) als die entsprechenden Werte der OOP-Later-Klasse. Neunmal sind die Werte sogar signifikant unterschiedlich.
- Das subjektive Erleben (Mittelwertbildung über alle drei Erlebnis-Dimensionen) ist bei den OOP-Later-Schülern in allen vier Frage-Bereichen signifikant besser als bei den OOP-First-Schülern (siehe Abbildung 174).
- Das subjektive Erleben (Mittelwertbildung über alle 4 Frage-Bereiche) ist bei den OOP-Later-Schülern in allen drei Erlebnis-Dimensionen besser als bei den OOP-First-Schülern (siehe Abbildung 174).
- Das subjektive Erleben der Lerngruppe der leistungsstärksten OOP-Later-Schüler bezogen auf die Themen ist bis auf eine Ausnahme (Thema 9: „Assoziation“) schlechter als das subjektive Erleben der leistungsstärksten OOP-First-Schüler (OOP-First: 1,96 / OOP-Later: 2,08). Obwohl der Unterschied nicht signifikant ist, sind die Werte m. E. aber trotzdem sehr interessant, weil in der Regel (s. o.) immer die OOP-Later-Schüler bessere Werte als die OOP-First-Schüler haben. Für leistungsstarke Schüler scheint somit im Hinblick auf das subjektive Erleben das OOP-First-Vorgehen von Vorteil zu sein.
- Das subjektive Erleben der Lerngruppe der leistungsschwächsten OOP-First-Schüler bezogen auf die Themen ist bis auf eine Ausnahme (Thema 2: „Datentypen und Steuerstruktur Sequenz“) schlechter als das subjektive Erleben der leistungsschwächsten OOP-Later-Schüler (OOP-First: 2,66 / OOP-Later: 2,54). Da der Unterschied (0,12) etwa in der Größenordnung liegt, welcher auch bei der gesamten Lerngruppe (2,40 – 2,22 = 0,18) vorhanden ist, kann man m. E. daraus nicht unbedingt folgern, dass für leistungsschwache Schüler im Hinblick auf das subjektive Erleben das OOP-Later-Vorgehen einen kleinen Vorteil hat.
- Bei der Lerngruppe der Schülerinnen wird aufgrund der kleinen Stichprobe (jeweils nur zwei Schülerinnen) auf eine konkrete Aussage verzichtet.

### 7.5.3 Können die Forschungshypothesen bestätigt werden?

Im Kapitel 2.4.6 waren ausgehend von den beiden Forschungsfragen zwei ungerichtete Alternativhypothesen bzw. zwei Nullhypothesen formuliert worden. In der Studie wurde dann für die Signifikanzberechnungen der zweiseitige t-Test (Bestandteil des Open-Source-Statistik-Tools *R* (Version 2.7.1)) angewendet (vgl. Kapitel 6.1).

Die erste ungerichtete Alternativhypothese lautete:

„Der Lernerfolg von Schülerinnen und Schülern bei einem OOP-Later-Vorgehen unterscheidet sich von dem Lernerfolg bei einem OOP-First-Vorgehen.“

Diese Hypothese wird abgelehnt: Für alle neun Themen und die beiden Zusatzthemen konnten (mit Hilfe des Nachhaltigkeitstests) keine signifikanten Unterschiede festgestellt werden.

Die zweite ungerichtete Alternativhypothese lautete:

„Das subjektive Erleben von Schülerinnen und Schülern bei einem OOP-Later-Vorgehen unterscheidet sich von dem subjektiven Erleben bei einem OOP-First-Vorgehen.“

Diese Hypothese wird unter Vorbehalt angenommen: In sehr vielen Teilbereichen des subjektiven Erlebens unterscheiden sich die Werte der OOP-First-Schüler signifikant von denen der OOP-Later-Schüler. In der Regel mit der Tendenz, dass OOP-Later-Schüler das positivere subjektive Erleben haben. Der Vorbehalt ergibt sich daraus, dass die OOP-First-Schüler auch bei den Werten zum privaten Umfeld besser liegen und nicht ausgeschlossen werden kann, dass evtl. doch die unterschiedlichen Lehrer die Ergebnisse beeinflusst haben.

Dementsprechend lautete die erste Nullhypothese:

„Der Lernerfolg von Schülerinnen und Schülern bei einem OOP-Later-Vorgehen ist gleich dem Lernerfolg bei einem OOP-First-Vorgehen.“

Diese Nullhypothese wird angenommen, da die entsprechende ungerichtete Alternativhypothese abgelehnt worden ist (Begründung s. o.). Es wird aber noch einmal darauf hingewiesen, dass diese Hypothese Gültigkeit hat bezogen auf das spezielle Studiendesign, die speziellen Unterrichts-Variablen und die speziellen Tests. Es wurden in dieser Studie keine Unterschiede im Lernerfolg gefunden, es könnten trotzdem aber welche existieren.

Die zweite Nullhypothese lautete:

„Das subjektive Erleben von Schülerinnen und Schülern bei einem OOP-Later-Vorgehen ist gleich dem subjektiven Erleben bei einem OOP-First-Vorgehen.“

Diese Nullhypothese wird abgelehnt, da die entsprechende ungerichtete Alternativhypothese angenommen worden ist (Begründung s. o.).

## 7.5.4 Gibt es Schlussfolgerungen für den Informatik-Anfangsunterricht?

Eine im Laufe dieser Studie oft gestellte Frage lautete:

„Welche der beiden fachdidaktischen Vorgehensweisen sollte denn nun gewählt werden: OOP-First oder OOP-Later?“

Da m. E. in der fachdidaktischen Diskussion die Lehrer bzw. Dozenten sich oft eine eigene Meinung gebildet haben und sich in der Regel einem der vielen „Lager“ zuordnen lassen (OOP-First, OOP-Later, objects-first, OOM-First usw.) ist eine mögliche (und evtl. die einfachste) Antwort:

„Es kann das Vorgehen gewählt werden, welches der Lehrer bzw. Dozent für das richtige hält. Aber er sollte sich mit den Nachteilen der Vorgehensweise auseinandersetzen und diese gezielt reduzieren, denn beide fachdidaktische Vorgehensweisen haben spezielle Vor- und Nachteile. Und er sollte die Lernerfolge der Schülerinnen und Schüler evaluieren, damit er seinen Unterricht immer mehr verbessern kann.“

Die Begründung für diese Antwort lautet, dass ein leidenschaftlicher OOP-First-Befürworter in der Regel besseren OOP-First-Unterricht machen wird als ein leidenschaftlicher OOP-Later-Befürworter, der gezwungenermaßen OOP-First machen muss. Und natürlich umgekehrt: Ein leidenschaftlicher OOP-Later-Befürworter wird in der Regel besseren OOP-Later-Unterricht machen als ein leidenschaftlicher OOP-First-Befürworter, der gezwungenermaßen OOP-Later machen muss, gleich gute Lehrer vorausgesetzt.

Dieser doch sehr einfache Sachverhalt sollte aber in den Kontext der Studienergebnisse gestellt werden. Die Ergebnisse lassen keine der beiden Vorgehensweisen als „strahlende Siegerin“ dastehen, sondern sie liefern Argumente für beide fachdidaktische Vorgehensweisen.

Die Argumente sowohl für eine OOP-First-Vorgehensweise als auch für OOP-Later-Vorgehensweise könnten wie folgt lauten:

- Die Lernergebnisse beider Vorgehensweisen sind nach einem Jahr gleich (siehe Kapitel 6.3.3).
- Die objektorientierte Modellierung (OOM) kann auch in der objektorientierten Programmierung (OOP) vermittelt werden (siehe Kapitel 7.3.4).

Die Argumente für ein OOP-Later-Vorgehen könnten wie folgt sein:

- Das subjektive Erleben der OOP-Later-Schüler ist (oft signifikant) besser als das der OOP-First-Schüler (siehe Kapitel 6.4.5).
- Für Lerngruppen, die überwiegend aus Schülerinnen oder aus sehr leistungsschwachen Schülern bestehen, scheint das OOP-Later-Vorgehen bezogen auf das subjektive Erleben einen kleinen Vorteil zu besitzen (siehe Kapitel 6.4.6).
- Die einzelnen Teilthemen bei einer OOP-Later-Vorgehensweise bauen mehr auf den vorher unterrichteten Themen auf als dies bei einem OOP-First-Vorgehen der Fall ist (siehe Kapitel 6.4.3 und Kapitel 7.4.5), die Themen scheinen besser vernetzt.

Die Argumente für ein OOP-First-Vorgehen könnten wie folgt sein:

- Für Lerngruppen, die überwiegend aus sehr leistungsstarken Schülern bestehen, scheint das OOP-First-Vorgehen bezogen auf das subjektive Erleben einen Vorteil zu besitzen (siehe Kapitel 6.4.6).
- Da das (oft signifikant) schlechtere subjektive Erleben der Schülerinnen und Schüler bei einem OOP-First-Vorgehen erstens keinen Einfluss auf den Lernerfolg hat (siehe Kapitel 7.4.1) und zweitens sich die Ergebnis-Unterschiede evtl. nur durch eine „Schieflage“ der Ergebnisse der OOP-First-Klasse im Verhältnis zu den Ergebnissen der OOP-Later-Klasse ergeben (siehe Kapitel 7.4.6), müssen diese Ergebnisse nicht berücksichtigt werden.

Es könnten einem OOP-First-Lehrer folgende Hinweise gegeben werden:

- Die OOP-First-Klasse erlebt den Texteditor (mit dem Thema 2: „Datentypen und Steuerstruktur Sequenz“) wesentlich schwerer als die OOP-Later-Klasse (siehe Kapitel 6.4.3). Vielleicht kann der Lehrer die Schüler im BlueJ-Tool schon verstärkt mit dem dort vorhandenen Texteditor arbeiten lassen (also nicht nur Objekte erzeugen und Methoden aufrufen, sondern auch Quelltext ändern und erweitern lassen) um damit den späteren Umstieg auf den Texteditor zu erleichtern.
- Der größte Sprung (siehe Kapitel 6.4.3) von „relativ schwer“ (3,1) auf „relativ leicht“ (2,4) mit dem Wert 0,7 findet vom Thema 7 („Arrays und Strings“) zum Thema 8 („Vererbung“) statt. Das Thema 7 könnte evtl. durch längeres bzw. besseres Unterrichten erleichtert werden.
- Evtl. „verführt“ das OOP-First-Vorgehen zu einem zu schnellen Unterrichten von einigen Themen (siehe Kapitel 6.2.1). Das OOP-First-Vorgehen hatte in der Vorstudie signifikant schlechtere Lernerfolge. Ein Begründungsversuch hierfür war z. B. das schnellere Vorgehen des OOP-First-Lehrers. Der OOP-First-Lehrer sollte sich also für die einzelnen Themen genügend Zeit lassen.

Hinweise für einen OOP-Later-Lehrer könnten wie folgt sein:

- Das Thema 4 („Methoden“) sollte nach den Ergebnissen dieser Studie erst nach dem Einstieg in die OOP unterrichtet werden (siehe Kapitel 3.3.14 und Kapitel 6.4.3), weil es dann als leichter erlebt wird. Der Grund hierfür könnte daran liegen, dass mit dem BlueJ-Tool erst die Methoden angewendet werden, bevor sie selber deklariert bzw. programmiert werden müssen. Dies scheint leichter zu sein als innerhalb der prozeduralen Vorgehensweise gleich mit der Methodendeklaration (inkl. formalen Parametern und Rückgabewert) beginnen zu müssen, da nur über die eigenen deklarierten (statischen) Methoden Methodenaufrufe erfolgen können.
- Der größte Sprung im „Schwere-Empfinden“ (siehe Kapitel 6.4.3) von „leicht“ (1,9) auf „durchschnittlich schwer“ (2,7) mit dem Wert 0,8 findet vom Thema 1 („Einführung in die OOP mit BlueJ“) zum Thema 3 („Klasse und Objekt“) statt. Vielleicht könnte der Lehrer innerhalb des BlueJ-Tools schon eigene Klassen erstellen lassen, so dass das Thema 1 evtl. ein bisschen schwerer und das Thema 3 ein bisschen leichter wird. Alles unter der fachdidaktischen Prämisse, dass große Sprünge vermieden werden sollten.

Unabhängig vom fachdidaktischen Vorgehen lassen sich die folgenden Vorschläge machen:

- Es spricht vieles für den Einsatz des BlueJ-Tools im Informatik-Anfangsunterricht (siehe Umfrage-Ergebnisse bezogen auf das Thema 1 („Einführung in die OOP mit BlueJ“) in Kapitel 6.4.2). Beim Wohl-  
gefühl- und Stressfaktor haben beide Klassen gute Werte: Die OOP-First-Klasse liegt mit dem Wert 2,2 nahe

an dem Wert 2,0 der OOP-Later-Klasse. Für die OOP-Later-Klasse ist dies mit Abstand der beste Wohlfühlwert aller Themen. Die Begründung für den augenscheinlich großen Unterschied (0,6) bei dem „Schwere-Empfinden“ zwischen den beiden Klassen (OOP-First: 2,5 / OOP-Later: 1,9) bei der „Einführung in die OOP“ liegt auf der Hand: Die OOP-Later-Klasse hat relativ großes Vorwissen bei dem erstmaligen Aufruf des BlueJ-Tools.

- Es spricht vieles dafür, dass Thema 9 („Assoziation“) aus dem Informatik-Anfangsunterricht (mit einem Stundenumfang von ca. 100 Schulstunden à 45 Minuten) herauszunehmen: Es schneidet durchgehend in den drei Erlebnisdimensionen (siehe Kapitel 6.4) und auch im Nachhaltigkeitstest (siehe Kapitel 6.3.3) schlecht ab. Für beide Klassen zusammen ist es mit Abstand das schwerste Thema (siehe Kapitel 7.3.1). Ohne das letzte Thema „Assoziation“ wären die Werte für das „subjektive Erleben“ bezogen auf die Themen am Anfang und am Ende des Schuljahres für beide fachdidaktischen Vorgehensweisen (fast) gleich (siehe Abbildung 139).
- Das Thema 7 („Arrays und Strings“) schneidet beim Nachhaltigkeitstest (siehe Abbildung 107) fast so schlecht ab wie das Thema 9 („Assoziation“). Und innerhalb des Themas 7 wird das Einzel-Thema „Arrays“ wesentlich schlechter von den Schülerinnen und Schülern bearbeitet als das Thema „Strings“ (siehe Abbildung 158). Daher könnte man Teile der eingesparten Zeit (durch den Wegfall des Themas „Assoziation“) dafür nehmen, das Thema „Arrays“ länger bzw. besser zu unterrichten. Dann würden m. E. auch die Lernerfolge bei diesem Thema größer werden.

### 7.5.5 Gibt es interessante Nebenaspekte?

Im Verlauf der Studie gab es einige Ergebnisse, die eigentlich nicht Ziel der Forschung gewesen sind, die aber trotzdem interessant genug sind, um erwähnt zu werden:

- Die schwache Synchronisation des OOP-First-Vorgehens mit dem OOP-Later-Vorgehen (so geschehen in der Vorstudie) führte dazu, dass die OOP-First-Klasse wesentlich schneller unterrichtet wurde, also mehr Themen angeboten bekam. Dies könnte man ja als Vorteil sehen, wenn nicht die OOP-First-Klasse in Hinblick auf den Lernerfolg (teilweise signifikant) schlechter abgeschnitten hätte. Vielleicht lag das schnelle Vorgehen am Lehrer, vielleicht „verführt“ aber auch ein OOP-First-Vorgehen zu einem relativ schnellen Unterrichten. Ein (relativ trivialer) didaktischer Grundsatz wird durch die Vorstudie aber auf jeden Fall bestätigt: Verschiedene Vorgehensweisen im (Informatik-) Unterricht können zu verschiedenen Lern-Ergebnissen führen (siehe Kapitel 3.3.14 und Kapitel 6.2.1).
- Ein weiterer (relativ trivialer) didaktischer Grundsatz konnte bestätigt werden: Durch fachdidaktische Entscheidungen kann Unterricht verbessert werden. Z. B. wurde in der Vorstudie durch die Datenerhebung ein schlechtes Abschneiden des Themas „Methoden“ im Hinblick auf das „Schwere-Empfinden“ bekannt (siehe Kapitel 3.3.14 und Kapitel 6.2.2). Die fachdidaktische Maßnahme, dieses Thema in der Hauptstudie zu einem späteren Zeitpunkt zu unterrichten, führte zu wesentlich besseren Werten im Hinblick auf das „Schwere-Empfinden“ (siehe Kapitel 6.4.3).
- Für jeden Unterricht ist es wichtig, dass ein „guter Lehrer“ unterrichtet. Viele Didaktik-Bücher und viele didaktische Seminare beschäftigen sich mit dem Thema, was einen „guten Lehrer“ ausmacht und ob dieses



Merkmal durch die Lehrerausbildung verbessert werden kann und wie stark die „angeborene Komponente“ ist. Die Studie zeigte, dass auch ein schon lange im Schuldienst tätiger Lehrer sich durch bewusste Reflexion seiner Person bzw. seines Verhaltens sich (aus Schülersicht) verbessern kann. Z. B. wurde der OOP-Later-Lehrer in der Vorstudie von den Schülerinnen und Schülern schlechter gewertet als der OOP-First-Lehrer. Der OOP-Later-Lehrer hat dann erfolgreich die Lehren aus der Vorstudie gezogen und bewusst sein Verhalten geändert. Dies hatte zur Folge, dass sich sein Wert 2,2 (in der Vorstudie) auf den Wert 1,8 (in der Hauptstudie) verbesserte. Nun wurden (in der Hauptstudie) die beiden Lehrer (im Hinblick auf die Frage 7) gleichartig (mit einem Wert von 1,8) von den Schülern erlebt (siehe Kapitel 3.3.14 und Kapitel 7.2.1).

- Die signifikanten Ergebnisunterschiede im Lernerfolg im Vergleichstest verschwanden im Nachhaltigkeits-test. Der favorisierte Erklärungsansatz hierfür war eine unterschiedlich Präparierung bzw. Vorbereitung der beiden Klassen auf den Vergleichstest. Durch die Ergebnisunterschiede konnte nun gezeigt werden, dass zum einen eine Vorbereitung auf einen Test mit (bzw. nahe an) den tatsächlichen Aufgaben nur einen kurzzeitigen Erfolg hat und keine Nachhaltigkeit besitzt und dass zum anderen von dieser speziellen Vorbereitung nur die leistungsschwachen Schüler profitieren. Bei den leistungsstarken Schülern hat dies keine offensichtlichen Auswirkungen (siehe Kapitel 6.3.3 und Kapitel 6.3.5).
- Nur eine von vier Schülerinnen gehört der leistungsschwächsten Schülergruppe an. Die Schülerinnen erzielen bessere Ergebnisse in der Informatik als in der Mathematik (siehe Kapitel 6.3.5).

Die mangelnde Synchronisation, die Unterschiede im Lernerfolg, das schnellere Vorgehen, das unterschiedliche Lehrerbild und die Ursachen für die Unterschiede in den Testergebnissen wurden durch Beobachtungen, Abfragen und Tests ermittelt. Somit wird ein weiterer (relativ trivialer) didaktischer Grundsatz bestätigt: Durch Evaluation lässt sich Unterricht verbessern.

### 7.5.6 Gibt es Einschränkungen bei der Gültigkeit?

Es sind die folgenden Fehl- oder Überinterpretationen der Studien-Ergebnisse möglich [ES09c, S. 131]:

- Es kann nicht entschieden werden über den Einfluss der gesetzten Randbedingungen. Was geschieht z. B. bei der Wahl anderer Werkzeuge, Programmiersprachen etc.? Auch kann nicht exakt der Einfluss der unterschiedlichen Lehrer bestimmt werden. Trotzdem ist es schon ein Erfolg, einen systematischen Vergleich geschafft zu haben. Darauf könnten Folgeuntersuchungen aufbauen (siehe auch nächstes Kapitel).
- Die Studie bezieht sich auf OOP-First und nicht auf objects-first, wobei der Unterschied marginal sein mag (Hinweis: Auch bei der zusätzlichen OO-Modellierungs-Aufgabe gab es keinen Unterschied im Lernerfolg). Größere Unterschiede könnten sich ergeben, wenn OOM-First mit OOP-Later verglichen würde. Das ist dann aber eher keine didaktisch-methodische Frage, sondern eher eine Frage der Gewichtung von Lernzielen.
- Die Testfragen waren überwiegend aus dem Anforderungsbereich II (Vergleichstest: 79% / Nachhaltigkeits-test: 80%), der Anforderungsbereich III fehlte gänzlich. Die Quelltext-Entwicklung bei den Tests erfolgte auf dem Papier, nicht am Rechner.

## 7.5.7 Gibt es weiterführende Forschungsfragen?

Das folgende Kapitel ist teilweise (auch wörtlich) übernommen aus den Papieren zur INFOS'09 [ES09c, S. 132] und zur ICER'09 [ES09a, S. 24].

Das Hauptresultat dieser Studie ist, dass keine (signifikanten) Unterschiede im Lernerfolg zwischen einem OOP-First- und einem OOP-Later-Vorgehen festgestellt bzw. gemessen wurden (die p-Werte für die elf abgefragten Themen liegen zwischen 0,18 und 0,83).

Unterschiede gab es hingegen beim subjektiven Erleben des Unterrichts.

Ausgehend von der Modellbildung, dass der Hauptunterschied zwischen den beiden fachdidaktischen Vorgehensweisen nur in der unterschiedlichen Sequenz gleichartiger Lerninhalte besteht, könnte nun daraus der Schluss gezogen werden, dass die Diskussion um OOP-First contra OOP-Later weitaus weniger wichtig ist als angenommen. Statt dessen könnte sich die Diskussion mehr auf die Detail- bzw. Prozessunterschiede fokussieren, die ausführlich in dieser Arbeit herausgearbeitet und (überwiegend) auch begründet werden konnten.

Ein paar Details und Fragen konnten aber nicht wirklich erklärt bzw. beantwortet werden. Diese könnten nun die Fragestellungen für weiterführende Studien sein:

- Gibt es wirklich keine (signifikanten) Unterschiede im Lernerfolg oder hat diese Studie nur die Unterschiede nicht gefunden? Ist das Ergebnis evtl. bedingt durch die kleine Stichprobengröße oder gibt es wirklich nur geringe Unterschiede?
- Was passiert, wenn die Randbedingungen geändert werden, z. B. das eingesetzte OOP-Einstiegs-Tool BlueJ oder die eingesetzte Programmiersprache Java? Wie stark war genau der Einfluss der beiden unterschiedlichen Lehrer?
- Wie stark hat sich der spezielle Schultyp des Beruflichen Gymnasiums bemerkbar gemacht? Das Berufliche Gymnasium ist kein „normales“ Gymnasium und fängt erst ab der Klassenstufe 11 an. Auf dieses kommen eher Real- und Gesamtschüler als ehemalige Gymnasiasten (siehe Kapitel 3.3.6). Sind diese Schüler nicht eher leistungsschwach? Gibt es an Beruflichen Gymnasien in Oberstufenzentren nicht überproportional viele verhaltensauffällige Schüler, die eine spezielle Betreuung benötigen?
- Was ist, wenn man von der sequenziellen Modellbildung abgeht, wenn z. B. das OOP-First-Vorgehen zu einem Spiralmodell verändert wird?
- Wie kann die innere Vernetzung der Themen bei OOP-First verbessert werden?
- Warum gibt es in der „empfundenen Schwere“ der Themen zwischen den beiden Klassen teilweise große Unterschiede, z. B. beim Thema „Arrays und Strings“, beim Thema „Vererbung“ und beim Thema „Assoziation“? Liegt es daran, dass das OOP-Later-Vorgehen die Themen schrittweise behandelt, wobei die Schritte klar voneinander abgegrenzt sind? Liegt es daran, dass das OOP-First-Vorgehen evtl. doch nicht die Themen ganz so stark sequenziell anbietet, sondern mehr integrativ? Erleben deshalb die OOP-First-Schüler die Themen durchschnittlich schwerer?
- Stimmt es, dass ein OOP-First-Vorgehen dazu „verführt“, die Themen schneller zu unterrichten?
- Warum gibt es zwischen der „empfundenen Schwere“ und der „tatsächlichen Schwere“ der Themen nur bei dem OOP-Later-Vorgehen eine nennenswerte Korrelation. Misst vielleicht der Faktor „empfundene

Schwere“ mehr den Lernprozess selber, während die Tests, die die „tatsächliche Schwere“ bestimmen, sich auf den Lern-Output fokussieren? War vielleicht der Nachhaltigkeitstest mehr im Sinne des OOP-Later-Lernprozesses?

- Warum genau ist das subjektive Erleben der OOP-Later-Klasse so viel besser als das der OOP-First-Klasse?
- Wenn die OOP-Later-Klasse sich so viel besser fühlt, warum hat dies keine Auswirkungen auf den Lernerfolg? Oder wird in Wahrheit beim OOP-Later-Vorgehen der Stoff „schlechter vermittelt“, aber das bessere Erleben gleicht diesen Nachteil wieder aus (indem es den Lernerfolg steigert)?
- Warum erlebt die OOP-First-Klasse die Schule und ihr privates Umfeld so viel negativer? Liegt es nur am unterschiedlichen „Ankreuzverhalten“ oder gab es andere Gründe?

Wenn man einige aktuelle Veröffentlichungen zu fachdidaktischen Fragestellungen betrachtet, so werden dort weiterhin Vorschläge zum konkreten Informatikunterricht gemacht. Sei es von Schubert und Schwill in Bezug auf „Unterrichtshilfen für den Informatikunterricht“ und auf „Anfangsunterricht“ (siehe [SS11, Kapitel 8 und 13]) oder von Oldenburg, der einen genetischen Weg zu OOP und OOM anregt (siehe [Old10]).

Die Unterrichtspraxis benötigt aber auch einen theoretischen Überbau. So ist ein letztes Problem im Hinblick auf die Interpretation der Ergebnisse dieser empirischen Studie und auf weiterführende Forschungsfragen das fehlende Erklärungsmodell. Fachdidaktische Fragen lassen sich nicht allein durch empirische Studien lösen, sondern brauchen auch einen theoretischen Rahmen. Empirische Ergebnisse müssen erklärbar gemacht, also in einen Zusammenhang von Ursache und Wirkung gebracht werden. Wenn der Unterschied zwischen OOP-First und OOP-Later nicht so groß ist wie angenommen, dann könnte man eine Theorie des Einstiegs in die Programmierung ohne „Paradigmen-Dogma“ entwickeln, die sich mehr auf die einzelnen Details des Lernprozesses konzentriert. Das würde der Diskussion einen Rahmen geben und die fachdidaktische Diskussion vertiefen und mit dem vorhandenen Wissen (auch aus anderen Studien) vernetzen. Einige fachdidaktische Arbeiten könnten bei diesem Erklärungsmodell mit einfließen: Das „Block Model“ als Hilfsmittel zur fachdidaktischen Analyse von Quelltexten in Bezug auf Dimensionen und Stufen des „Programmier-Lernens“ [Sch08], die „Object Interaction Hierarchy“, welche Kompetenzstufen für den Austausch von Botschaften zwischen Objekt beschreibt [BS06] und die „SOLO-Taxonomy“, welche eigentlich eine allgemeine Lern-Systematik ist, aber auch auf den Informatik-Anfangsunterricht angewendet werden kann [LSua06] [LAua04].

Will man einige Ergebnisse dieser Studie kreativ interpretieren, dann könnte man evtl. vermuten, dass das OOP-Later-Vorgehen durch seine vernetzte Themenstruktur den leistungsschwachen Schülern und den Schülerinnen entgegenkommt, während das OOP-First-Vorgehen mit seiner (gerade am Anfang) unklaren Themenabgrenzung die leistungsstarken Schüler anspricht. Wenn an das Berufliche Gymnasium des OSZ IMT eher leistungsschwache Schüler wechseln, wäre dies dann eine mögliche Erklärung, warum das OOP-Later-Vorgehen in dieser Studie die besseren subjektiven Erlebniswerte erzielt.

Die Entwicklung eines Erklärungsmodells war allerdings nicht die Aufgabe dieser Dissertationsschrift. Um mehr Daten für einen „Unterbau“ einer neuen fachdidaktischen Theorie zu gewinnen, könnte man sich der Meinung eines anonymen Gutachters (für das Papier zur ITiCSE 2010 (siehe [ES10a])) anschließen, der folgende Anregung in seiner Rückmeldung gab:

„Try to recruit people to redo the study in various forms.“

## 7.6 Zusammenfassung

Das Kapitel 7 gliederte sich in fünf Bereiche und sollte spezielle Fragestellungen beantworten bzw. spezielle Auswertungen und Interpretationen vornehmen.

Im Kapitel 7.1 wurde festgestellt, dass die neun Themen wie geplant unterrichtet werden konnten (siehe Kapitel 7.1.1) und dass die Abfragen zum subjektiven Erleben zeitlich wie geplant erfolgten (siehe Kapitel 7.1.2).

Danach wurden drei Problemfelder der empirischen Studie (siehe Kapitel 7.2) noch einmal bewertet. Im Kapitel 7.2.1 wurde nachgewiesen, dass die unterschiedlichen Lehrer relativ gleichartig waren. Dann wurde dargestellt (siehe Kapitel 7.2.2), dass die Ergebnisse des Vortests die Vermutung belegen, dass die Schüler (fast) ohne Programmierkenntnisse in den Informatik-Anfangsunterricht kommen. Im Kapitel 7.2.3 wurden die Regeln für den Umgang mit fehlenden Schülern erläutert. Diese Regeln sind der Grund, warum bei den Fragebögen zum subjektiven Erleben jeweils 22 Schüler berücksichtigt wurden, aber bei den Tests zum Lernerfolg nur 19 Schüler.

Im Hinblick auf den Lernerfolg in beiden Klassen (siehe Kapitel 7.3) wurde herausgefunden, dass die Themen „Assoziation“, „Arrays“ und „Methoden“ schwer und die Themen „Selektion“ und „OOM“ leicht sind (siehe Kapitel 7.3.1). Objektorientierte Themen sind durchschnittlich nicht (signifikant) schwerer als prozedurale Themen (siehe Kapitel 7.3.2). Es wurde ein Zusammenhang zwischen den Themen „Assoziation“ und „Arrays“ hergestellt und die Begründung dafür geliefert, warum die beiden Themen schwierige Themen sind (siehe Kapitel 7.3.3). In Kapitel 7.3.4 wurde herausgearbeitet, dass OOM in OOP-Kursen gut vermittelt werden kann.

In Bezug auf das subjektive Erleben des Informatikunterrichts (siehe Kapitel 7.4) wurde festgestellt, dass das subjektive Erleben (fast) keinen Einfluss auf den Lernerfolg hat (siehe Kapitel 7.4.1) und dass die beiden Klassen völlig unterschiedliche Themen als leicht bzw. schwer empfinden (siehe Kapitel 7.4.2). Die Korrelation zwischen der „empfundenen Schwere“ der Themen und den Ergebnissen des Nachhaltigkeitstests ist (nur) in der OOP-Later-Klasse relativ gut (siehe Kapitel 7.4.3). Im Kapitel 7.4.4 wurden dann Berührungspunkte zu anderen empirischen Untersuchungen angegeben. Im Kapitel 7.4.5 wurden zwei Begründungen dafür gegeben, warum die Themen bei einem OOP-Later-Vorgehen besser vernetzt sind. Danach wurde die Frage gestellt, ob die OOP-Later-Schüler den Unterricht wirklich subjektiv angenehmer erleben (siehe Kapitel 7.4.6).

Das letzte Kapitel (siehe Kapitel 7.5) schloss den Kreis von den Ergebnissen der Studie zu den ursprünglichen Thesen und Forschungsfragen. Die ursprünglichen Thesen konnten zum Teil belegt, zum Teil widerlegt werden (siehe Kapitel 7.5.1). Die beiden Forschungsfragen (siehe Kapitel 7.5.2) und die Forschungshypothesen (siehe Kapitel 7.5.3) wurden beantwortet bzw. überprüft: Die beiden Vorgehensweisen unterscheiden sich nicht im Lernerfolg, aber im subjektiven Erleben. Danach wurden Verbesserungsvorschläge für den Informatikunterricht gegeben (siehe Kapitel 7.5.4), z. B. sollte das Thema „Assoziation“ aus dem Unterricht herausgenommen werden. Einer von vielen interessanten Nebenaspekten dieser Studie (siehe Kapitel 7.5.5) war, dass unterschiedliche Vorbereitung auf einen Test zu unterschiedlichen Test-Ergebnissen führt, dieser Effekt aber nicht nachhaltig ist. Das vorletzte Kapitel gab Einschränkungen bei der Gültigkeit an (siehe Kapitel 7.5.6), z. B. im Hinblick auf den Einfluss der gesetzten Randbedingungen. Das letzte Kapitel stellte weiterführende Forschungsfragen (siehe Kapitel 7.5.7) und wies auf ein fehlendes fachdidaktisches Erklärungsmodell hin.

Auf Seite 385 werden die Ergebnisse dieser Studie kurz zusammengefasst.

## Literaturverzeichnis (Quellenangabe)

- [ABua05] Astrachan, O. (Moderator), Bruce, K., Koffman, E., Kölling, M. und Reges, S.: *Resolved: Objects Early has Failed*.  
In Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, St. Louis, Missouri, USA, 2005, SIGCSE '05, ACM, New York, S. 451-452,  
DOI=<http://doi.acm.org/1047344.1047359>, geprüft: 01.02.2010
- [Bal06] Balzert, Helmut: *Java5: Objektorientiert programmieren*.  
W3L GmbH, 2006, Herdecke, Bochum, ISBN 978-3-937137-08-7
- [Bal08] Balzert, Helmut: *Java, der Einstieg in die Programmierung*.  
W3L GmbH, 2008, 2. Auflage, Herdecke, Bochum, ISBN 978-3-868-34000-6
- [Bau01] Baumann, Rüdiger: *Assoziieren und Spezialisieren – Beispiele zum objektorientierten Entwurf in JAVA*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft 2, S. 10-17, LOG IN Verlag, 2001
- [Bau07] Baumann, Rüdiger: *Modellierung von Zuständen, Objekten und Suchbäumen*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 148/149, S. 73-80, LOG IN Verlag, 2007
- [BB04] Balzert, H. und Balzert, H.: *Modellieren oder Programmieren oder beides? – Plädoyer für einen schrittweisen Aufbau mentaler Modelle im Unterricht*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 128/129, S. 20-25, LOG IN Verlag, 2004
- [BCua07] Bergin, J., Clancy, M., Slater, D., Goldweber, M. und Levine, D. B.: *Day One of the Objects-First First Course: What To Do*.  
In Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education, Covington, Kentucky, USA, 2007, SIGCSE '07, ACM, New York, S. 264-265,  
DOI=<http://doi.acm.org/10.1145/1227310.1227404>, geprüft: 08.12.2009
- [BD02] Bortz, J. und Döring, N.: *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler*.  
Springer Medizin Verlag Heidelberg, 2002, 3. Auflage, Nachdruck 2005, ISBN 3-540-41940-3
- [BDua09] Bieniusa, A., Degen, M., Heidegger, P., Thiemann, P., Wehr, S., Gasbichler, M., Crestani, M., Klaeren, H., Knauel, E. und Sperber, M.: *Auf dem Weg zu einer robusten Programmierausbildung*.  
In Andreas Schwill (Hrsg.): Hochschuldidaktik der Informatik. HDI2008 – 3. Workshop des GI-Fachbereichs Ausbildung und Beruf/Didaktik der Informatik, Universitätsverlag Potsdam 2009, S. 67-79, ISBN 97-3-940793-75-1
- [BDSG09] Bundesbeauftragter für den Datenschutz und die Informationsfreiheit: *Bundesdatenschutzgesetz in der Fassung vom 01.09.2009*.  
[http://www.bfdi.bund.de/cln\\_136/SharedDocs/Publikationen/GesetzeVerordnungen/BDSG.html](http://www.bfdi.bund.de/cln_136/SharedDocs/Publikationen/GesetzeVerordnungen/BDSG.html)  
bzw. <http://www.bfdi.bund.de/cae/servlet/contentblob/409518/publicationFile/41798/BDSG.pdf>,  
geprüft: 06.02.2010
- [Ber00] Bergin, Joseph: *Why Procedural is the Wrong First Paradigm if OOP is the Goal*.  
Pace University, New York City, USA, 2000,  
<http://csis.pace.edu/~bergin/papers/Whynotproceduralfirst.html>, geprüft: 03.11.2009
- [BK06] Barnes, D. J. und Kölling, M.: *Java lernen mit BlueJ*.  
Pearson Education Deutschland, München, 2006, ISBN 978-3-8273-7152-2
- [BL03] Bortz, J. und Lienert, G. A.: *Kurzgefasste Statistik für die klinische Forschung. Leitfaden für die verteilungsfreie Analyse kleiner Stichproben*.  
Springer Medizin Verlag Heidelberg, 2003, ISBN 978-3-540-00069-3
- [BR05] Bergin, S. und Reilly, R.: *Programming: factors that influence success*.  
In ACM SIGCSE Bulletin, Volume 37, Issue 1, 2005, ACM, New York, S. 411-415,  
DOI=<http://doi.acm.org/10.1145/1047124.1047480>, geprüft: 02.04.2010
- [Bri00] Brinda, Thorsten: *Objektorientiertes Modellieren – Sammlung und Strukturierung von Übungsaufgaben im Informatikunterricht*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft 5, S. 39-49, LOG IN Verlag, 2000

- [Bri04] Brinda, Thorsten: *Didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sekundarstufe II*.  
Dissertation an der Universität Siegen, 2004, siehe auch: [http://deposit.ddb.de/cgi-bin/dokserv?idn=971190747&dok\\_var=d1&dok\\_ext=pdf&filename=971190747.pdf](http://deposit.ddb.de/cgi-bin/dokserv?idn=971190747&dok_var=d1&dok_ext=pdf&filename=971190747.pdf),  
geprüft: 01.11.2010
- [Bru05] Bruce, Kim B.: *Controversy on How to Teach CS 1: A Discussion on the SIGCSE-members Mailing List #*.  
In ACM SIGCSE Bulletin, Volume 37, Issue 2, Juni 2005, ACM, New York, S. 111-117,  
DOI=<http://doi.acm.org/10.1145/1083431.1083477>, geprüft: 10.12.2009,  
siehe auch: <http://listserv.acm.org/archives/sigcse-members.html>, geprüft 22.10.2010
- [BS06] Bennedsen, J. und Schulte, C.: *A Competence Model for Object-Interaction in Introductory Programming*.  
In 18th Workshop of the Psychology of Programming Interest Group, University of Sussex, September 2006, PPIG 2006, <http://www.ppig.org/papers/18th-bennedsen.pdf>, geprüft: 02.04.2010
- [BS07] Bennedsen, J. und Schulte, C.: *What does "Objects-First" Mean? An International Study of Teachers' Perceptions of Objects-First*.  
In Lister, R. und Simon, B. (Hrsg.): Proceedings of the Seventh Baltic Sea Conference on Computing Education Research, Koli Calling 2007, Finnland, CRPIT Vol. 88, S. 21-29,  
ISBN 978-1-920682-69-9, siehe auch: <http://crpit.com/confpapers/CRPITV88Bennedsen.pdf>,  
geprüft: 10.12.2009
- [BSua05] Bergin, J., Stehlik, M., Roberts, J. und Pattis, R.: *Karel J. Robot A Gentle Introduction to the Art of Object-Oriented Programming in Java*.  
Dream Songs Press, Redwood City, Kalifornien, USA, Februar 2005, ISBN 0970579519
- [CBL07] Caspersen, M. E., Bennedsen, J. und Larsen, K. D.: *Mental Models and Programming Aptitude*.  
In Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, Dundee, Schottland, 2007, ITiCSE '07, ACM, New York, S. 206-210,  
DOI=<http://doi.acm.org/10.1145/1268784.1268845>, geprüft: 08.12.2009
- [CH95] Crutzen, C. K. M. und Hein, H.-W.: *Objektorientiertes Denken als didaktische Basis der Informatik*.  
In Schubert, S. (Hrsg.): Innovative Konzepte für die Ausbildung. GI-Fachtagung Informatik und Schule 1995, S. 149-158, Berlin, Informatik Aktuell Springer, 1995, ISBN 3-540-60245-3
- [Chr05] Christensen, Henrik Bærbak: *Implications of Perspective in Teaching Objects First and Object Design*.  
In Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, Caparica, Portugal, Juni 2005, ITiCSE '05, ACM, New York, S. 94-98,  
DOI=<http://doi.acm.org/10.1145/1067445.1067474>, geprüft: 08.12.2009
- [CMS06] Chen, T., Monge, A. und Simon, B.: *Relationship of Early Programming Language to Novice Generated Design*.  
In Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, Houston, Texas, USA, März 2006, SIGCSE '06, ACM, New York, S. 495-499,  
DOI=<http://doi.acm.org/10.1145/1121341.1121496>, geprüft: 08.12.2009
- [Coh88] Cohen, Jacob: *Statistical power analysis for the behavioral sciences*.  
Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1988, 2. Auflage, ISBN 978-0805802835
- [CS01] IEEE Computer Society und Association for Computing Machinery: *Computing Curricula 2001 Computer Science*.  
Final Report, 15. Dezember 2001, The Joint Task Force on Computing Curricula, National Science Foundation, Grant No. 0003263,  
[http://www.acm.org/education/education/education/curric\\_vols/cc2001.pdf](http://www.acm.org/education/education/education/curric_vols/cc2001.pdf), geprüft: 30.09.2010
- [CS08] IEEE Computer Society und Association for Computing Machinery: *Computer Science Curriculum 2008: An Interim Revision of CS2001*.  
Report from the Interim Review Task Force, Dezember 2008,  
<http://www.acm.org/education/curricula/ComputerScience2008.pdf>, geprüft: 30.09.2010
- [DDua10] Diethelm, I., Dörge, C., Hildebrandt, C. und Schulte, C. (Hrsg.): *Didaktik der Informatik – Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik*.  
6. Workshop der GI-Fachgruppe DDI, 16.-17.09.2010 in Oldenburg, GI-Edition - Lecture Notes in Informatics (LNI), P-168, Bonner Köllen Verlag, 2010, ISBN 3-88579-262-8

- [Dec03] Decker, Adrienne: *A TALE OF TWO PARADIGMS*.  
Journal of Computing Sciences in Colleges, Volume 19, Issue 2, Dezember 2003, S. 238-246,  
<http://portal.acm.org/citation.cfm?id=948820>, geprüft: 10.12.2009
- [DFM08] Von Detten, P., Faude, O. und Meyer, T.: *Leitfaden zur statistischen Auswertung von empirischen Studien*.  
Universität Paderborn, Institut für Sportmedizin, Oktober 2008, [http://dsg.uni-paderborn.de/fileadmin/dsg/ab3/Leitfaden\\_zur\\_statistischen\\_Auswertung\\_von\\_Studien.pdf](http://dsg.uni-paderborn.de/fileadmin/dsg/ab3/Leitfaden_zur_statistischen_Auswertung_von_Studien.pdf),  
geprüft: 01.07.2011
- [DFP11] Dietz, A., Frank, J. und Punkenburg, R.: *Objektorientierte Programmierung mit Java im Informatikunterricht*.  
Ankündigung einer Fortbildungsveranstaltung im Rahmen der Regionalen Fortbildungen, Berlin, 2011, Veranstaltung: 11.1-1059, [http://www.fortbildung-regional.de/suchen/veranstaltungsdaten\\_anzeigen.php?pageID=cb&kurseid=29883](http://www.fortbildung-regional.de/suchen/veranstaltungsdaten_anzeigen.php?pageID=cb&kurseid=29883),  
geprüft: 14.03.2011
- [DH94] Decker, R. und Hirshfield, S.: *The Top 10 Reasons Why Object-Oriented Programming Can't Be Taught in CS 1*.  
In Proceedings of the 25th SIGCSE Technical Symposium on Computer Science Education, 1994, ACM, New York, S. 51-55, <http://portal.acm.org/citation.cfm?id=191033.191054>, geprüft: 02.02.2010
- [Die07] Diethelm, Ira: *"Strictly models and objects first" – Unterrichtskonzept und -methodik für objektorientierte Modellierung im Informatikunterricht*.  
Dissertation, Kassel, Universität, 2007, urn:nbn:de:hebis: 34-2007101119340
- [Dob00] Doberkat, Ernst-Erich: *Die Hofzwerge – Ein kurzes Tutorium zur objektorientierten Modellierung*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft 3/4, S. 71-81, LOG IN Verlag, 2000
- [EH08] Edlinger, H. und Hascher, T.: *Von der Stimmungs- zur Unterrichtsforschung: Überlegungen zur Wirkung von Emotionen auf schulisches Lernen und Leisten*.  
In Unterrichtswissenschaft, 36. Jahrgang, 2008, Heft 1, S. 55-70, Juventa Verlag GmbH, Weinhheim
- [Ehl05] Ehlert, Albrecht: *Objektorientierte Programmierung (OOP) und Unified Modeling Language (UML) mit Übungen in Java*.  
Skript, Oktober 2005, 3. Auflage, OSZ IMT, Berlin  
<http://www.ehlert.oszimt.de/dokumente/Java-OOP-Skript.zip>, geprüft: 23.08.2009
- [Ehl06] Ehlert, Albrecht: *Objektorientierte Programmierung im Anfängerunterricht nicht(!) von Anfang an?*  
Vortrag auf dem 5. Berliner MNU-Kongress 2006,  
<http://www.ehlert.oszimt.de/dokumente/MNU-2006-Vortrag-Ehlert.pdf>, geprüft: 30.11.2009
- [Ehl07a] Ehlert, Albrecht: *Einführung in die Programmierungstechnik*.  
Skript, 14.01.2007, 4. Auflage, OSZ IMT, Berlin,  
<http://www.ehlert.oszimt.de/dokumente/JESkript.zip>, geprüft: 23.08.2009
- [Ehl07b] Ehlert, Albrecht: *Studie: Objects-First- und Objects-Later-Einstieg*.  
Praxisband der 12. Fachtagung "Informatik und Schule - INFOS 2007" der Gesellschaft für Informatik e. V., S. 17-20, Herausgeber: Peer Stechert, Universität Siegen 2007,  
ISBN 978-3-936533-23-1,  
siehe auch: <http://www.ehlert.oszimt.de/dokumente/ObjectsFirstLaterEhlert4INFOS2007.pdf>,  
geprüft: 30.11.2009
- [Ehl07c] Ehlert, Albrecht: *Objects Really First?*  
Vortrag auf der 12. GI-Fachtagung 2007, Siegen, 2007,  
<http://www.ehlert.oszimt.de/dokumente/GI-Fachtagung-2007-Ehlert.pdf>, geprüft: 30.11.2009
- [Ehl08a] Ehlert, Albrecht: *Unterschiede im emotional-motivationalen Erleben des Unterrichtsgeschehens und (signifikante) Unterschiede im Lernerfolg von (leistungsschwachen) Schülerinnen und Schülern im Informatik-Anfängerunterricht (11. Klasse OG) in Abhängigkeit von der zeitlichen Reihenfolge der einzelnen Themen (OOP-First bzw. OOP-Later)*.  
Vorstellung des Promotionsvorhabens an der FU Berlin, 2008,  
<http://www.ehlert.oszimt.de/dokumente/FU-2008-Ehlert3.pdf>, geprüft: 30.11.2009

- [Ehl08b] Ehlert, Albrecht: *OOP first oder OOP later - Ergebnisse der Hauptstudie*. Vortrag auf dem 6. Berliner MNU-Kongress 2008, <http://www.ehlert.oszimt.de/dokumente/MNU-2008-Ehlert.pdf>, geprüft: 30.11.2009
- [Els04] Elser, Thomas: *Statistik für die Praxis: Vom Problem zur Methode*. Wiley-VCH Verlag, Weinheim, 1. Auflage, 2004, ISBN 978-3527500970
- [Eng03] Engelmann, Lutz: *Methodenkompetenz im Informatikunterricht*. In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 121, S. 30-36, LOG IN Verlag, 2003
- [Eng09] Engbring, Dieter: *Wozu objektorientiertes Programmieren? Versuch einer Begründung aus der Informatik-Geschichte*. In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 157/158, S. 25-33, LOG IN Verlag, 2009
- [Erl00] Erlenkötter, Helmut: *C++ - Objektorientiertes Programmieren von Anfang an*. Rowohlt Taschenbuch Verlag GmbH, Hamburg, 2. Auflage, 2000, ISBN 978-3499-60077-3
- [ES07a] Ehlert, A. und Schulte, C.: *Learners Views on Objects-First and Objects-Later - Results of an Exploratory Study*. Papier für den 11. Workshop TLOOC auf der ECOOP 2007, Berlin, S. 1-7, <http://www.ehlert.oszimt.de/dokumente/EhlertSchulteLearnersViewsOnObjectsFirstOrObjectsLater.pdf>, geprüft: 30.11.2009, vgl. auch: Böstler J. und Hadar I: *Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts (Report on the 11th Workshop TLOOC at ECOOP 2007)*. ISBN 978-3-540-78194-3, Object-Oriented Technology, ECOOP 2007 Workshop Reader, Lecture Notes in Computer Science, 2008, Volume 4906/2008, S. 182-192, DOI: 10.1007/978-3-540-78195-0\_18, <http://www.springerlink.com/content/912n26256h33h184>, geprüft: 20.10.2010
- [ES07b] Ehlert, A. und Schulte, C.: *Learners Views on Objects-First and Objects-Later - Results of an Exploratory Study*. Vortrag auf der ECOOP 2007, Berlin, <http://www.ehlert.oszimt.de/dokumente/ECOOPworkshoppresentation2.pdf>, geprüft: 30.11.2009
- [ES09a] Ehlert, A. und Schulte, C.: *Empirical comparison of objects first and objects later*. International Computing Education Research ICER'09, Proceedings of the 2009 ACM Workshop, S. 15-26, 10. - 11. August 2009, Berkeley, Kalifornien, USA, ACM Order No 457097, ISBN 978-1-60558-615-1, DOI=<http://doi.acm.org/10.1145/1584322.1584326>, geprüft: 10.12.2009, siehe auch: [http://www.ehlert.oszimt.de/dokumente/ICER09\\_OOP\\_firstOOP\\_later\\_pub\\_version\\_1.pdf](http://www.ehlert.oszimt.de/dokumente/ICER09_OOP_firstOOP_later_pub_version_1.pdf), geprüft: 30.11.2009
- [ES09b] Ehlert, A. und Schulte, C.: *Empirical Comparison of Objects-First and Objects-Later*. Vortrag auf der ICER 2009, Berkeley, Kalifornien, USA, <http://www.ehlert.oszimt.de/dokumente/ICER-2009-Ehlert-END.pdf>, geprüft: 30.11.2009
- [ES09c] Ehlert, A. und Schulte, C.: *Unterschiede im Lernerfolg von Schülerinnen und Schülern in Abhängigkeit von der zeitlichen Reihenfolge der Themen (OOP-First bzw. OOP-Later)*. INFOS 2009, Berlin, In Bernhard Koerber (Hrsg.): *Zukunft braucht Herkunft*, GI-Edition, ISBN 978-3-88579-250-5, S. 121-132, siehe auch: <http://www.ehlert.oszimt.de/dokumente/INFOS2009EhlertOOPFirstLaterCSAE.pdf>, geprüft: 30.11.2009
- [ES09d] Ehlert, A. und Schulte, C.: *Unterschiede im Lernerfolg von Schülerinnen und Schülern in Abhängigkeit von der zeitlichen Reihenfolge der Themen (OOP-First bzw. OOP-Later)*. Vortrag auf der INFOS 2009, Berlin, <http://www.ehlert.oszimt.de/dokumente/INFOS-2009-Ehlert.pdf>, geprüft: 30.11.2009
- [ES10a] Ehlert, A. und Schulte, C.: *Comparison of OOP First and OOP Later – First Results Regarding the Role of Comfort Level*. ITiCSE '10, Proceedings of the fifteenth Annual Conference on Innovation and Technology in Computer Science Education, Bilkent, Ankara, Türkei, 26.-30. Juni 2010, S. 108-112, ISBN 978-1-60558-729-5, <http://doi.acm.org/10.1145/1822090.1822122>, geprüft: 15.07.2010, siehe auch: <http://www.ehlert.oszimt.de/dokumente/fp062-ehlert.pdf>, geprüft: 15.05.2011
- [ES10b] Ehlert, A. und Schulte, C.: *Empirical Comparison of Objects-First and Objects-Later*. ITiCSE '10: Annual Conference on Innovation and Technology in Computer Science Education, Bilkent, Ankara, Türkei, 26.-30. Juni 2010, Vortragsfolien vom 29.06.2010, [http://www.ehlert.oszimt.de/dokumente/ITiCSE\\_2010\\_END.pdf](http://www.ehlert.oszimt.de/dokumente/ITiCSE_2010_END.pdf), geprüft: 28.09.2010



- [Fae07] Faes, Günter: *Einführung in R - Ein Kochbuch zur statistischen Datenanalyse mit R*. Books on Demand GmbH, Norderstedt, 2007, ISBN 978-3-83349-184-9
- [FK05] Fischer, H. und Knapp, T.: *Modellieren im Informatikunterricht der Sekundarstufe I*. In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 135, S. 69-73, LOG IN Verlag, 2005
- [Fri90] Friedrichs, Jürgen: *Methoden empirischer Sozialforschung*. VS Verlag für Sozialwissenschaften, Wiesbaden, 14. Auflage, 1990, ISBN 3-531-22028-4
- [FUB07] Mitteilungen der FU Berlin: *Promotionsordnung des Fachbereichs Mathematik und Informatik der Freien Universität Berlin*. Amtsblatt der Freien Universität Berlin, S. 14-21, 2/2007, 8. Januar 2007, ISSN 0723-0745, siehe auch: <http://www.fu-berlin.de/service/zuvdocs/amtsblatt/2007/ab022007.pdf>, geprüft: 28.11.2009
- [Fül04] Füller, Klaus: *Objektorientiertes Modellieren von Einpersonen-Spielen*. In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 128/129, S. 40-43, LOG IN Verlag, 2004
- [FWua87] Fraser, B. J., Walberg, H. J., Welch, W. W. und Hattie, J. A.: *Synthesis of Educational Productivity Research*. International Journal of Educational Research 11, 1987, S. 145-252
- [Gri08] Gries, David: *A principled approach to teaching OO first*. Proceedings of the 39th SIGCSE technical symposium on Computer science education, 2008, ACM, New York, S. 31-35, DOI=<http://doi.acm.org/1352135.1352149>, geprüft: 01.02.2010
- [Her96] Hermes, Alfred: *OOP im Unterricht – Ein Plädoyer für einen gleitenden Paradigmenwechsel*. In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft 4, S. 29-33, LOG IN Verlag, 1996
- [Heu04] Heubaum, Alfred: *Möglichkeiten und Grenzen maschineller Intelligenz*. In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 128/129, S. 62-79, LOG IN Verlag, 2004
- [HNR07] Hartmann, W., Näf, M. und Reichert, R.: *Informatikunterricht planen und durchführen*. Springer-Verlag Berlin Heidelberg 2006, 1. korrigierter Nachdruck 2007, ISBN 978-3-540-34484-1
- [Hu05] Hu, Chenglie: *Rethinking of Teaching Objects-First*. Education and Information Technologies, Volume 9, September 2004, ACM, New York, S. 209-218, <http://portal.acm.org/citation.cfm?id=1023591>, geprüft: 24.09.2010
- [Hub07] Hubwieser, Peter: *Didaktik der Informatik*. Springer-Verlag Berlin Heidelberg, 3. Auflage, 2007, ISBN 978-3-540-72477-3
- [Hum06] Humbert, Ludger: *Didaktik der Informatik – mit praxiserprobtem Unterrichtsmaterial*. B. G. Teubner Verlag, Wiesbaden, 2. Auflage, 2006, ISBN 978-3-835-10112-8
- [Jac98] Jacobs, Bernard: *Klassifizierungsansätze von Untersuchungen*. 1998, last update 6.3.1998, pf00bj@rz.uni-sb.de, <http://www.phil.uni-sb.de/~jakobs/seminar/vpl/experiment/klassifikation.htm>, geprüft: 5.08.2009
- [KJ03] Kubinger, K. D. und Jäger, R. S. (Hrsg.): *Schlüsselbegriffe der psychologischen Diagnostik*. Beltz PVU, Weinheim, 2003, ISBN 978-3-6212-7472-2
- [Kmk04] Kultusministerkonferenz: *Einheitliche Prüfungsanforderungen Informatik*. Beschluss der Kultusministerkonferenz vom 01.12.1989 i. d. F. vom 05.02.2004, [http://www.kmk.org/fileadmin/veroeffentlichungen\\_beschluesse/1989/1989\\_12\\_01-EPA-Informatik.pdf](http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf), geprüft: 19.09.2010
- [KMua09] Kortenkamp, U., Modrow, E., Oldenburg, R., Poloczek, J. und Rabel, M.: *Objektorientierte Modellierung – aber wann und wie?* In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 160/161, S. 41-47, LOG IN Verlag, 2009
- [Koe04] Koerber, Bernhard: *Von Dingen und Objekten*. In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 128/129, S. 3 (Editorial), LOG IN Verlag, 2004
- [Kro06] Kromrey, Helmut: *Empirische Sozialforschung*. Lucius & Lucius Verlagsgesellschaft mbH, Stuttgart, 2006, 11. Auflage, ISBN 3-8252-1040-5
- [KS07] Krüger, G. und Stark, T.: *Handbuch der Java-Programmierung*. Addison-Wesley, München, November 2007, ISBN 978-3827323736

- [KSD06] Keefe, K., Sheard, J. und Dick, M.: *Adopting XP Practices for Teaching Object Oriented Programming*.  
In Proceedings of the 8th Australian Conference on Computing Education - Volume 52, Hobart, Australien, 16.-19. Januar 2006, S. 91-100,  
<http://portal.acm.org/citation.cfm?id=1151869.1151882>, geprüft: 08.12.2009
- [Küc06] Küchelmann, Stefan: *Gehirngerechtes Lernen*.  
Diplomarbeit, 2006, Mönchengladbach, <http://www.kuechelmann.net/Diplomarbeit.pdf>, geprüft: 23.01.2009
- [LAua04] Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J. E., Sanders, K., Seppälä, O., Simon, B. und Thomas, L.: *A multi-national study of reading and tracing skills in novice programmers*.  
In Working group reports from ITiCSE on Innovation and Technology in Computer Science Education, ITiCSE-WGR '04, Leeds, Großbritannien, ACM, New York, S. 119-150
- [LAJ05] Lahtinen, E., Ala-Mutka, K. und Järvinen, H.-M.: *A Study of the Difficulties of Novice Programmers*.  
ACM SIGCSE Bulletin, Volume 37, S. 14-18, September 2005, ACM, New York,  
<http://portal.acm.org/citation.cfm?id=1067453>, geprüft: 10.12.2009
- [Lau75] Lauer, H.C.: *Discussion - On Ph.D. thesis proposals in computing science*.  
Computing Laboratory, University of Newcastle upon Tyne,  
The Computer Journal, Volume 18, Number 3, 1975, S. 279-281,  
<http://comjnl.oxfordjournals.org/cgi/reprint/18/3/279>, geprüft: 10.12.2009, siehe auch:  
<https://www.mi.fu-berlin.de/wiki/pub/Mi/PromotionsVerfahren/Lauer75.pdf>, geprüft: 23.11.2009
- [Lav04] Lavergne, Harro v.: *Objekte, Klassen, Module, Kontrakte und Komponenten*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 131/132, S. 46-55, LOG IN Verlag, 2004
- [Lav05] Lavergne, Harro v.: *Über Züge, Ampeln und Objekte – oder: Das A und O der OOA ist die Analyse*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 136/137, S. 54-60, LOG IN Verlag, 2005
- [LBua06] Lister, R., Berglund, A., Clear, T., Bergin, J., Garvin-Doxas, K., Hanks, B., Hitchner, L., Luxton-Reilly, A., Sanders, K., Schulte, C. und Whalley, J. L.: *Research Perspectives on the Objects-Early Debate*.  
In Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education, Bologna, Italien, 26.-28. Juni 2006, ACM, New York, S. 146-165,  
DOI=<http://doi.acm.org/10.1145/1189215.1189183>, geprüft: 10.12.2009, siehe auch:  
<http://listserv.acm.org/archives/sigcse-members.html>, geprüft 22.10.2010
- [Lei04] Leipholtz-Schumacher, Barbara: *Objektorientiertes Modellieren und Programmieren*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 128/129, S. 32-39, LOG IN Verlag, 2004
- [LI04] Fachbereich Erziehung, Wissenschaft und Psychologie an der Freien Universität Berlin (Hrsg.): *Objektorientiertes Programmieren und Modellieren*.  
LOG IN, Heft Nr. 128/129, LOG IN Verlag, 2004
- [LI09] Fachbereich Erziehung, Wissenschaft und Psychologie an der Freien Universität Berlin (Hrsg.): *LOG IN Register 1981-2009*.  
LOG IN Verlag, 2009, siehe auch [http://www.log-in-verlag.de/PDF-Dateien/Antiquarische&Register/Register\\_1981-2009.pdf](http://www.log-in-verlag.de/PDF-Dateien/Antiquarische&Register/Register_1981-2009.pdf), geprüft: 22.10.2010
- [LR98] Lienert, G. A. und Raatz, U.: *Testaufbau und Testanalyse*.  
BELTZ Psychologie Verlags Union, Weinheim, 6. Auflage 1998, ISBN 978-3-621-27424-1
- [LSua06] Lister, R., Simon, B., Thompson, E., Whalley, J. L. und Prasad, C.: *Not seeing the forest for the trees: novice programmers and the SOLO taxonomy*.  
In ITiCSE '06, Proceedings of the 11th annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ACM Press, New York, S. 118-122,  
DOI=<http://doi.acm.org/10.1145/1140124.1140157>, geprüft: 20.07.2011
- [NDH02] Nieschlag, R., Dichtl, E. und Hörschgen, H.: *Marketing*.  
19. Auflage 2002, Duncker & Humblot, Berlin, ISBN 978-3428109302

- [MD04] McKinney, D. und Denton, L. F.: *Houston, we have a problem: there's a leak in the CS1 affective oxygen tank*.  
In Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, Norfolk, Virginia, USA, 03.03 – 07.03.2004, SIGCSE '04, ACM, New York, S. 236-239,  
DOI=<http://doi.acm.org/10.1145/971300.971386>, geprüft: 18.01.2010
- [Mis05] Misch, Jens-Peter: *Java 4 U – Programmentwicklung mit Java*.  
Bildungsverlag EINS, Troisdorf, 2005, ISBN 3-427-01144-5
- [MK08] Moosbrugger, H. und Kelava, A.: *Testtheorie und Fragebogenkonstruktion*.  
Springer Medizin Verlag, Heidelberg, 2008, ISBN 987-3-540-71634-1
- [MR02] Milne, I. und Rowe, G.: *Difficulties in Learning and Teaching Programming - Views of Students and Tutors*.  
Education and Information Technologies, Volume 7, Number 1, Kluwer Academic Publisher, März 2002, S. 55-66
- [MR06] Mannila, L. und de Raadt, M.: *An Objective Comparison of Languages for Teaching Introductory Programming*.  
In Proceedings of the 6th Baltic Sea Conference on Computing Education Research, Koli Calling 2006, Uppsala, Schweden, Februar 2006, Baltic Sea '06, Volume 276, ACM, New York, S. 32-37,  
DOI=<http://doi.acm.org/10.1145/1315803.1315811>, geprüft: 08.12.2009
- [Mül08] Müllener, Jenna: *Nachhaltiges Lernen ist lustvoll, sinnvoll, positiv*.  
Fachzeitschrift "4 bis 8", Nr. 9, S. 16-17, Schulverlag plus AG Bern (Hrsg.), September 2008,  
siehe auch: [http://www.ppz.ch/pdf/nachhaltiges\\_lernen.pdf](http://www.ppz.ch/pdf/nachhaltiges_lernen.pdf), geprüft: 23.11.2009
- [MWua05] Moritz, S. H., Wei, F., Parvez, S. M. und Blank, G. D.: *From Objects-First to Design-First with Multimedia and Intelligent Tutoring*.  
Proceedings of the 10th annual SIGCSE Conference on Innovation and Technology in Computer Science Education, Caparica, Portugal, 2005, ACM, S. 99-103,  
DOI=<http://portal.acm.org/citation.cfm?id=1067445.1067475>, geprüft: 08.12.2009
- [Old10] Oldenburg, Reinhard: *Klassensysteme selbstgebaut. Ein genetischer Weg zu OOP und OOM*.  
In LOG IN, Hrsg.: FB ErzwissPsychFUB, Heft Nr. 166/167, S. 85-91, LOG IN Verlag, 2010
- [Pet10] Peters, Bernhard: *Hundert Spiele mit einem Klick – Wie man Spieler und Mannschaften mit wissenschaftlichen Methoden verbessern kann*.  
Artikel in DER TAGESSPIEGEL, Verlag Der Tagesspiegel GmbH, Berlin, 26. Juli 2010, Seite 16, NR. 20687
- [PFS10] Penon, J., Frank, J. und Schindler, A.: *Workshop 6: Einführung in die objektorientierte Programmierung*.  
Ankündigung eines Workshops auf dem Informatiktag Berlin-Brandenburg 2010,  
[http://bildungsserver.berlin-brandenburg.de/informatiktag\\_ws6.html](http://bildungsserver.berlin-brandenburg.de/informatiktag_ws6.html), geprüft: 15.02.2010
- [Pre09] Prechelt, Lutz (Hrsg.): *Promotion am Fachbereich Mathematik und Informatik, Informationsseite der FU Berlin*.  
Freie Universität Berlin, Revision 23, 22 Juni 2009,  
<https://www.mi.fu-berlin.de/w/Mi/PromotionsVerfahren>, geprüft: 31.01.2010
- [PSua07] Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M. und Paterson, J.: *A survey of literature on the teaching of introductory programming*.  
In Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education, Dundee, Schottland, Dezember 2007, ITiCSE-WGR '07, ACM, New York, S. 204-223,  
DOI=<http://doi.acm.org/10.1145/1345443.1345441>, geprüft: 02.02.2010
- [PTK10] Porsch, R., Tesch, B. und Köller, O. (Hrsg.): *Standardbasierte Testentwicklung und Leistungsmessung*.  
Waxmann Verlag GmbH, Münster, 2010, ISBN 978-3-8309-2274-2
- [PW05] Pallmer, J. und Wachler, K.-H.: *Microsoft Visual C#.NET 2003 für Windows – Grundlagen*.  
Schulungsunterlage (SPC the campus), HERDT-Verlag, Bodenheim, 1. Auflage, 15.12. 2005
- [Reg06] Reges, Stuart: *Back to basics in CS1 and CS2*.  
ACM SIGSE Bulletin, SIGCSE '06, ACM, New York, 2006, S. 293-297,  
DOI=<http://doi.acm.org/1121341.1121432>, geprüft 01.02.2010

- [Röh09] Röhner, Gerhard: *Objektorientierte Modellierung mit dem JAVA-Editor*. In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 159, S. 60-66, LOG IN Verlag, 2009
- [Röh10] Röhner, Gerhard: *JAVA-Editor*. Informationen unter <http://www.javaeditor.org>, geprüft: 02.11.2010, Download unter [http://www.heise.de/software/download/java\\_editor/61374](http://www.heise.de/software/download/java_editor/61374), geprüft: 02.11.2010
- [RRR03] Robins, A., Rountree, J. und Rountree, N.: *Learning and Teaching Programming: A Review and Discussion*. Computer Science Education, Volume 13, Issue 2, 2003, S. 137–172
- [RVB01] Rheinberg, F., Vollmeyer, R. und Burns, B. D.: *FAM: Ein Fragebogen zur Erfassung aktueller Motivation in Lern- und Leistungssituationen*. In Diagnostica, Hogrefe Verlag, Göttingen, Band 47, Nummer 2, 2001, S. 57-66
- [Sav01] Savitch, Walter: *Java: An Introduction to Computer Science & Programming*. Zweite Auflage, Prentice Hall, New Jersey, USA, 2001, ISBN 978-0130316974
- [Sch95] Schwill, Andreas: *Programmierstile im Anfangsunterricht*. In Schubert, S. (Hrsg.): *Innovative Konzepte für die Ausbildung*, S. 178-187, 7. GI-Fachtagung Informatik und Schule, INFOS'95, Chemnitz, 25-28. September 1995, ISBN 3-540-60245-3
- [Sch01] Schübler, Ingeborg: *Nachhaltiges Lernen*. Grundlagen der Weiterbildung - Praxishilfen, Loseblattsammlung, 2001, Luchterhand, Neuwied, S. 1-27, siehe auch: [http://www.die-bonn.de/portrait/aktuelles/DIE\\_Forum\\_2005\\_Schuessler\\_NachhaltigesLernen.pdf](http://www.die-bonn.de/portrait/aktuelles/DIE_Forum_2005_Schuessler_NachhaltigesLernen.pdf), geprüft: 22.02.2010
- [Sch03a] Schlittgen, Rainer: *Einführung in die Statistik*. Oldenbourg Wissenschaftsverlag, München, 2003, 10. Auflage, ISBN 3-486-27446-5
- [Sch03b] Schulte, Carsten: *Lehr- Lernprozesse im Informatik-Anfangsunterricht, Theoriegeleitete Entwicklung und Evaluation eines Unterrichtskonzepts zur Objektorientierung in der Sekundarstufe II*. Dissertation, Paderborn, 2003, <http://ddi.uni-muenster.de/Examensarbeiten/Schulte2003.pdf>, geprüft: 7.12.2009
- [Sch08] Schulte, Carsten: *Block Model: an educational model of program comprehension as a tool for a scholarly approach to teaching*. In Proceeding of the Fourth International Workshop on Computing Education Research, Sydney, Australien, September 2008, ICER '08, ACM, New York, S. 149-160, DOI=<http://doi.acm.org/10.1145/1404520.1404535>, geprüft: 29.09.2010
- [Sch09] Scholl, Arnim: *Die Befragung*. UVK Verlagsgesellschaft mbH, Konstanz, 2. Auflage, 2009, ISBN 978-3-8252-2413-4
- [Sei04] Seifried, Jürgen: *Fachdidaktische Variationen in einer selbstorganisationsoffenen Lernumgebung, Eine empirische Untersuchung im Rechnungswesenunterricht*. Dissertation Universität Bamberg, Deutscher Universitäts Verlag (DUV), 1. Auflage, 2004, ISBN 3-8244-0753-1
- [Sem92] Sembill, Detlef: *Problemlösefähigkeit, Handlungskompetenz und Emotionale Befindlichkeit*. Verlag Hogrefe, Göttingen, 1992, Serie: Ergebnisse der pädagogischen Psychologie, Band 10, ISBN 3-8017-0434-3
- [Sem05] Sembill, Detlef: *Emotional-motivationales Lernen in berufsbildenden Lernprozessen*. Otto-Friedrich-Universität Bamberg, 13.01.2005, [http://www.ewi-psy.fu-berlin.de/einrichtungen/arbeitsbereiche/allg\\_paedagogik/media/izll/06\\_sebill.pdf](http://www.ewi-psy.fu-berlin.de/einrichtungen/arbeitsbereiche/allg_paedagogik/media/izll/06_sebill.pdf), geprüft: 29.01.2010
- [SH09] Sachs, L. und Hedderich, J.: *Angewandte Statistik*. Springer Verlag, Berlin, Heidelberg, 2009, 13. Auflage, ISBN 978-3-540-88901-4
- [SHE08] Schnell, R., Hill, P. B. und Esser, E.: *Methoden der empirischen Sozialforschung*. Oldenbourg Wissenschaftsverlag GmbH, 2008, 8. Auflage, ISBN 978-3-486-58708-1
- [SIG10] SIGCSE (ACM Special Interest Group on Computer Science Education) Member Forum: *Archives of sigcse-members@ACM.ORG*. <http://listserv.acm.org/archives/sigcse-members.html>, geprüft: 23.10.2010, siehe auch <http://www.sigcse.org>, geprüft: 23.10.2010

- [SK06] Seifried, J. und Klüber, C.: *Unterrichtserleben in schüler- und lehrerzentrierten Unterrichtsphasen*.  
2006, Unterrichtswissenschaft 34, 1, S. 2-21, Juventa Verlag, Weinheim,  
siehe auch: <http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-71130>, geprüft: 9.08.2009
- [SKT08] Sajaniemi, J., Kuittinen, M. und Tikansalo, T.: *A Study of the Development of Students' Visualizations of Program State during an Elementary Object-Oriented Programming Course*.  
Journal on Educational Resources in Computing Volume 7, Issue 4, Januar 2008, S. 1-31,  
DOI=<http://doi.acm.org/10.1145/1316450.1316453>, Stand 08.12.2009
- [Spo95] Spolwig, Siegfried: *Objektbasierte Programmierung im Anfangsunterricht*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft 3, S. 43-49, LOG IN Verlag, 1995
- [Spo01] Spolwig, Siegfried: *Methodische Probleme mit OOP im Anfangsunterricht*.  
[http://oszhdl.be.schule.de/gymnasium/faecher/informatik/didaktik/oop\\_probleme.htm](http://oszhdl.be.schule.de/gymnasium/faecher/informatik/didaktik/oop_probleme.htm), 2001, geprüft: 03.10.2009
- [Spo04] Spolwig, Siegfried: *Kritisches zu "Stiften und Mäusen" – Was ist objektorientierte Modellierung?*  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 130, S. 35-39, LOG IN Verlag, 2004
- [Spo05] Spolwig, Siegfried.: *Karel D. Robot – Der Delphi Karel*.  
OSZ Handel, Berlin, 2005,  
[http://www.oszhandel.de/gymnasium/faecher/informatik/delphi\\_karel/index.htm](http://www.oszhandel.de/gymnasium/faecher/informatik/delphi_karel/index.htm), geprüft: 29.09.2010
- [SS04] Schubert, S. und Schwill, A.: *Didaktik der Informatik*.  
Spektrum Akademischer Verlag, Heidelberg, 2004, ISBN 3-8274-1382-6
- [SS11] Schubert, S. und Schwill, A.: *Didaktik der Informatik*.  
Spektrum Akademischer Verlag, Heidelberg, 2. Auflage 2011, ISBN 978-3-8274-2652-9
- [SSua09] Sheard, J., Simon, S., Hamilton, M. und Lönnberg, J.: *Analysis of research into the teaching and learning of programming*.  
Proceedings of the fifth International Workshop on Computing Education Research, Berkeley, Kalifornien, USA, ICER '09, S. 93-104, <http://portal.acm.org/citation.cfm?id=1584322.1584334>, geprüft 02.04.2010
- [Ste04] Steinbrucker, Christian: *Objektorientierung im Anfangsunterricht!!! Simulation einer Taschenlampe mit DELPHI*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 131/132, S. 56-61, LOG IN Verlag, 2004
- [Ste07] Stechert, Peer (Herausgeber, Vorwort): *Praxisband der 12. Fachtagung "Informatik und Schule - INFOS 2007"*.  
Universität Siegen, 2007, S. 17–20, ISBN 978-3-936533-23-1, [http://www.die.informatik.uni-siegen.de/lehrstuhl/stechert/publikationen/Informatische\\_Bildung\\_2007.pdf](http://www.die.informatik.uni-siegen.de/lehrstuhl/stechert/publikationen/Informatische_Bildung_2007.pdf), geprüft: 07.08.2009
- [Thi08] Thiele, Otto: *Objektorientierte Programmierung von Schachproblemen*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 150/151, S. 66-68, LOG IN Verlag, 2008
- [Tho04] Thomas, Marco: *Objektorientierung und informatische Bildung*.  
In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 128/129, S. 26-31, LOG IN Verlag, 2004
- [TMG05] Tew, A. E., McCracken, W. M. und Guzdial, M.: *Impact of alternative introductory courses on programming concept understanding*.  
In Proceedings of the First International Workshop on Computing Education Research, Seattle, Washington, USA, Oktober 2005, ICER '05, ACM, New York, S. 25-35,  
DOI=<http://doi.acm.org/10.1145/1089786.1089789>, geprüft: 29.09.2010
- [Ven04] Ventura, Philip R.: *On the Origins of Programmers: Identifying Predictors of Success for an Objects First CS1*.  
Dissertation, UMI Order Number: AAI3113537, State University of New York at Buffalo, 2004,  
<http://portal.acm.org/citation.cfm?id=997592#>, geprüft 29.06.2010
- [VZG07] Vilner, T., Zur, E. und Gal-Ezer, J.: *Fundamental concepts of CS1: procedural vs. object oriented paradigm - a case study*.  
ACM SIGCSE Bulletin, Volume 39, Issue 3, Juni 2007, S. 171-175,  
DOI=<http://doi.acm.org/10.1145/1269900.1268835>, geprüft: 01.02.2010

- [Weh03] Wehrheim, Otto: *Objektorientiertes Modellieren mit DELPHI – am Beispiel des Memory-Spiels*. In LOG IN, Hrsg.: FB ErzWissPsychFUB, Heft Nr. 125, S. 30-35, LOG IN Verlag, 2003
- [WRua99] Wiedenbeck, S., Ramalingam, V., Sarasamma, S. und Corritore, C.: *A comparison of the comprehension of object-oriented and procedural programs by novice programmers*. Interacting With Computers, Volume 11, Issue 3, 1999, S. 255-282
- [WS01] Wilson, B. C. und Shrock, S.: *Contributing to success in an introductory computer science course: a study of twelve factors*. ACM SIGCSE Bulletin, Volume 33, Issue 1, März 2001, ACM, New York, S. 184-188, DOI=<http://doi.acm.org/10.1145/366413.364581>, geprüft: 18.01.2010
- [Zor09] Zorn, Silke: *Die größte IT-Schule der Stadt*. Artikel in DER TAGESSPIEGEL, Verlag Der Tagesspiegel GmbH, Berlin, 15.09.2009, <http://www.tagesspiegel.de/berlin/familie/schule/art295.2899725>, geprüft: 03.10.2009
- [ZS08] Zendler, A. und Spannagel, C.: *Empirical Foundation of Central Concepts for Computer Science Education*. In Journal of Educational Resources in Computing, Volume 8, Issue 2, 2008, Article No. 6, <http://portal.acm.org/citation.cfm?id=1362790>, geprüft: 17.04.2010
- [ZZ03] Zhu, H. und Zhou, M.: *Methodology First and Language Second: A Way to Teach Object-Oriented Programming*. In Companion of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, Anaheim, Kalifornien, USA, Oktober 2003, OOPSLA '03, ACM, New York, S. 140-147, DOI=<http://doi.acm.org/10.1145/949344.949389>, geprüft: 08.12.2009

# Abbildungsverzeichnis

Abbildung 1: Themen von 164 veröffentlichten Papieren mit Bezug zu Untersuchungen von 6 verschiedenen internationalen Konferenzen zur informatischen Bildung in den Jahren 2005-2008 .....	5
Abbildung 2: Thema Iteration beim OOP-First-Ansatz .....	8
Abbildung 3: Thema Iteration beim OOP-Later-Ansatz .....	8
Abbildung 4: Typische Themenfolge von OOP-First und OOP-Later .....	11
Abbildung 5: Gleiche Themen bei OOP-First und OOP-Later .....	11
Abbildung 6: „Klassendiagramm“ für eine Hofzweigen-Modellierung .....	17
Abbildung 7: „Schulmodell“ als Musterlösung einer OOM-Aufgabe .....	18
Abbildung 8: Objektorientierung im Anfangsunterricht (Modellierung einer Taschenlampe) .....	19
Abbildung 9: Wirkungen von positiven Stimmungen auf kognitive Prozesse .....	26
Abbildung 10: Anfrage an Prof. Balzert und seine Antwort .....	33
Abbildung 11: Inhalte des Informatikunterrichts (früher und heute) .....	34
Abbildung 12: Erste Gedanken zu den Problemen im OOP-Anfangsunterricht .....	34
Abbildung 13: Erstes Resümee am Beginn der Studie .....	35
Abbildung 14: These 1 und These 2 aus dem Jahr 2006 .....	35
Abbildung 15: These 3 und These 4 aus dem Jahr 2006 .....	35
Abbildung 16: Beispiel für die Komplexität der ersten BlueJ-Übung .....	36
Abbildung 17: Visualisierung der Anzahl der BlueJ-Teilthemen bei OOP-First .....	36
Abbildung 18: Visualisierung der Anzahl der BlueJ-Teilthemen bei OOP-Later .....	37
Abbildung 19: Erste schematische Veranschaulichung der 3 Thesen .....	38
Abbildung 20: Zweite schematische Veranschaulichung der 3 Thesen .....	38
Abbildung 21: Gebäudeansicht des Oberstufenzentrums Informations- und Medizintechnik in Berlin .....	48
Abbildung 22: Der erste Abiturjahrgang am OSZ IMT im Jahr 2005 .....	48
Abbildung 23: Einzel- und Partnerarbeit im Computerlabor am OSZ IMT .....	48
Abbildung 24: Design der Hauptstudie .....	50
Abbildung 25: Phasendarstellung der Vorstudie und der Hauptstudie .....	51
Abbildung 26: Klassenstatistik für die Hauptstudie 2007/2008 .....	54

Abbildung 27: Ergebnisse im Hinblick auf die Computervorkenntnisse .....	55
Abbildung 28: Pseudonymisierung des Schülernamens .....	57
Abbildung 29: Reihenfolge der OOP-First-Themen .....	60
Abbildung 30: Reihenfolge der OOP-Later-Themen .....	60
Abbildung 31: Geplante Reihenfolge der Themen bzw. geplante Befragungstermine in der Hauptstudie .....	60
Abbildung 32: Teil eines allgemeinen Informationsblatts (hier Thema 8: Vererbung) .....	61
Abbildung 33: Teil eines Informationsblatts zur Java-Implementierung (hier Thema 8: Vererbung) .....	62
Abbildung 34: Auszug aus einem typischen Arbeitsblatt (hier Thema 8: Vererbung) .....	62
Abbildung 35: Kurzarbeitsplan (1. Halbjahr 2007/2008) .....	65
Abbildung 36: Kurzarbeitsplan (2. Halbjahr 2007/2008) .....	65
Abbildung 37: Themen und Themenschwerpunkte im Schuljahr 2007/2008) für die beiden Lehrer .....	66
Abbildung 38: Mehrdimensionalität des Qualitätsmerkmals „Nachhaltigkeit“ .....	69
Abbildung 39: Befragungstermine zu den einzelnen Modulen in der OOP-First-Klasse .....	72
Abbildung 40: Befragungstermine zu den einzelnen Modulen in der OOP-Later-Klasse .....	72
Abbildung 41: Cut-off-Kriterien für die Umfragen .....	75
Abbildung 42: Cut-off-Kriterien für die Tests .....	76
Abbildung 43: Anforderungsbereiche und Punkteverteilung für die Indikatoren des Themas T2 .....	80
Abbildung 44: Lernziele und Indikatoren für das Thema 2 (Java-Editor, Attribute, Datentypen, Sequenz) .....	82
Abbildung 45: Lernziele und Indikatoren für das Thema 5 (Selektion) .....	83
Abbildung 46: Lernziele und Indikatoren für das Thema 6 (Iteration) .....	84
Abbildung 47: Lernziele und Indikatoren für das Thema 7 (Arrays und Strings) .....	85
Abbildung 48: Lernziele und Indikatoren für das Thema 4 (Methoden) .....	86
Abbildung 49: Lernziele und Indikatoren für die Themen 1 und 3 (OOP-Einführung / Klasse und Objekt) .....	87
Abbildung 50: Lernziele und Indikatoren für das Thema 8 (Vererbung) .....	88
Abbildung 51: Lernziele und Indikatoren für das Thema 9 (Assoziation) .....	89
Abbildung 52: Vorgegebene Java-Klassen für das Zusatzthema Z1 (dynamische OOP) .....	91
Abbildung 53: Lernziele und Indikatoren für das Zusatzthema Z1 (dynamische OOP) .....	92
Abbildung 54: Lernziele und Indikatoren für das Zusatzthema Z2 (OOM) .....	93



Abbildung 55: Vergleichstests vor der Studie (aus dem Jahr 2004) .....	94
Abbildung 56: Dynamische Entwicklung des Vergleichstests im Laufe der Studie .....	94
Abbildung 57: Vergleichstest-Fragen in der Vor- und Hauptstudie (Teil 1) .....	95
Abbildung 58: Vergleichstest-Fragen in der Vor- und Hauptstudie (Teil 2) .....	96
Abbildung 59: Punkte aus Aufgaben des Vergleichstests der Vorstudie in Aufgaben der Hauptstudie .....	97
Abbildung 60: Themen im Vergleichstest der Hauptstudie im Kontext zu den Anforderungsbereichen .....	98
Abbildung 61: Themen des Vergleichstests der Hauptstudie nach Schwierigkeit (AFII) geordnet .....	99
Abbildung 62: Zeitlicher Rahmen des Vergleichstests der Hauptstudie .....	100
Abbildung 63: Exemplarische Beispiele der Unterschiede zwischen Nachhaltigkeits- und Vergleichstest .....	103
Abbildung 64: Nachhaltigkeitstest im Kontext zum Vergleichstest (Teil 1) .....	104
Abbildung 65: Nachhaltigkeitstest im Kontext zum Vergleichstest (Teil 2) .....	105
Abbildung 66: Itemschwierigkeit der 49 Aufgaben des Vergleichstests (Hauptstudie, OOP-Later) .....	108
Abbildung 67: Itemschwierigkeit der 49 Aufgaben des Vergleichstests (Hauptstudie, OOP-Later) .....	109
Abbildung 68: Itemschwierigkeit der Aufgaben des Vergleichstests (Hauptstudie, OOP-Later) .....	110
Abbildung 69: Itemschwierigkeit der 48 Aufgaben des Nachhaltigkeitstests (Hauptstudie) .....	110
Abbildung 70: Erlebnis-Dimensionen und Frage-Bereiche als Matrix .....	115
Abbildung 71: Beispiel einer Skalierung einer Antwort .....	117
Abbildung 72: Umsetzung der Skalierung in eine Intervallskala .....	117
Abbildung 73: Fragen zur Schule und zum privaten Umfeld (Emotionale Dimension) .....	118
Abbildung 74: Fragen zum Fach Informatik (Emotionale Dimension) .....	119
Abbildung 75: Fragen zum Lehrer (Emotionale Dimension) .....	119
Abbildung 76: Fragen zum letzten Thema (Emotionale Dimension) .....	120
Abbildung 77: Fragen zur emotionalen Dimension .....	120
Abbildung 78: Frage zur Schule (Kognitive Dimension) .....	121
Abbildung 79: Frage zum Fach Informatik (Kognitive Dimension) .....	121
Abbildung 80: Frage zum Lehrer (Kognitive Dimension) .....	121
Abbildung 81: Fragen zum letzten Thema (Kognitive Dimension) .....	122
Abbildung 82: Fragen zur kognitiven Dimension .....	122

Abbildung 83: Frage zur Schule und zum privaten Umfeld (Motivationale Dimension) .....	123
Abbildung 84: Frage zum Fach Informatik (Motivationale Dimension) .....	123
Abbildung 85: Fragen zum letzten Thema (Motivationale Dimension) .....	123
Abbildung 86: Fragen zur motivationalen Dimension .....	124
Abbildung 87: Fragen im Kontext zu den Erlebnis-Dimensionen und den Frage-Bereichen .....	124
Abbildung 88: Statistische Verfahren zum Testen von Hypothesen .....	128
Abbildung 89: Anwendung des t-Tests (Welch Two Sample t-Test) mit Hilfe des Statistik-Tools R .....	128
Abbildung 90: Ergebnisse des Vergleichstests in der Vorstudie 2006/2007 .....	130
Abbildung 91: Darstellung der Ergebnisse auf der ECOOP-Konferenz 2007 .....	131
Abbildung 92: Auswertbare Fragebögen im Hinblick auf die Modul-Themen .....	133
Abbildung 93: Ergebnisse zur Frage 2 (OOP-First: $\bar{x} = 2,0$ / OOP-Later: $\bar{x} = 2,1$ ) .....	134
Abbildung 94: Ergebnisse zur Frage 4 (OOP-First: $\bar{x} = 2,4$ / OOP-Later: $\bar{x} = 2,3$ ) .....	134
Abbildung 95: Ergebnisse zur Frage 5 (OOP-First: $\bar{x} = 2,4$ / OOP-Later: $\bar{x} = 2,5$ ) .....	135
Abbildung 96: Ergebnisse zur Frage 3 (OOP-First: $\bar{x} = 1,8$ / OOP-Later: $\bar{x} = 2,2$ ) .....	135
Abbildung 97: Ergebnisse zur Frage 6 (OOP-First: $\bar{x} = 2,4$ / OOP-Later: $\bar{x} = 2,5$ ) .....	136
Abbildung 98: Ergebnisse zur Frage 7 (OOP-First: $\bar{x} = 2,6$ / OOP-Later: $\bar{x} = 2,6$ ) .....	136
Abbildung 99: Ergebnisse zur Frage 8 (OOP-First: $\bar{x} = 2,2$ / OOP-Later: $\bar{x} = 2,2$ ) .....	136
Abbildung 100: Ergebnisse zur Frage 9 (OOP-First: $\bar{x} = 2,5$ / OOP-Later: $\bar{x} = 2,6$ ) .....	137
Abbildung 101: Schwere-Empfinden der OOP-First-Klasse .....	138
Abbildung 102: Schwere-Empfinden der OOP-Later-Klasse .....	138
Abbildung 103: Vergleich des Schwere-Empfindens mit Hilfe eines Box-Whisker-Plots .....	139
Abbildung 104: Subjektiver Schwierigkeitsfaktor und objektives Testergebnis .....	140
Abbildung 105: Ergebnisse des Vortests am Ende der Hauptstudie im Spätsommer 2008 .....	143
Abbildung 106: Ergebnisse des Vergleichstests in der Hauptstudie 2007/2008 .....	145
Abbildung 107: Ergebnisse des Nachhaltigkeitstests in der Hauptstudie 2007/2008 .....	146
Abbildung 108: Vergleichs- und Nachhaltigkeitstest in der Hauptstudie 2007/2008 .....	147
Abbildung 109: Ergebnisse der Tests in der Hauptstudie 2007/2008 .....	148
Abbildung 110: Lernerfolg in der Hauptstudie 2007/2008 bezogen auf die ersten acht Themengebiete .....	148

Abbildung 111: Ergebnisse der leistungsstärksten Schüler .....	150
Abbildung 112: Ergebnisse der leistungsschwächsten Schüler .....	150
Abbildung 113: Ergebnisse der Schülerinnen .....	151
Abbildung 114: Fragen im Kontext zu den Erlebnis-Dimensionen und den Frage-Bereichen .....	154
Abbildung 115: Subjektives Erleben im Kontext zu den Erlebnis-Dimensionen und den Frage-Bereichen .....	154
Abbildung 116: Subjektives Erleben im Kontext zur emotionalen Dimension und zu den Frage-Bereichen .....	155
Abbildung 117: Das emotionale Erleben der OOP-First-Klasse .....	156
Abbildung 118: Das emotionale Erleben der OOP-Later-Klasse .....	157
Abbildung 119: Emotionales Erleben in Abhängigkeit von den Themen .....	158
Abbildung 120: Rangfolge des „emotionalen Erlebens“ (Wohlfühl- und Stressfaktor) der Themen .....	159
Abbildung 121: Direkte Frage nach dem Fach (Frage 6) und Stress- und Wohlfühlfaktor bei OOP-First .....	160
Abbildung 122: Direkte Frage nach dem Fach (Frage 6) und Stress- und Wohlfühlfaktor bei OOP-Later .....	160
Abbildung 123: Subjektives Erleben im Kontext zur kognitiven Dimension und zu den Frage-Bereichen .....	161
Abbildung 124: Das kognitive Erleben der OOP-First-Klasse .....	163
Abbildung 125: Das kognitive Erleben der OOP-Later-Klasse .....	164
Abbildung 126: Kognitives Erleben (alle sieben Items) in Abhängigkeit von den Themen .....	164
Abbildung 127: Kognitive Dimension (mit sieben Items) aufgegliedert in 3 Unterbereiche für OOP-First .....	166
Abbildung 128: Kognitive Dimension (mit sieben Items) aufgegliedert in 3 Unterbereiche für OOP-Later .....	167
Abbildung 129: „Empfundene Schwere“ (mit nur einem Item: Frage 12) der einzelnen Themen .....	168
Abbildung 130: Schwere-Empfinden der OOP-First-Klasse in Abhängigkeit der Themenfolge .....	169
Abbildung 131: Schwere-Empfinden der OOP-Later-Klasse in Abhängigkeit der Themenfolge .....	170
Abbildung 132: Vergleich des Schwere-Empfindens mit Hilfe eines Box-Whisker-Plots .....	170
Abbildung 133: Schwere-Empfinden des Themas „Methoden“ in der Vor- und Hauptstudie .....	171
Abbildung 134: Subjektives Erleben im Kontext zur motivationalen Dimension und zu den Frage-Bereichen .....	172
Abbildung 135: Das motivationale Erleben der OOP-First-Klasse .....	172
Abbildung 136: Das motivationale Erleben der OOP-Later-Klasse .....	173
Abbildung 137: Motivationales Erleben in Abhängigkeit von den Themen .....	174
Abbildung 138: Subjektives Erleben in allen vier Frage-Bereichen .....	175


Abbildung 139: Subjektives Erleben in Abhängigkeit vom Fach und von den Themen .....	176
Abbildung 140: Subjektives Erleben der leistungsstärksten Schülerinnen und Schüler .....	177
Abbildung 141: Subjektives Erleben der leistungsschwächsten Schülerinnen und Schüler .....	178
Abbildung 142: Subjektives Erleben der Schülerinnen .....	179
Abbildung 143: Geplante und tatsächliche Reihenfolge der Themen .....	186
Abbildung 144: Geplante und tatsächliche Termine der Befragungen zum subjektiven Erleben .....	186
Abbildung 145: Beurteilung der beiden Lehrer .....	188
Abbildung 146: Beurteilung der Lehrer in Abhängigkeit von den Themen .....	189
Abbildung 147: Beurteilung des OOP-First-Lehrers (mit vier Indikatoren) in Abhängigkeit von den Themen	189
Abbildung 148: Beurteilung des OOP-Later-Lehrers (mit vier Indikatoren) in Abhängigkeit von den Themen	190
Abbildung 149: Beurteilung der Rahmenbedingungen des Unterrichts .....	191
Abbildung 150: Beurteilung der Zufriedenheit mit dem Lehrer .....	191
Abbildung 151: Beurteilung der individuellen Komponenten .....	192
Abbildung 152: Allgemeine Einschätzung des Unterrichts .....	192
Abbildung 153: Bewertungsschlüssel der Einführungsphase (Berufliches Gymnasium) .....	192
Abbildung 154: Anzahl der fehlenden Schüler pro Befragungstermin .....	193
Abbildung 155: Cut-off-Kriterien für die Umfragen .....	194
Abbildung 156: Gewertete Testdaten .....	194
Abbildung 157: Geordnete Ergebnisse des Nachhaltigkeitstests in der Hauptstudie 2007/2008 .....	195
Abbildung 158: Ergebnisse des Nachhaltigkeitstests unter Berücksichtigung des Anforderungsniveaus .....	196
Abbildung 159: Schwere der Themen, geordnet nach der sequenziellen Abfolge der Themen .....	197
Abbildung 160: Prozedurale und objektorientierte Themen, geordnet nach der Schwere .....	198
Abbildung 161: Beispielhafte Multiple-Choice-Frage zum Thema „Arrays“ .....	199
Abbildung 162: Beispielhafte Frage und Antwort zum Thema „Arrays“ .....	199
Abbildung 163: Beispielhafte Frage und Antwort zum Thema „Assoziation“ .....	199
Abbildung 164: Beispielhafte Frage zum Thema „OOM“ .....	201
Abbildung 165: Beispielhafte Antwort zum Thema „OOM“ .....	202
Abbildung 166: Subjektives Erleben im Kontext zu zwei speziellen Frage-Bereichen .....	203

Abbildung 167: Schwere-Empfinden in Abhängigkeit vom Thema .....	204
Abbildung 168: Rangfolge der „empfundenen Schwere“ der Themen .....	204
Abbildung 169: Ergebnisse des Nachhaltigkeitstests und Werte der „empfundenen Schwere“ .....	206
Abbildung 170: Ergebnisse des Nachhaltigkeitstests dargestellt mit Hilfe der „Item-Skala“ .....	207
Abbildung 171: „Empfundene und tatsächliche Schwere“ der Themen im Kontext zur „LAJ-Untersuchung“ .	208
Abbildung 172: „Empfundene und tatsächliche Schwere“ der Themen im Kontext zur „MR-Untersuchung“ .	208
Abbildung 173: Vernetzung der einzelnen Themen mit vorher schon unterrichteten Themen .....	210
Abbildung 174: Subjektives Erleben im Kontext zu den Erlebnis-Dimensionen und zu den Frage-Bereichen ..	211
Abbildung 175: Subjektives Erleben des privaten Umfelds .....	212



# Anhang

## A) Erstes Exposé (Sommer 2006)

<b>Fachbereich Informationstechnik</b>	
Albrecht Ehlert, 15741 Bestensee, Böcklinstr. 21a, ehler@oszimt.de	Datum: 05.08.2006
<b>Exposé: Fachdidaktische Untersuchung zur OOP im Anfängerunterricht bei relativ leistungsschwachen Schülern</b>	
<b>1. Vorbemerkungen</b>	
<b>Zu meiner Person</b>	
Ich bin 48 Jahre alt, unterrichte seit mehr als 18 Jahren die Fächer Elektrotechnik, Sozialkunde und Informatik, seit ein paar Jahren ausschließlich nur noch das Fach Informatik in der gymnasialen Oberstufe bzw. das Fach Anwendungssysteme in der Berufsschule. An meiner Schule leite ich den Fachbereich Informationstechnik. Mehr unter <a href="http://www.AlbrechtEhlert.de">www.AlbrechtEhlert.de</a>	
<b>Die Schule</b>	
Das Oberstufenzentrum Informations- und Medizintechnik ist die größte IT-Schule Berlins ( <a href="http://www.oszimt.de">www.oszimt.de</a> ). Sie beherbergt unterschiedliche Bildungsgänge unter einem Dach, wie z.B. Berufsschule und gymnasiale Oberstufe.	
<b>Der Bildungsgang</b>	
Die gymnasiale Oberstufe beginnt mit der 11. Klasse (Einführungsphase). Dort haben die Schüler ein 6stündiges Fach Technik, welches sich zu gleichen Teilen aus der Elektrotechnik und der Informatik zusammensetzt.	
<b>Die Schüler</b>	
Untersucht werden sollen zwei Parallelklassen der gymnasialen Oberstufe. Die Schüler kommen überwiegend aus Realschulen und haben eine Gymnasial-Empfehlung. Die Erfahrungen der letzten Jahre zeigen jedoch, dass es sich um leistungsschwache Schüler handelt	
<b>2. Untersuchung</b>	
<b>Hintergrund</b>	
Es geht um die Art des Einstiegs in den Informatik-Unterricht und in die objektorientierte Programmierung. In den meisten Publikationen und Informatik-Lehrer-Zirkeln gilt überwiegend das Prinzip (Dogma?): „Objektbasierung bzw. Objektorientierung von Anfang an“. Die aufgetretenen Probleme werden als methodische Probleme abgetan (s. Spolwig).	
<b>Anlass</b>	
Ich (als „Marco Polo“) habe die Erfahrung gemacht, dass sich speziell leistungsschwache Schüler mit der objektorientierten Modellierung und Programmierung schwer tun.	
<b>Idee</b>	
Ist es vielleicht für dieses Schülerklientel motivierender bzw. leichter und von mehr Lernerfolg gekrönt, wenn der Informatik-Unterricht imperativ (Algorithmen-orientiert) beginnt und erst nach den „Kontrollstrukturen und Datentypen“ auf die „Begrifflichkeiten der OOP“ umsteigt?	
Ehlert - Exposé 19.07.2006	Seite 1 von 4

### Lerninhalte

Die Inhalte des einjährigen Faches (Wochenstunden: 3, Jahresstunden: 120) „Technik(Informatik)“ können in ca. 20 Module eingeteilt werden, die aus dem gültigen „Kernarbeitsplan“ für das Fach „Technik(Informatik)“ in der Einführungsphase der gymnasialen Oberstufe am OSZ IMT gewonnen werden:

Klasse: OG61 / OG62		1. Schulhalbjahr: 2006/2007	Fach: Te
Lehrer: Francioni / Ehlert		3 h / Wochen	
Modul	Thema	Programm-Beispiele, Bemerkungen	
1	Einführung in die Informatik, Klassifizierung der Informatik		
	Einordnung von Java im Kontext anderer Programmiersprachen Compiler, Interpreter, JDK und VM, Bytecode	Sch. installieren JDK und Arbeitswerkzeuge zu Hause	
	Einführung in das Labor und in die Entwicklungsumgebung Laborordnung	Hallo-Welt	
2	Einführung in die OOP Objekte als Instanzen von Klassen Klassen als Abstraktionen von Objekten Attribut und Methode OSZ-IMT-Richtlinien für Namen und Layout	<b>Mit BlueJ</b> Überblick über Objekte; Methoden anhand des Beispiels Figuren/Zeichnung Geldautomat	
3	Klassen-Modellierung Klassen- und Objektdiagramme UML – Notation: OOA, OOD Trennung von Fachkonzept und Nutzerschicht	Beispiele Buch, Kunde UML-Tools Beispielklasse und Testklasse	
4	Objekterzeugung (Instanz), Referenz, this parameterlose und parametrisierte Konstruktoren	Mehrwertsteuerberechnung für ein Buch (verminderter Satz)	
5	Attribute: Typen, Deklaration, Initialisierung, Gültigkeitsbereich Zugriffsspezifizierer (public, private), Kapselung Verwaltungsmethoden: Getter und Setter	Anhand von Beispielen aus Klassendiagrammen mit Implementierung einer Beispielklasse und Testklasse	
6	Fundam. Datentypen: byte, short, int, long, char, float, double Konstanten Datentypen und einfache Operatoren		
7	Methoden: Signatur, Parameter, Rückgabewerte, Zugriffsspezifizierer	Genauere Analyse von Konstruktor, Berechnungsmethode am Beispiel Mehrwertsteuerberechnung	
8	Der Algorithmusbegriff Kontrollstruktur Sequenz (Folge): Einfaches Programm mit Eingabe, Verarbeitung, Ausgabe (EVA- Prinzip), Quelltext-Richtlinien, Kommentare	Mehrwertsteuerberechnung	
9	Kontrollstruktur Selektion (Auswahl): Ein- und zweiseitige Auswahl, Vergleichsoperatoren Logischer Datentyp: boolean Verschachtelte Auswahl, Logische Operatoren: AND, OR, NOT Verknüpfung von Bedingungen	Programm zu Ohmschen Gesetzen (Strom-, Spannungs- und Widerstandsrechnung)	
10	Struktogramm (DIN 66261) Strukturelemente zu allen Kontrollstrukturen	Wird parallel zu den Kontrollstrukturen unterrichtet (Struktogramm-Editor)	



Klasse: OG61 / OG62

2. Schulhalbjahr: 2006/2007

Fach: Te

Lehrer: Francioni / Ehlert

3 h / Wochen

Modul	Thema	Programm-Beispiele, Bemerkungen
11	Kontrollstruktur Selektion (Auswahl): Fallunterscheidung	Programm zur Währungsumrechnung mit Wechselgebühren (EUR in USD) ODER Umbau des Programms zu Ohmschen Gesetzen
12	Kontrollstruktur Iteration: Vorprüfende bzw. kopfgesteuerte Schleife, nachprüfende bzw. fußgesteuerte Schleife, Zählschleife Struktogramm, Schreibtischtest	Addition der Zahlen von 1 bis 100 mit allen 3 Schleifen anhand des gaußschen Algorithmus unter Verwendung eines Arrayattributs
13	Arrays (nicht Klasse Array) Deklaration, Initialisierung, Wertzuweisung Attribut length; Methoden equals, clone, sort	Z. B.: Höhe, Breite, Tiefe in cm eines Artikel-Objektes
14	Datentyp/Klasse String Methoden: length, equals, charAt, indexOf, toUpperCase, toLowerCase	Caesar-Ver- und Entschlüsselung
15	Vererbung in UML Vererbung in Java (super)	Beispiel Person, Schüler, Lehrer Implementierung der Personenliste mit Vererbung
16	Feld von Objektreferenzen (oder Sammlungsdatentyp ArrayList Methoden add, delete (subList, size; ListIterator bei ArrayList)	Beispiel Schülerliste
17	Dokumentationen: Klassen- und Objektdiagramm JavaDoc Quelltexte	UML-, Javadoc- und OSZ IMT- Richtlinien
18	Vererbung: Überladen und Überschreiben von Methoden Abstrakte Klasse	Beispiel Medien-Sammlung mit CD, mp3, Video z. B. Medien
19	Klassenattribute, Klassenoperationen	anhand Medien-Sammlung
20	Implementierung einfacher Assoziationen (Array-Attribut) (allg. ohne Differenzierung)	Beispiel Kunde, Bank, Konto (BlueJ?)
	Wiederholung der Themen für einen Vergleichstest	

#### Zeitlicher Rahmen / Organisatorisches bzw. Statistisches

Für die Untersuchung werden zwei Parallelklassen mit allen Modulen beschult, bei der einen Klasse allerdings ohne objektorientierten Einstieg, also alle grau unterlegten Module zuerst.

Die Klassen sollen in der Altersdurchmischung, Geschlechtsthroughmischung und Mathedurchschnittszensur nahezu identisch sein. Ein Problem für die Untersuchung ist, dass zwei verschiedene Lehrer in den Klassen eingesetzt werden. Mein Koll. Francioni ist aber von der Erfahrung und vom Lehrer-Typ mir sehr ähnlich. Der Mathematikunterricht wird vom selben Lehrer gehalten. Die Mathedurchschnittsnote am Ende des Untersuchungsjahres ist somit ein Indikator für die „Mathe-/Informatik-Begabung“.

#### Dokumente

- (Fast) jedes Modul wird mit einer Evaluation abgeschlossen, die die Motivation der Schüler, die Zufriedenheit mit dem Lehrer und die Schwere des Stoffes abfragt. Diese Evaluation findet mit Hilfe einer Online-Befragung statt.
- Am Ende des Schuljahres wird ein großer Vergleichstest geschrieben, z.B. über die Themengebiete: Kontrollstrukturen, Struktogramme, UML und OOP.
- Einmal im Monat gibt es ein Gespräch mit den Klassen (inkl. Gesprächsprotokoll) mit Fragestellungen wie z.B.: „Wie gefällt euch der Stoff?“, „Was findet ihr gut, was nicht so gut?“, „Was könnte besser gemacht werden?“ etc.

## Mögliche Ergebnisansichten

### a) Mögliche Datenbasis (Simulation)

	OG61 SG- Module OOPvAa	OG61 Mot- OOPvAv	OG62 SG- ImpEin	OG62 Mot- ImpEin
1	2	2	2	2
2	3	3	3	4
3	2	2	2	3
6	3	4	3	4
9	4	5	2	3
10	3	4	4	2
12	2	3	4	3
13	3	3	3	2
14	4	2	2	3
15	3	3	2	2
18	2	4	3	3
19	3	3	2	2
Durchschnitt	2,91	3,27	2,73	2,82

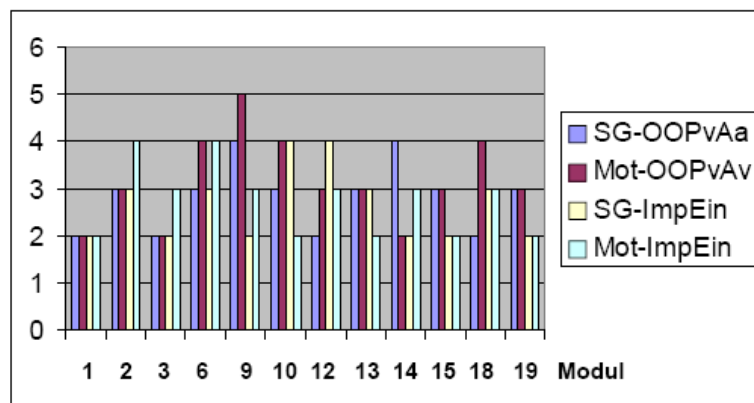
### b) Allgemeine mögliche Ergebnisansicht

Klasse	Mathe- Anfang	Mathe- Endnote	Motivations- durchschnitt	Schwere- durchschnitt	Kontroll- strukturen	Strukto- gramme	UML	OOP	Summe
OG61	3,1	3,4	3,27	2,91	2,8	2,7	3,1	3,2	2,95
OG62	3,2	3,3	2,82	2,73	2,9	2,5	3,2	3,4	3,00

Mögliches Kurzfazit aus den o. g. Zahlen:


Der Lernerfolg war in beiden Klassen fast der gleiche, die OG62 war aber durchschnittlich motivierter und hat den Unterricht nicht ganz so schwer empfunden. Die fast gleiche Mathe-Durchschnittsnote rundet das Bild ab.

### c) Speziell mögliche Ergebnisansicht



Hier muss für jedes einzelne Modul eine Aussage getroffen werden!

## B) Vorhandener Kernarbeitsplan vor der Studie (2006)

Kernarbeitsplan Technik / Informatik Einführungsphase, 1. Halbjahr		
Klasse:	Schulhalbjahr:	Fach: <b>Tech</b>
Lehrer(in):	3 h / Wochen	
Woche / Datum	Thema	Programm-Beispiele, Bemerkungen
	Einführung in die Informatik, Klassifizierung der Informatik	
	Einführung in das Labor und in die Entwicklungsumgebung Laborordnung	Hallo-Welt
	Einordnung von Java im Kontext anderer Programmiersprachen Compiler, Interpreter, JDK und VM, Bytecode	Sch. installieren JDK und Arbeitswerkzeuge zu Hause
	Einführung in die OOP Objekte als Instanzen von Klassen Klassen als Abstraktionen von Objekten Attribut und Methode OSZ-IMT-Richtlinien für Namen und Layout	Überblick über Objekte; Methoden anhand des Beispiels Figuren/Zeichnung (BlueJ)
	Klassen- und Objektdiagramme UML – Notation: OOA, OOD Klassen-Modellierung	Beispiele Buch, Kunde
	Attribute: Typen, Deklaration, Initialisierung, Gültigkeitsbereich Fundam. Datentypen: byte, short, int, long, char, float, double Konstanten Datentypen und einfache Operatoren Zugriffsspezifizierer (public, private), Kapselung Verwaltungsmethoden: Getter und Setter	Anhand von Beispielen aus Klassendiagrammen mit Implementierung einer Beispielfunktion und Testklasse
	Trennung von Fachkonzept und Nutzerschicht (ggf. weglassen: 3-Schichten-Architektur (three tier))	Beispielfunktion und Testklasse
	Objekterzeugung (Speicher), Instanz, Referenz, Konstruktoren	Mehrwertsteuerberechnung für ein Buch (vermindertes Satz)
	Kontrollstruktur Sequenz (Folge): Einfaches Programm mit Eingabe, Verarbeitung, Ausgabe (EVA-Prinzip), Quelltext-Richtlinien, Kommentare	Mehrwertsteuerberechnung
	Methoden: Signatur, Parameter, Rückgabewerte, Zugriffsspezifizierer	Genaue Analyse von Konstruktor, Berechnungsmethode am Beispiel Mehrwertsteuerberechnung
	Kontrollstruktur Selektion (Auswahl): Ein- und zweiseitige Auswahl, Vergleichsoperatoren Logischer Datentyp: boolean Struktogramm (DIN 66261)	Programm zu Ohmschen Gesetzen (Strom-, Spannungs- und Widerstandsrechnung) Struktogramm-Editor
	Der Algorithmusbegriff Fallunterscheidung Struktogramm	Programm zur Währungsumrechnung mit Wechselgebühren (EUR in USD, CHF, usw.) ODER Umbau des Programms zu Ohmschen Gesetzen
	Online-Kurs: Beginning to Program With the JPL	WJ-1102
	Verschachtelte Auswahl, Logische Operatoren: AND, OR, NOT Verknüpfung von Bedingungen	Selbst gewähltes Beispiel
	<i>1. Klassenarbeit</i>	
	<i>2. Klassenarbeit</i>	

**Kernarbeitsplan Technik / Informatik  
Einführungsphase, 2. Halbjahr**



Klasse: Schulhalbjahr: Fach: **Tech**  
 Lehrer(in): 3 h / Wochen

Woche / Datum	Thema	Programm-Beispiele, Bemerkungen
	Wiederholung: Klassen- und Objektdiagramme UML – Notation Klassen-Modellierung	Beispiel: Artikel Einführung von GO oder anderem UML-Tool
	Arrays (nicht Klasse Array) Deklaration, Initialisierung, Wertzuweisung Attribut length; Methoden equals, clone, sort	Z. B.: Höhe, Breite, Tiefe in cm eines Artikel-Objektes
	Kontrollstruktur Iteration Vorprüfende bzw. kopfgesteuerte Schleife, nachprüfende bzw. fußgesteuerte Schleife, Zählschleife Struktogramm, Schreibtischtest	Addition der Zahlen von 1 bis 100 mit allen 3 Schleifen anhand des gaußschen Algorithmus unter Verwendung eines Arrayattributs (Übung: iterative Aufgaben aus JavaKara)
	Datentyp/Klasse String Methoden: length, equals, charAt, indexOf, toUpperCase, toLowerCase JavaDoc lesen	Caesar-Ver- und Entschlüsselung
	Online-Kurs: Using Decision and Looping Constructs	WJ-1103
	Vererbung in UML Parametrisierter Konstruktor	Beispiel Person, Schüler, Lehrer
	Feld von Objektreferenzen (oder Sammlungsdatentyp ArrayList Methoden add, delete (subList, size; ListIterator bei ArrayList)	Beispiel Schülerliste
	Vererbung in Java parametrisierte Konstruktoren, super JavaDoc anfertigen, Quelltexte dokumentieren	Implementierung und Dokumentation der Personenliste mit Vererbung Anwendung eines Feldes von Objektreferenzen (oder ArrayList)
	Vererbung in UML und Java Überladen und Überschreiben von Methoden Abstrakte Klasse	Beispiel Medien-Sammlung mit CD, mp3, Video z. B. Medien
	Klassenattribute, Klassenoperationen	anhand Medien-Sammlung
	Implementierung einfacher Assoziationen (Array-Attribut) (allg. ohne Differenzierung)	Beispiel Kunde, Bank, Konto (BlueJ)
	Wiederholung Themen für Vergleichstest	
	<i>1. Klassenarbeit</i>	
	<i>2. Klassenarbeit</i>	
	<i>Vergleichstest OG online</i>	
ggf. falls noch Zeit	einfach verkettete Listen Wiederholung Referenz Methoden suchen, einfügen, entfernen (arbeitsteilig)	Kundenliste, Kontenliste
	Java-GUIs am Beispiel Applet Einfache GUI-Elemente, z. B. Textfeld	Einfaches Applet z. B. Eingabe von Name und Passwort

## C) Vergleichstests vor der Studie (2004)

### 1. Vergleichstest für das Thema Datentypen

<p><b>OSZ</b> Informations- und Medizintechnik</p> <p>Klasse: .....</p> <p>Name: .....</p>	<p><b>Vergleichstest in TmL (Informatik)</b></p>	<p>Berlin, .....</p> <p>Fach: .....</p> <p>Seite: 1</p>												
<p><u>Hinweise zur Bewertung:</u> Bei den folgenden Fragen kann jeweils eine Antwort oder mehrere Antworten richtig sein. Falsche oder fehlende Antworten führen innerhalb einer Aufgabe zu Punktabzügen.</p>														
<p><b>1. Aufgabe</b></p> <p>Durch den Datentyp eines Attributs wird festgelegt, ...</p> <p>(A) ... welche Werte das Attribut annehmen kann.          (B) ... welche Operationen damit ausgeführt werden können.          (C) ... wie groß der zur Verfügung gestellte Speicherplatz ist.          (D) ... welchen Wert das Attribut hat.          (E) ... welchen Gültigkeitsbereich (scope) das Attribut hat.</p> <p style="text-align: right;">1</p>														
<p><b>2. Aufgabe</b></p> <p>Welche Aussage(n) beschreibt/beschreiben die folgende Deklaration <u>vollständig</u>?</p> <pre>public static final int MAX = 37;</pre> <p>(A) Eine öffentliche Klassenkonstante mit dem Wert 37.0 wird vereinbart.          (B) Eine auch für andere Klassen zugängliche, ganzzahlige Konstante wird vereinbart.          (C) Eine öffentliche Klassenvariable vom Typ int mit dem Initialwert 37 wird vereinbart.          (D) Eine für andere Klassen zugängliche Klassenkonstante vom Typ int mit dem unveränderbaren Wert 37 wird vereinbart.</p> <p style="text-align: right;">1</p>														
<p><b>3. Aufgabe</b></p> <p>Die Tabelle zeigt den Inhalt der int-Variablen x, y und der boolean-Variablen vergleich vor und nach der Ausführung einer Programmzeile.</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th></th> <th>x</th> <th>y</th> <th>vergleich</th> </tr> </thead> <tbody> <tr> <td>Inhalt vor der Ausführung</td> <td>7</td> <td>4</td> <td>true</td> </tr> <tr> <td>Inhalt nach der Ausführung</td> <td>7</td> <td>4</td> <td>false</td> </tr> </tbody> </table> <p>Geben Sie die zutreffende(n) Zeile(n) an!</p> <p>(A) vergleich &lt; (x+y);          (B) vergleich = (x&lt;=y);          (C) x&lt;y = vergleich;          (D) vergleich = !(x==y);</p> <p style="text-align: right;">1</p>				x	y	vergleich	Inhalt vor der Ausführung	7	4	true	Inhalt nach der Ausführung	7	4	false
	x	y	vergleich											
Inhalt vor der Ausführung	7	4	true											
Inhalt nach der Ausführung	7	4	false											
<p><b>4. Aufgabe</b></p> <pre>public class Test1{     public String getAdresse(String name) {         ..     } }</pre> <p>Um welche der folgenden Methoden könnte die Klasse Test1 jeweils einzeln ergänzt werden?</p> <p>(A) private String getAdresse(String vorname)          (B) public String getAdresse(String name, int personenNummer)          (C) public String getAdresse(String firma)          (D) public String getAdresse(String name, String firma)</p> <p style="text-align: right;">1</p>														
<p><b>5. Aufgabe</b></p> <pre>public int quadriere(int a) {     a = a * a;     return a; } int x=12, y =9; y = quadriere(x); System.out.println("x = "+x+" / y = "+y);</pre> <p>Welche Ausgabe erzeugt der dargestellte Programmausschnitt?</p> <p>(A) x = 12 / y = 144          (B) x = 144 / y = 144          (C) x = 144 / y = 9          (D) x = 12 / y = 9</p> <p style="text-align: right;">1</p>														

<p style="text-align: center;"><b>OSZ</b> Informations- und Medizintechnik</p> <p>Klasse: .....</p> <p>Name: .....</p>	<p><b>Vergleichstest in TmL (Informatik)</b></p>	<p>Berlin, .....</p> <p>Fach: .....</p> <p>Seite: 2</p>
<p><b>6. Aufgabe</b></p> <pre>int x[] = new int[25];</pre> <p>Welche Aussage(n) trifft/treffen nach der Ausführung der Programmzeile zu?</p> <p>(A) x[24] hat den Wert 0.  (B) Der Wert von x[24] ist nicht definiert.  (C) x[25] hat den Wert 0.  (D) x[0] hat den Wert null.  (E) x.length hat den Wert 25.</p> <p style="text-align: right;">1</p> <p><b>7. Aufgabe</b> <pre>class Abzieher{     public void abziehen(double decrementMe){         decrementMe = decrementMe -1     } } //Abzieher class Test2{     public static void main(String args){         double d = 12.3;         Abzieher abzieh = new Abzieher();         abzieh.abziehen(d);         System.out.println(d);     } //main } //Test2</pre> <p>Welche Ausgabe erzeugt das Programm Test2?</p> <p>(A) 0.0  (B) -1.0  (C) 12.3  (D) 11.3  (E) 12.0</p> <p style="text-align: right;">1</p> <p><b>8. Aufgabe</b></p> <p>Welche Werte kann eine Variable vom Typ byte annehmen?</p> <p>(A) 0 bis 255  (B) 0 bis 65535  (C) -128 bis 127  (D) Der Wert hängt von der verwendeten Hardware ab.</p> <p style="text-align: right;">1</p> <p><b>9. Aufgabe</b> <pre>int x = 3, int y = 10; System.out.println(y % x);</pre> <p>Welche Ausgabe erzeugt der dargestellte Programmausschnitt?</p> <p>(A) 0  (B) 1  (C) 2  (D) 3</p> <p style="text-align: right;">1</p> <p><b>10. Aufgabe</b></p> <p>Welche der folgenden Ausdrücke sind zulässig?</p> <p>(A) String st = "Hallo"; int y = 9; st = st + y;  (B) String st = "Hallo"; int y = 9; if(st == y){System.out.println(st);}  (C) String st = "Hallo"; int y = 9; y = y + st;  (D) String st = "Hallo"; int y = 9; st = y + st;</p> <p style="text-align: right;">1</p> </p></p>		

<p style="text-align: center;"><b>OSZ</b>  <small>Informations- und Medizintechnik</small></p> <p>Klasse: .....</p> <p>Name: .....</p>	<p><b>Vergleichstest in TmL (Informatik)</b></p>	<p>Berlin, .....</p> <p>Fach: .....</p> <p>Seite: 3</p>																														
<p><b>Lösungen</b></p> <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 10%;">1</td><td style="width: 80%;">A, B, C</td><td style="width: 10%;"></td></tr> <tr><td>2</td><td>D</td><td></td></tr> <tr><td>3</td><td>B</td><td></td></tr> <tr><td>4</td><td>B, D</td><td></td></tr> <tr><td>5</td><td>A</td><td></td></tr> <tr><td>6</td><td>A, E</td><td></td></tr> <tr><td>7</td><td>C</td><td></td></tr> <tr><td>8</td><td>C</td><td></td></tr> <tr><td>9</td><td>B</td><td></td></tr> <tr><td>10</td><td>A, D</td><td></td></tr> </table>			1	A, B, C		2	D		3	B		4	B, D		5	A		6	A, E		7	C		8	C		9	B		10	A, D	
1	A, B, C																															
2	D																															
3	B																															
4	B, D																															
5	A																															
6	A, E																															
7	C																															
8	C																															
9	B																															
10	A, D																															

2. Vergleichstest für das Thema Steuerstrukturen

<b>OSZ IMT</b>	<b>Vergleichstest</b> Steuerstrukturen	Berlin, d. ....
Klasse: .....		Fach: TmL
Name: .....		Seite 1 von 1
Notieren Sie die Antwort bzw. kreuzen Sie die richtige Antwort an.		
Nr	Aufgabe	Antwort
1	Notieren Sie eine beliebige Zuweisung zur Variablen <code>String s;</code>	
2	<pre>int i; for (i = 1; i &lt; 10; i++) { /* Anweisungen, die i nicht verändern */ } System.out.println(i);</pre> Welche Zahl wird ausgegeben?	a) 1 <input type="checkbox"/> b) 9 <input type="checkbox"/> c) 10 <input type="checkbox"/> d) 11 <input type="checkbox"/>
3	Welche Schleife bildet die Zählschleife aus Aufgabe 2 nach? a) <code>int i = 1; while (i &lt; 10) { /* Anweisungen */ }</code> b) <code>int i = 1; while (i &lt; 10) { /* Anweisungen */; i++; }</code> c) <code>int i = 1; while (i &lt; 10) { i++; /* Anweisungen */ }</code> d) <code>int i = 1; while (i &lt; 10) { i++; } /* Anweisungen */</code>	a) <input type="checkbox"/> b) <input type="checkbox"/> c) <input type="checkbox"/> d) <input type="checkbox"/>
4	<pre>if ( i != 10 ) {     if ( i &gt; 8 ) {         if ( i &lt; 12 ) {             System.out.println("Ende"); } } } </pre> Bei welchen Zahlen i wird Ende ausgegeben?	a) 8,9,10,11,12 <input type="checkbox"/> b) 8,9,11,12 <input type="checkbox"/> c) 9,10,11 <input type="checkbox"/> d) 9,11 <input type="checkbox"/>
5	Eine Schleife soll so oft durchlaufen werden, wie der Benutzer es wünscht, mindestens jedoch ein mal. Welche Art Schleife verwenden Sie?	a) Zählschleife <input type="checkbox"/> b) Kopfgesteuerte S. <input type="checkbox"/> c) Fußgesteuerte S. <input type="checkbox"/>
6	Eine Zählvariable i soll die ersten acht 2er-Potenzen (also ?, ?, 4, 8, 16, 32, 64, 128) durchlaufen. Ergänzen Sie: <pre>for (int i = _____; i &lt;= 128; _____)</pre>	
8	<pre>int i = 10; int j = 5; while ( i &lt; 20 ) {     j = j + 1;     if ( i &gt;= 18 ) { j = j + 2; }     i = i + 2; } System.out.println(j);</pre> Welche Zahl wird ausgegeben?	
9	<pre>if ( i &gt; 0 ) {     if ( i &lt; 10 ) { System.out.println(1); }     if ( i &gt; 20 ) { System.out.println(2); } else {         if ( i &lt; 30 ) { System.out.println(3); }         else { System.out.println(4); }     } }</pre> Nennen Sie einen Wert für i, damit nur 3 ausgegeben wird!	
10	Welche Verknüpfung gibt den folgenden Programmausschnitt korrekt wieder? <pre>if ( i &gt; 0 ) {     if ( i &lt; 10 ) {         if ( i &gt; 5 ) { /* A */ }         else { /* B */ }     } }</pre>	a) <code>if ( i &gt; 0 &amp;&amp; i &lt; 10 &amp;&amp; i &gt; 5 )</code> <code>{ /* A */ } else { /* B */ }</code> <input type="checkbox"/> b) <code>if ( i &gt; 0    i &lt; 10    i &gt; 5 )</code> <code>{ /* A */ } else { /* B */ }</code> <input type="checkbox"/> c) <code>if ( i &gt; 0 &amp;&amp; i &lt; 10 ) { if ( i &gt; 5 )</code> <code>{ /* A */ } else { /* B */ } }</code> <input type="checkbox"/> d) <code>if ( i &gt; 0    i &lt; 10 ) { if ( i &gt; 5 )</code> <code>{ /* A */ } else { /* B */ } }</code> <input type="checkbox"/>
<b>Punkte:</b> / 10		



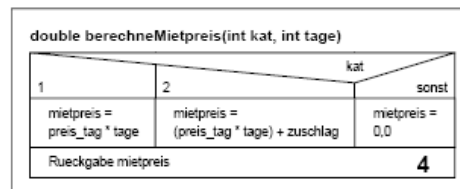
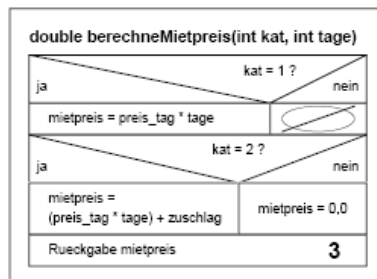
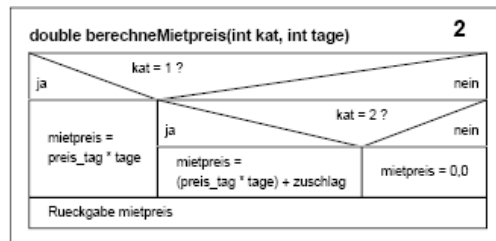
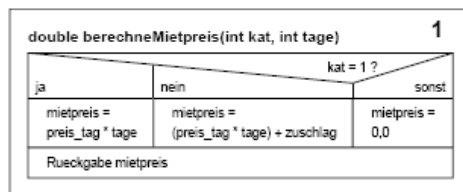
<b>OSZ IMT</b>	<b>Vergleichstest</b> Steuerstrukturen	Berlin, d. ....
Klasse: .....		Fach: TmL
Name: .....		Seite 2 von 1
Lösungen		
1	s = „Test“;	
2	c	
3	b	
4	d	
5	c	
6	1	
7	i = i * 2	
8	12	
9	beliebige Zahl zwischen 10 und 20	
10	c	

### 3. Vergleichstest für das Thema Struktogramme

<b>OSZ IMT</b> Technik mit Laborübungen Berlin,	<b>Vergleichstest                  Informatik OG                  Struktogramme</b>	Klasse: Name: E-Phase															
<p>1. Wählen Sie durch Ankreuzen aus, welche der folgenden Aussagen auf Struktogramme zutreffen.</p> <p>[ 1 ] Da es für Struktogramme keine genaueren Vorgaben gibt, muss man sich in die Logik eines Struktogramms immer erst eindenken.</p> <p>[ 2 ] Mit Struktogrammen werden Klassen innerhalb der OOP definiert.</p> <p>[ 3 ] Struktogramme werden ausschließlich zur Dokumentation der Anwendungsschicht benutzt.</p> <p>[ 4 ] In einer DIN Norm sind Symbole für Steuerstrukturen festgelegt, die in Struktogrammen benutzt werden, um Algorithmen darzustellen.</p> <p>[ 5 ] Struktogramme sind überholt, heute werden sie nicht mehr benutzt.</p> <p>[ 6 ] Mit Hilfe von Struktogrammen können Methoden einer Klasse sowie die Anwenderschicht (im Sinne der DOS-Konsole) von objektorientierten Programmen geplant und dokumentiert werden.</p> <p>2. Ordnen Sie zu, indem Sie 4 von insgesamt 7 Ziffern des Struktogramms in die Kästchen bei den Steuerstruktur-Namen eintragen! (4 * 1 Pkt)</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: left; padding: 5px;">Struktogramm</th> <th style="width: 50%; text-align: left; padding: 5px;">Struktogramm-Symbole</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"> </td> <td style="padding: 5px;">                 Fallunterscheidung [   ]                  Fußgesteuerte Schleife [   ]                  Verzweigung, Abfrage [   ]                  Kopfgesteuerte Schleife [   ]             </td> </tr> <tr> <td style="padding: 5px;"> </td> <td style="padding: 5px;">                 / 4             </td> </tr> </tbody> </table> <p>3. Welchen Wert hat Variable <b>z</b> bei der Ausgabe am Ende des little algorithm?</p> <div style="border: 1px solid black; padding: 10px; width: fit-content;"> <p><b>little algorithm</b></p> <p>x = 4, y = 3, z = 7</p> <p>Solange x &lt; 7</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border: 1px solid black; padding: 2px;">ja</td> <td style="width: 10%; text-align: center; border: none;">x modulo y = 0 ?</td> <td style="width: 40%; border: 1px solid black; padding: 2px;">nein</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">z = (z + y) * x</td> <td style="border: none;"></td> <td style="border: 1px solid black; padding: 2px;"></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">x = x + 1</td> <td style="border: none;"></td> <td style="border: 1px solid black; padding: 2px;"></td> </tr> </table> <p style="border: 1px solid black; padding: 2px;">z = z * y</p> <p style="border: 1px solid black; padding: 2px;">Ausgabe: z</p> </div> <div style="border: 1px solid black; padding: 10px; width: fit-content; margin-top: 10px;"> <p>modulo = Restwert der ganzzahligen Division mit Rest</p> <p>Bsp. 41 / 8 = 5, Rest 1                  16 / 4 = 4, Rest 0</p> </div> <p style="text-align: right; margin-top: 10px;"><b>z hat den Wert [   ]</b></p>			Struktogramm	Struktogramm-Symbole		Fallunterscheidung [   ] Fußgesteuerte Schleife [   ] Verzweigung, Abfrage [   ] Kopfgesteuerte Schleife [   ]		/ 4	ja	x modulo y = 0 ?	nein	z = (z + y) * x			x = x + 1		
Struktogramm	Struktogramm-Symbole																
	Fallunterscheidung [   ] Fußgesteuerte Schleife [   ] Verzweigung, Abfrage [   ] Kopfgesteuerte Schleife [   ]																
	/ 4																
ja	x modulo y = 0 ?	nein															
z = (z + y) * x																	
x = x + 1																	

4. Welches Struktogramm bildet die in Java-Code dargestellte Methode berechneMietpreis(int kat, int tage) korrekt ab? Tragen Sie die Nummer unten ein.

```
public double berechneMietpreis(int kat, int tage) {
    if (kat==1)
        mietpreis = preis_tag * tage;
    else if (kat==2)
        mietpreis = (preis_tag * tage) + zuschlag;
    else
        mietpreis = 0.0;
    return mietpreis;
}
```



Richtige Lösung in Struktogramm: [   ]

/2

Punkte:  10

### Lösungen


1	4 und 6
2	2 / 7 / 5 / 4
3	180
4	2

4. Vergleichstest für das Thema UML

<p><b>OSZ IMT</b></p> <p>Klasse: .....</p> <p>Name: .....</p>	<p><b><u>Vergleichstest: Thema UML</u></b></p> <p><b>Bitte beachten:</b> Es können mehrere Antworten richtig sein!</p>	<p><b>Informatik</b></p> <p>Berlin, den .....</p> <p>Blatt-Nr.: .....</p>							
<p>1. Wie heißen die Verwaltungsoperationen („getter“ bzw. „setter“) der Klasse „Wuerfel“ für das Attribut „augenzahl“?</p> <p><input type="radio"/> setWuerfel</p> <p><input type="radio"/> getWuerfel</p> <p><input type="radio"/> setAugenzahl</p> <p><input type="radio"/> getAugenzahl</p> <p><input type="radio"/> setZahl</p> <p><input type="radio"/> getZahl</p> <p style="text-align: right;"><b>1P</b></p>	<p>2. Attribute werden im</p> <p><input type="radio"/> Klassendiagramm mindestens mit dem Datentyp</p> <p><input type="radio"/> Klassendiagramm nur mit einem Wert</p> <p><input type="radio"/> Objektdiagramm mindestens mit dem Datentyp</p> <p><input type="radio"/> Objektdiagramm nur mit einem Wert</p> <p>angegeben!</p> <p style="text-align: right;"><b>1P</b></p>								
<p>3. Methoden werden im Objektdiagramm</p> <p><input type="radio"/> angegeben</p> <p><input type="radio"/> nicht angegeben</p> <p><input type="radio"/> je nach Bedarf angegeben</p> <p style="text-align: right;"><b>1P</b></p>	<p>4. Wie heißt der Konstruktor der Klasse „Weltraumschiff“?</p> <p>_____</p> <p style="text-align: right;"><b>1P</b></p>								
<p>5. Die Methodenbeschreibung im UML-Diagramm lautet</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> <p>- setZaehlerstand(void): int</p> </div> <p>Diese Darstellung enthält</p> <p><input type="radio"/> keinen (Hinweis: Der Name setZaehlerstand ist richtig!)</p> <p><input type="radio"/> einen</p> <p><input type="radio"/> zwei</p> <p><input type="radio"/> drei</p> <p><input type="radio"/> vier</p> <p>Modellierungs-Fehler!</p> <p style="text-align: right;"><b>1P</b></p>	<p>6. <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> <p><b><u>einKunde</u></b></p> </div></p> <p>Dies ist die Darstellung eines / einer</p> <p><input type="radio"/> Klasse</p> <p><input type="radio"/> Objekts</p> <p><input type="radio"/> Attributs</p> <p><input type="radio"/> Methode</p> <p style="text-align: right;"><b>1P</b></p>								
<p>7. <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> <p><b>Baum</b></p> </div></p> <p>Dies ist die Darstellung eines / einer</p> <p><input type="radio"/> Klasse</p> <p><input type="radio"/> Objekts</p> <p><input type="radio"/> Attributs</p> <p><input type="radio"/> Methode</p> <p style="text-align: right;"><b>1P</b></p>	<p>8. Welche Methoden bzw. Konstruktoren liefern in jedem Fall einen Rückgabewert?</p> <p><input type="radio"/> unparametrisierter Konstruktor</p> <p><input type="radio"/> parametrisierter Konstruktor</p> <p><input type="radio"/> „getter“ (get-Verwaltungsmethode)</p> <p><input type="radio"/> „setter“ (set-Verwaltungsmethode)</p> <p style="text-align: right;"><b>1P</b></p>								
<p>9. Welche Methoden bzw. Konstruktoren haben / brauchen einen Parameter?</p> <p><input type="radio"/> unparametrisierter Konstruktor</p> <p><input type="radio"/> parametrisierter Konstruktor</p> <p><input type="radio"/> „getter“ (get-Verwaltungsmethode)</p> <p><input type="radio"/> „setter“ (set-Verwaltungsmethode)</p> <p style="text-align: right;"><b>1P</b></p>	<p>10. Welches Vergabe der Zugriffsspezifizierer findet man am häufigsten?</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border: 1px solid black; padding: 5px;"><input type="radio"/> + farbe: String</td> <td style="width: 50%; border: 1px solid black; padding: 5px;"><input type="radio"/> + farbe: String</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;"><input type="radio"/> + male(): void</td> <td style="border: 1px solid black; padding: 5px;"><input type="radio"/> - male(): void</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;"><input type="radio"/> - farbe: String</td> <td style="border: 1px solid black; padding: 5px;"><input type="radio"/> - farbe: String</td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;"><input type="radio"/> + male(): void</td> <td style="border: 1px solid black; padding: 5px;"><input type="radio"/> - male(): void</td> </tr> </table> <p>(„oberes Rechteck“ fehlt jeweils!)</p> <p style="text-align: right;"><b>1P</b></p>	<input type="radio"/> + farbe: String	<input type="radio"/> + farbe: String	<input type="radio"/> + male(): void	<input type="radio"/> - male(): void	<input type="radio"/> - farbe: String	<input type="radio"/> - farbe: String	<input type="radio"/> + male(): void	<input type="radio"/> - male(): void
<input type="radio"/> + farbe: String	<input type="radio"/> + farbe: String								
<input type="radio"/> + male(): void	<input type="radio"/> - male(): void								
<input type="radio"/> - farbe: String	<input type="radio"/> - farbe: String								
<input type="radio"/> + male(): void	<input type="radio"/> - male(): void								

OSZ IMT	Vergleichstest: Thema UML	Informatik			
Klasse: ..... Name: .....	<b>Bitte beachten:</b> Es können mehrere Antworten richtig sein!	Berlin, den ..... Blatt-Nr.: .....			
<p>1. Wie heißen die Verwaltungsoperationen („getter“ bzw. „setter“) der Klasse „Wuerfel“ für das Attribut „augenzahl“?</p> <p> <input type="radio"/> setWuerfel  <input type="radio"/> getWuerfel  <input checked="" type="radio"/> setAugenzahl  <input checked="" type="radio"/> getAugenzahl  <input type="radio"/> setZahl  <input type="radio"/> getZahl           </p> <p style="text-align: right;"><b>1P</b></p>	<p>2. Attribute werden im</p> <p> <input checked="" type="radio"/> Klassendiagramm mindestens mit dem Datentyp  <input type="radio"/> Klassendiagramm nur mit einem Wert  <input type="radio"/> Objektdiagramm mindestens mit dem Datentyp  <input checked="" type="radio"/> Objektdiagramm nur mit einem Wert           </p> <p>angegeben!</p> <p style="text-align: right;"><b>1P</b></p>				
<p>3. Methoden werden im Objektdiagramm</p> <p> <input type="radio"/> angegeben  <input checked="" type="radio"/> nicht angegeben  <input type="radio"/> je nach Bedarf angegeben           </p> <p style="text-align: right;"><b>1P</b></p>	<p>4. Wie heißt der Konstruktor der Klasse „Weltraumschiff“?</p> <p style="text-align: center;">Weltraumschiff()</p> <p style="text-align: right;"><b>1P</b></p>				
<p>5. Die Methodenbeschreibung im UML-Diagramm lautet</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">       - setZaehlerstand(void): int     </div> <p>Diese Darstellung enthält</p> <p> <input type="radio"/> keinen  <input type="radio"/> einen  <input type="radio"/> zwei  <input checked="" type="radio"/> drei  <input type="radio"/> vier           </p> <p style="text-align: right;">(Hinweis: Der Name setZaehlerstand ist richtig!) <b>1P</b></p>	<p>6.</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <u>einKunde</u> </div> <p>Dies ist die Darstellung eines / einer</p> <p> <input type="radio"/> Klasse  <input checked="" type="radio"/> Objekts  <input type="radio"/> Attributs  <input type="radio"/> Methode           </p> <p style="text-align: right;"><b>1P</b></p>				
<p>7.</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">       Baum     </div> <p>Dies ist die Darstellung eines / einer</p> <p> <input checked="" type="radio"/> Klasse  <input type="radio"/> Objekts  <input type="radio"/> Attributs  <input type="radio"/> Methode           </p> <p style="text-align: right;"><b>1P</b></p>	<p>8. Welche Methoden bzw. Konstruktoren liefern in jedem Fall einen Rückgabewert?</p> <p> <input type="radio"/> unparametrisierter Konstruktor  <input type="radio"/> parametrisierter Konstruktor  <input checked="" type="radio"/> „getter“ (get-Verwaltungsmethode)  <input type="radio"/> „setter“ (set-Verwaltungsmethode)           </p> <p style="text-align: right;"><b>1P</b></p>				
<p>9. Welche Methoden bzw. Konstruktoren haben / brauchen einen Parameter?</p> <p> <input type="radio"/> unparametrisierter Konstruktor  <input checked="" type="radio"/> parametrisierter Konstruktor  <input type="radio"/> „getter“ (get-Verwaltungsmethode)  <input checked="" type="radio"/> „setter“ (set-Verwaltungsmethode)           </p> <p style="text-align: right;"><b>1P</b></p>	<p>10. Welches Vergabe der Zugriffsspezifizierer findet man am häufigsten?</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border: 1px solid black; padding: 5px;"> <input type="radio"/> + farbe: String  <input type="radio"/> + male(): void         </td> <td style="width: 50%; border: 1px solid black; padding: 5px;"> <input type="radio"/> + farbe: String  <input type="radio"/> - male(): void         </td> </tr> <tr> <td style="border: 1px solid black; padding: 5px;"> <input checked="" type="radio"/> - farbe: String  <input type="radio"/> + male(): void         </td> <td style="border: 1px solid black; padding: 5px;"> <input type="radio"/> - farbe: String  <input type="radio"/> - male(): void         </td> </tr> </table> <p>(„oberes Rechteck“ fehlt jeweils!) <b>1P</b></p>	<input type="radio"/> + farbe: String <input type="radio"/> + male(): void	<input type="radio"/> + farbe: String <input type="radio"/> - male(): void	<input checked="" type="radio"/> - farbe: String <input type="radio"/> + male(): void	<input type="radio"/> - farbe: String <input type="radio"/> - male(): void
<input type="radio"/> + farbe: String <input type="radio"/> + male(): void	<input type="radio"/> + farbe: String <input type="radio"/> - male(): void				
<input checked="" type="radio"/> - farbe: String <input type="radio"/> + male(): void	<input type="radio"/> - farbe: String <input type="radio"/> - male(): void				

D) Fragebogen zum subjektiven Erleben der Schüler (Vorstudie 2006/2007)

<b>Schülerbefragung zur Informatik</b>		
Fachbereich Informationstechnik		ehler@oszimt.de
Nachname: _____	Klasse: ____	Datum: _____
1. In den letzten Stunden behandelten wir im Informatik-Unterricht das Modul (Nr.): _____		
----- allgemeine Fragen -----		
2. Das Fach Informatik (innerhalb der Technik) gefällt mir ...		
<input type="radio"/> sehr gut <input type="radio"/> gut <input type="radio"/> durchschnittlich <input type="radio"/> nicht so gut <input type="radio"/> nicht		
3. Der unterrichtende Lehrer (Hr. Francioni oder Hr. Ehlert) gefällt mir ...		
<input type="radio"/> sehr gut <input type="radio"/> gut <input type="radio"/> durchschnittlich <input type="radio"/> nicht so gut <input type="radio"/> nicht		
<input type="radio"/> keine Aussage		
4. Meine Motivation, die Themen des Faches Informatik zu verstehen (evt. auch durch verstärktes Lernen zu Hause) ist ...		
<input type="radio"/> sehr hoch <input type="radio"/> hoch <input type="radio"/> durchschnittlich <input type="radio"/> niedrig <input type="radio"/> sehr niedrig		
5. Ich fühle mich allgemein beim Unterricht im Fach Informatik eher ...		
<input type="radio"/> sehr wohl <input type="radio"/> wohl <input type="radio"/> normal <input type="radio"/> unwohl <input type="radio"/> sehr unwohl		
----- Fragen zum letzten Modul (Thema siehe Tafelanschrieb) -----		
6. Der Stoff des letzten Moduls hat zu meinem Gesamtverständnis (der Themen) des Faches Informatik ...		
<input type="radio"/> sehr viel beigetragen <input type="radio"/> viel beigetragen		
<input type="radio"/> durchschnittlich beigetragen <input type="radio"/> wenig beigetragen		
<input type="radio"/> überhaupt nicht beigetragen <input type="radio"/> kann ich nicht beurteilen		
7. Den Stoff des letzten Moduls empfand ich als ...		
<input type="radio"/> sehr leicht <input type="radio"/> leicht <input type="radio"/> mittelschwer <input type="radio"/> schwer <input type="radio"/> sehr schwer		
8. Unabhängig von der Schwere des Stoffes fand ich das Thema des letzten Moduls ...		
<input type="radio"/> sehr interessant <input type="radio"/> interessant <input type="radio"/> durchschnittlich		
<input type="radio"/> langweilig <input type="radio"/> sehr langweilig		
9. Ich fühlte mich speziell beim letzten Modul eher ...		
<input type="radio"/> sehr wohl <input type="radio"/> wohl <input type="radio"/> normal <input type="radio"/> unwohl <input type="radio"/> sehr unwohl		
----- Anmerkungen -----		
10. Anmerkungen (z.B. zur Lern- und Motivationssituation, zur Klassensituation etc.):		
_____		
_____		

Blatt - Schülerbefragung/Nov  
31.10.2006

## E) Daten zum subjektiven Erleben der Schüler (Vorstudie 2006/2007)

Klasse: OG61  
 Datum: 29.09.2006  
 Lfd. Nummer: 1  
 Modul-Nummer: 1  
 Thema: Einführung in die OOP mit BlueJ

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6101			2	2	2	1	1	3	2	2	
6102			2	1	1						
6103			2	2	3	3	1	1	2	2	
6104			2	2	3	2	2	3	2	2	
6105			1	2	1	1	1	1	2	1	
6106			2	1	2	1	2	3	2	2	
6107			3	3	3	4	3	3	2	3	
6108			1	2	2	2	1	2	3	2	
6109			2	2	3	2	4		2	3	
6110			1	2	2	2	2	2	3	3	
6111			1	1	2	2	1	2	2	2	
6112			2		2	3	2		3	3	
6113			1	1	1	1	2	1	1	1	
6114			2	2	2	3	2	2	3	3	
6115			1	1	2	1	1	2	3	1	
6116			1	2	2	2	2	3	3	3	
6117			2	2	4	4	2	3	5	4	
6118			2	3	4	3	3		4	3	
6119			3	3	4	3	3	4	2	3	
6120			2	2	3	3	2	1	1	3	
6121			2	2	3	3	2	2	2	3	
6122			2	2	3	3	2	3	3	3	
6123			2	1	1	2	2	3	3	1	
6124			2	1	2	1	2	3	2	2	
6125			1	1	2	1	2	2	1	1	
6126			2	1	2	2	1	2	3	2	
	Durchschnitt:	#DIV/0!	1,7892	1,76	2,34615	2,18	1,92	2,4091	2,44	2,32	#DIV/0!

Klasse: OG61  
 Datum: 10.11.2006  
 Lfd. Nummer: 3  
 Modul-Nummer: 2  
 Thema: Klassenkonzept: setter/getter/Konstruktoren, Trennung: Klasse/Anwendungsprogramm

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6101			1	1	1	1	1	2	1	1	
6102			2	1	1	2	2	3	1	3	
6103			3	2	3	3	3	4	1	3	
6104			2	1	3	2	3	3	2	2	
6105			1	1	1	1	1	2	1	1	
6106			2	2	3	2	2	3	2	3	
6107			3	3	3	4	4	4	2	4	
6108			1	2	2	2	3	2	1	1	
6109			2	2	2	2	3	2	2	3	
6110			1	1	1	1	1	3	1	1	
6111			2	2	3	2	1	3	2	3	
6112			3		3	3	3	3	2	2	
6113			1	1	1	1	1	1	1	1	
6114			1	1	2	2	1	2	2	2	
6115			2	2	2	2	2	3	2	2	
6116			1	2	2	3		3	2	3	
6117			4		4	5	1	3	2	4	
6118			3	2	2	3	2	4	2	3	
6119			1	2	1	2	2	2	2	2	
6120			2	2	2	3	3	1	2	3	
6121			2	2	2	2	2	2	2	2	
6122			3	2	2	3	2	4	2	3	
6123			1	1	1	1	1	2	1	1	
6124			1	1	2	1	2	2	1	1	
6125			1	2	2	1	2	1	2	1	
6126			2	1	2	2	2	3	2	2	
	Durchschnitt:	#DIV/0!	1,8462	1,625	2,03846	2,15385	2	2,5769	1,65385	2,19231	#DIV/0!



Klasse: OG61  
 Datum: 15.12.2006  
 Lfd. Nummer: 5  
 Modul-Nummer: 3  
 Thema: Methoden (inkl. Parametrisierter Konstruktor)

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6101			3	3	3	3	6	1	5	3	
6102			3	1	2	3	3	3	1	3	
6103											
6104			3	1	3	3	3	4	2	2	
6105			1	1	1	1	1	1	1	1	
6106			2	2	3	2	2	2	2	2	
6107			3	3	3	3	3	3	3	3	
6108			1	2	2	2	3	3	1	2	
6109			2	2	3	2	3	2	2	2	
6110			1	1	2	1	2	2	1	1	
6111			2	2	2	3	2	3	2	3	
6112			3		3	3	3	3	2	3	
6113			1	1	1	1	1	1	1	1	
6114			3	2	3	3	3	3	2	3	
6115			1	1	2	2	2	3	2	3	
6116											
6117			4		4	5	5	3	4	4	
6118			3	2	2	3	4	4	2	3	
6119			3	3	3	2	3	3	3	3	
6120											
6121			3	2	3	2	3	3	2	3	
6122			2	1	3	2	3	3	2	3	
6123			1	1	2	2	3	2	4	3	
6124			1	1	2	2	3	2	3	2	
6125			1	1	1	2	3	1	2	1	
6126											
Durchschnitt:	#DIV/0!	2,1364	1,65	2,40909	2,36364	2,9091	2,5	2,22727	2,45455	#DIV/0!	

Klasse: OG61  
 Datum: 02.02.2007  
 Lfd. Nummer: 7  
 Modul-Nummer: 5  
 Thema: Arrays und Strings

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6101			3	2	5	2	3	5	4	4	
6102			3	1	2	3	3	3	2	2	
6103											
6104			3	1	3	3	3	4	3	3	
6105			1	1	1	1	2	3	1	2	
6106			2	2	3	2	2	3	3	3	
6107											
6108			1	2	2	2	3	4	1	2	
6109			2	2	3	3	3	3	3	3	
6110			2	2	3	3	4	4	2	4	
6111			2	2	3	2	1	3	2	3	
6112			3		4	3		4	3	3	
6113			1	1	1	1	1	1	1	1	
6114			2	2	2	2	2	2	2	3	
6115											
6116			2	2	3	3	4	3	2	3	
6117											
6118			4	3	3	4	3	5	2	4	
6119											
6120			2	2	2	3	3	1	2	3	
6121			2	2	3	3	2	3	2	3	
6122			2	2	3	3	4	4	3	3	
6123			1	1	2	3	3	3	1	3	
6124											
6125			1	1	1	2	2	1	2	1	
6126			2	2	3	3	3	3	3	3	
Durchschnitt:	#DIV/0!	2,05	1,73684	2,6	2,55	2,6842	3,1	2,2	2,8	#DIV/0!	

Klasse: OG61  
 Datum: 16.02.2007  
 Lfd. Nummer: 8  
 Modul-Nummer: 6  
 Thema: Datenpersistenz (Dateien, Serialisierung)

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6101			3	3	3	3		3	3	3	
6102			2	1	3	3		4	2	4	
6103											
6104			3	1	3	3	3	3	3	3	
6105			1	1	1	1	1	2	1	1	
6106			2	2	3	2	2	3	2	2	
6107			5	5	5	5		5	5	5	
6108			1	2	2	2	2	3	1	2	
6109			2	2	3	3	3	2	3	3	
6110			1	1	3	2	3	3	3	3	
6111			2	2	2	3	2	3	2	3	
6112			3		3	3		3	3	3	
6113			1	1	2	2	4	4	3	4	
6114			2	2	3	3	3	3	3	3	
6115											
6116			2	2	3	3	2	3	2	3	
6117											
6118			3	2	1	2	2	4	1	3	
6119			4	5	5	5		5	5	4	
6120			2	2	2	3	3	1	2	3	
6121			3	2	3	3	3	3	3	3	
6122			3	2	4	3	4	4	4	3	
6123			1	1	1	2	2	2	1	2	
6124			2	1	3	2	3	2	1	2	
6125			1	1	1	1	2	1	1	1	
6126			2	2	3	3	3	3	3	3	
	Durchschnitt:	#DIV/0!	2,2174	1,95455	2,69565	2,69565	2,6111	3	2,47826	2,86957	#DIV/0!

Klasse: OG61  
 Datum: 30.03.2007  
 Lfd. Nummer: 9  
 Modul-Nummer: 07  
 Thema: Vererbung

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6101			2	1	4	3		3	3	3	
6102			3	1	4	4	4	3	2	4	
6103			4	2	3	4	3	3	3	4	
6104			4	1	4	3		3	3	3	
6105			1	1	1	1	1	3	1	2	
6106			3	3	3	2	2	3	2	2	
6107			3	3	2	3	3	3	2	3	
6108			1	1	2	2	2	4	1	2	
6109			2	2	3	3	3	2	3	3	
6110			2	2	3	3	2	3	3	3	
6111			3	2	3	3	2	3	3	4	
6112			3		3	3		4	2	3	
6113			1	1	2	2	3	3	2	3	
6114			2	2	3	2	2	2	2	3	
6115			2	2	2	2	2	3	2	2	
6116			2	2	3	3	4	3	2	3	
6117											
6118											
6119			3	3	3	2	2	2	2	2	
6120			1	2	3	3	3	1	2	3	
6121											
6122			2	2	4	4	3	4	3	3	
6123			2	1	3	3	4	3	5	3	
6124			2	2	3	3	4	2	3	3	
6125			1	1	1	1	3	1	1	1	
6126			3	2	3	3	2	3	2	3	
	Durchschnitt:	#DIV/0!	2,2609	1,77273	2,82609	2,69565	2,7	2,7826	2,34783	2,82609	#DIV/0!

Klasse: OG61  
 Datum: 03.11.2006  
 Lfd. Nummer: 2  
 Modul-Nummer: 10  
 Thema: Einführung in die IDE, Variablen, Konstanten, Datentypen, Wertzuweisungen

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6101			2	3	3	2	3	3	2	3	
6102			2	1	1	3	2	3	2	2	
6103			3	4	3	4	3	3	3	4	
6104			3	1	3	2	3	3	3	3	
6105			1	1	1	1	1	2	1	1	
6106			2	2	3	3	3	3	2	3	
6107			3	3	3	3	3	3	2	3	
6108											
6109			1	2	2	2	3	2	2	2	
6110			1	2	2	2	1	3	2	2	
6111			2	2	2	3	3	3	2	4	
6112			4		4	4		4	3	4	
6113			1	1	1	1	1	3	1	1	
6114			2	2	2	2	1	2	2	2	
6115			2	2	2	2	3	3	2		
6116			2		2	3		3	2	3	
6117			5	5	5	5	4	5	5	5	
6118			3		3	3	4	4	3	3	
6119			2	2	2	2	2	3	2	2	
6120			2	2	2	3	3	1	3	3	
6121			3	3	2	3	3	4	2	3	
6122			2	2	3	3	2	3	2	3	
6123			3	3	3	3	2	3	3	4	
6124			2	2	3	2	2	3	3	3	
6125			1	2	2	1	2	1	2	2	
6126			2	2	2	2	3	3	2	2	
Durchschnitt:	#DIV/0!	2,24	2,22727	2,44	2,56	2,4783	2,92	2,32	2,79167	#DIV/0!	

Klasse: OG61  
 Datum: 01.12.2006  
 Lfd. Nummer: 4  
 Modul-Nummer: 11 bzw. 12  
 Thema: Steuerstruktur: Sequenz und alle Selektionen

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6101			2	2	1	3	2	3	1	2	
6102			2	1	1	3	3	3	2	3	
6103											
6104			2	1	3	2	2	3	2	3	
6105			1	1	1	1	1	2	1	1	
6106			2	2	3	2	2	2	2	2	
6107			3	3	3	3	3	3	3	3	
6108			1	2	2	2	2	2	2	2	
6109			2	2	3	2	3	2	3	3	
6110			1	1	1	2	1	2	1	2	
6111			2	2	2	2	2	3	2	3	
6112			2		3	3	3	3	1	3	
6113			1	1	1	1	1	1	1	1	
6114			2	2	3	2	3	3	2	3	
6115			2	2	2	2	3	3	2	2	
6116			2	2	2	3	4	3	2	4	
6117			4		4	5	4	3	3	4	
6118			3		1	2	2	3	2	2	
6119			2	3	3	2	2	2	2	2	
6120			2	2	2	3	3	1	3	3	
6121			2	2	2	3	2	3	2	2	
6122			2	1	3	2	2	3	1	3	
6123			2	1	2	3	3	2	2	2	
6124			1	1	2	1	2	2	1	1	
6125			1	1	1	1	2	1	3	1	
6126			2	2	3	2	3	3	3	2	
Durchschnitt:	#DIV/0!	1,92	1,68182	2,16	2,28	2,4	2,44	1,96	2,36	#DIV/0!	

Klasse: OG61  
 Datum: 22.12.2006  
 Lfd. Nummer: 6  
 Modul-Nummer: 20  
 Thema: Kopf- und fußgesteuerte Schleife, Zählschleife inkl. Struktogramme

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6101			1	1	2	1	3	2	2	2	
6102			2	1	2	3	3	3	1	3	
6103			3	2	3	3	3	3	2	3	
6104			3	1	3	3	3	4	2	3	
6105			1	1	1	1	2	2	1	1	
6106			2	2	3	2	3	3	3	3	
6107			3	3	3	3	3	3	3	3	
6108			1	2	2	2	2	2	2	2	
6109			2	2	3	2	3	2	3	3	
6110			2	2	2	2	2	2	2	2	
6111			2	2	2	3	2	3	2	3	
6112			3		3	3	4	3	3	3	
6113			1	1	1	1	1	1	1	1	
6114			2	2	3	2	2	3	2	3	
6115											
6116			2	2	3	3	4	3	2	2	
6117			5		5	5	4	3	5	5	
6118											
6119			3	3	3	2	3	3	3	2	
6120			2	2	2	3	3	1	3	3	
6121			2	2	3	2	2	3	2	3	
6122			2	2	3	3	3	3	4	3	
6123			2	1	2	2	3	1	3	2	
6124											
6125			1	1	1	1	2	1	1	1	
6126											
Durchschnitt:	#DIV/0!	2,1364	1,75	2,5	2,36364	2,7273	2,4545	2,36364	2,54545	#DIV/0!	

Klasse: OG62  
 Datum: 22.02.2007  
 Lfd. Nummer: 6  
 Modul-Nummer: 01  
 Thema: Einführung in die OOP mit BlueJ

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6201			1	3	1	1	1	1	1	1	
6202			3	3	1	1	3	1	3	3	
6203			1	1	1	1	1	3	1	1	
6204			2	2	2	2	2	2	2	1	
6205			3	3	3	3		3	2	3	
6206			2	2	2	1	2	2	1	1	
6207			2	2	3	2	1	2	2	2	
6208			2	2	2	1	1	2	1	1	
6209			2	3	3	3	3	2	2	2	
6210			3	3	2	3	4	2	3	3	
6211											
6212			1	2	2	2	2	2	2	2	
6213			3		3	3	2	1	2	2	
6214			2	2	3	3	3	3	2	3	
6215			2		2	3	2	2	2	3	
6216			2	2	2	3	2	2	1	2	
6217			1	2	1	1	3	1	2	2	
6218			1	2	1	2	2	3	1	2	
6219			1	2	1	2	2	2	1	1	
6220			2	2	3	2	3	1	2	1	
6221			1	3	3	3	4	1	4	3	
6222			2	2	2	2	1	3	1	1	
6223			2	2	3	2	2	2	1	2	
6224											
6225											
6226			2	2	3	2	1	2	1	2	
6227			5		2	4	3	3	2	3	
Durchschnitt:	#DIV/0!	2	2,2381	2,125	2,16667	2,17391	2	1,75	1,95833	#DIV/0!	

Klasse: OG62  
 Datum: 29.03.2007  
 Lfd. Nummer: 7  
 Modul-Nummer: 02  
 Thema: Klassenkonzept: setter/getter/Konstruktoren, Trennung: Klasse/Anwendungsprogramm

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6201			1	3	1	1	1	2	1	1	
6202			2	2	2	2	3	2	2	3	
6203			2	2	4	3	3	3	2	3	
6204			2	2	3	4	3	3	3	4	
6205			2	2	3	4	5	5	3	4	
6206			2	2	2	3	2	3	2	3	
6207			2	3	2	2	2	2	1	2	
6208			2	2	3	2	2	2	2	2	
6209			2	2	3	3	3	3	2	3	
6210			3	3	2	3	4	2	3	3	
6211											
6212			1	2	2	2	1	3	2	3	
6213			3		3	3	4	3	4	3	
6214			2	2	3	3	2	3	2	3	
6215			2		2	3	3	2	2	3	
6216			2	2	3	3	2	3	3	2	
6217			1	1	2	1	1	2	2	2	
6218			1	1	1	1	2	3	1	2	
6219											
6220			2	2	2	2	3	2	2	2	
6221			1	2	3	3	3	3	2	4	
6222			2	3	2	2	1	3	1	2	
6223			2	2	3	2	2	3	3	3	
6224											
6225			2	2	3	2	2	3	2	3	
6226			2	2	3	3	2	4	2	4	
6227			5		2	5		4	4	5	
Durchschnitt:	#DIV/0!		2	2,09524	2,45833	2,58333	2,43478	2,83333	2,20833	2,875	#DIV/0!

Klasse: OG62  
 Datum: 18.01.2007  
 Lfd. Nummer: 5  
 Modul-Nummer: 03  
 Thema: Methoden (Deklaration, Rückgabewert, aktueller und formaler Parameter)

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6201			1	3	1	1	1	3	1	1	
6202			1	3	2	1	3	2	2	3	
6203											
6204			3	3	2	3	2	4	2	3	
6205											
6206			3	3	2	3	2	3	2	2	
6207			2	2	2	3	2	3	2	3	
6208			3	3	3	2	3	4	2	4	
6209			3	3	3	3	3	4	3	3	
6210			3	3	2	3	4	1	3	3	
6211											
6212			2	2	3	2	3	4	2	3	
6213			3		3	4	4	4	4	5	
6214											
6215			2		2	3	2	4	2	3	
6216			2	2	3	3	2	3	2	3	
6217			1	1	2	2	1	3	2	2	
6218			1	2	1	1	2	2	1	2	
6219			2	2	2	3	3	3	2	3	
6220			2	2	3	2	2	3	2	3	
6221			3	2	3	3	3	3	5	4	
6222			3	3	3	3	2	3	1	3	
6223			2	2	2	3	2	4	3	4	
6224											
6225			2	2	3	2	2	3	3	3	
6226			3	3	3	4		4	2	4	
6227			5		2	5	3	4	4	5	
Durchschnitt:	#DIV/0!		2,3636	2,42105	2,36364	2,68182	2,42857	3,22727	2,36364	3,13636	#DIV/0!

Klasse: OG62  
 Datum: 24.05.2007  
 Lfd. Nummer: 8  
 Modul-Nummer: 07  
 Thema: Vererbung

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6201			1	1	1	1	2	2	1	1	
6202			2	2	1	2	3	1	3	3	
6203			2	2	3	3	3	3	2	2	
6204			1	2	1	3	2	2	1	1	
6205											
6206			2	3	2	2	2	2	2	2	
6207			2	2	3	2	2	2	2	3	
6208			2	3	2	3	2	3	2	3	
6209			2	3	3	3	3	3	2	3	
6210			3	3	2	3	4	2	3	3	
6211											
6212			2	2	2	2	1	3	1	2	
6213			3	5	2	4	4	3	3	3	
6214			2	3	3	3	2	3	2	3	
6215			2		2	3	4	2	3	3	
6216			2	2	3	3	2	3	2	3	
6217											
6218			1	1	1	1	2	2	2	2	
6219			2	2	2	3	3	3	3	3	
6220			2	2	2	2	3	2	3	2	
6221			3	2	3	3	2	1	3	3	
6222			3		2	3	1	3	2	2	
6223			1	2	3	3	2	3	2	3	
6224											
6225			2	2	3	3	2	3	2	3	
6226			2	2	4	2	2	3	1	3	
6227											
	Durchschnitt:	#DIV/0!	2	2,3	2,27273	2,59091	2,40909	2,45455	2,13636	2,54545	#DIV/0!

Klasse: OG62  
 Datum: 28.09.2006  
 Lfd. Nummer: 1  
 Modul-Nummer: 10  
 Thema: Einführung in Java, Hallo Welt, IDE, Variablen, Konstanten, Wertzuweisung

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6201			1	2	1	1	3	4	1	3	
6202			2		2	3	4	4	1	2	
6203											
6204			1	2	3	1	3	3	2	3	
6205			4		2	3	2	1	3	3	
6206			2	1	2	3	3	2	3	2	
6207			2	2	3	2	3	3	2	3	
6208			1	2	2	1	1		3	2	
6209			2	2	2	3	3	3	2	3	
6210			2	3	1	3	3	5	1	3	
6211											
6212			1	2	3	2	2	2	2	3	
6213			3	1	2	3	3	3	3	3	
6214			2	2	3	3	2	2	3	3	
6215			2	2	2	3	2	2	3	3	
6216			2	2	2	1	3		2	2	
6217			1	1	1	2	1	2	3	2	
6218											
6219			2	1	2	2	2	2	2	2	
6220			1	1	2	2	2	3	1	3	
6221			1	2	1	2	3	3	2	3	
6222			2	2	3	2	3		2	2	
6223			1	1	2	2	2	2	3	2	
6224			2	2	2	3	2	2	2	3	
6225			1	1	2	1	2	3	2	2	
6226			2	2	3	3	2	1	3	3	
6227			3	2	2	2	3	2	3	2	
	Durchschnitt:	#DIV/0!	1,7917	1,72727	2,08333	2,20833	2,4583	2,5714	2,25	2,58333	#DIV/0!

Klasse: OG62  
 Datum: 02.11.2006  
 Lfd. Nummer: 2  
 Modul-Nummer: 11  
 Thema: Sequenz und Fallunterscheidung (inkl. Struktogramme)

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6201			2	3	1	1	3	1	1	2	
6202			2	2	1	3	4	1	3	2	
6203			1	1	2	2		2	2	2	
6204			2	2	2	3	2	3	3	4	
6205			3	3	3	4	2	3	2	3	
6206			2	2	2	2	3	2	1	3	
6207			2	2	3	2	3	2	3	3	
6208			1	2	2	3	3	2	3	3	
6209			2	2	3	2	3	3	2	3	
6210			3	3	2	3	4	1	4	3	
6211			3	5	3	4		3	3	4	
6212			2	2	2	2	2	2	2	2	
6213			3	2	3	3		3	3	4	
6214			2	2	3	3	3	3	2	3	
6215			2	2	2	3	2	3	2	2	
6216			2	2	2	3	2	3	2	3	
6217			1	1	1	2	2	2	2	2	
6218			1	1	1	2	2	3	1	2	
6219			2	2	2	3	2	3	2	3	
6220			1	1	2	1	3	2	1	2	
6221			1	3	3	2	4	1	4	3	
6222			1	2	2	2		2	3	1	
6223			1	2	2	2	3	3	2	3	
6224											
6225			2	2	3	2	3	2	3	2	
6226			2	2	3	3	2	3	2	2	
6227			4	3	2	4		4	3	4	
	Durchschnitt:	#DIV/0!	1,9231	2,16	2,19231	2,53846	2,7143	2,3846	2,34615	2,69231	#DIV/0!

Klasse: OG62  
 Datum: 20.11.2006  
 Lfd. Nummer: 3  
 Modul-Nummer: 12  
 Thema: Ein- und Zweiseitige Auswahl (inkl. Struktogramme)

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6201			2	3	1	2	1	1	1	1	
6202			2	2	2	3	4	2	3	3	
6203			1	2	2	2	1	3	2	3	
6204			2	2	2	3	3	3	4	3	
6205			2	2	3	3	1	3	2	2	
6206			2	3	2	3	2	2	2	2	
6207			2	2	2	2	3	2	2	3	
6208			2	3	2	3	2	3	2	3	
6209			2	2	3	3	3	3	3	3	
6210			3	3	2	3	5	1	3	3	
6211											
6212			2	2	3	3	3	3	2	2	
6213			4		3	4	4	4	4	4	
6214			2	2	3	3	2	3	2	2	
6215			2	2	2	3	2	3	2	3	
6216			3	2	3	3		4	3	3	
6217			1	2	1	2	2	2	1	2	
6218											
6219			2	2	2	2	1	3	2	2	
6220			2	2	3	2	3	2	2	2	
6221			1	2	2	3	4	1	2	3	
6222			2	3	2	3	3	2	3	3	
6223			1	2	2	2	2	3	2	3	
6224											
6225			1	2	1	2	2	2	2	2	
6226											
6227			5		2	5	4	4	4	5	
	Durchschnitt:	#DIV/0!	2,087	2,25	2,17391	2,78261	2,5909	2,5652	2,3913	2,69565	#DIV/0!


Klasse: OG62  
 Datum: 14.12.2006  
 Lfd. Nummer: 4  
 Modul-Nummer: 20  
 Thema: Schleifen

Nr	Name	Frage 1	Frage 2	Frage 3	Frage 4	Frage 5	Frage 6	Frage 7	Frage 8	Frage 9	Frage 10
6201			1	3	1	1	2	2	1	1	
6202			2	2	3	1	4	1	4	3	
6203			2	1	3	3	1	3	2	3	
6204			2	3	1	3	3	2	2	3	
6205			2	2	3	2	3	3	1	2	
6206			3	3	2	3	2	3	2	3	
6207			2	2	3	2	3	1	3	3	
6208			3	3	2	2	3	2	2	3	
6209			2	2	3	3	3	3	2	3	
6210			3	3	2	3	4	2	4	3	
6211			3	5	3	4	3	3	2	4	
6212			2	2	2	2	2	3	2	2	
6213			3		3	4	2	2	3	2	
6214			2	2	3	3	2	2	2	2	
6215			2		2	3	2	3	2	3	
6216			2	3	2	3	2	2	2	2	
6217			1	2	2	1	2	2	1	2	
6218											
6219			2	3	2	3	3	3	2	3	
6220			2	2	2	2	3	2	2	2	
6221											
6222			3	4	2	3	3	3	2	3	
6223											
6224											
6225			2	2	2	3	3	3	3	4	
6226			2	3	3	3	2	4	1	3	
6227			4		2	5		4	4	3	
	Durchschnitt:	#DIV/0!	2,2609	2,6	2,30435	2,69565	2,5909	2,5217	2,21739	2,69565	#DIV/0!



## F) Vergleichstest (Post-Test) inkl. Löser (Vorstudie 2006/2007)

In der Vorstudie gab es nur einen Vergleichstest und keinen Nachhaltigkeitstest.

	<b>Lernerfolgskontrolle (inkl. Löser)</b>		Informatik Einführungsphase
	Name:	Klasse:	Datum:
			Blatt Nr.: 1 / 1 lfd. Nr.:

In der folgenden Lernerfolgskontrolle werden Fragen zu 5 Themengebieten gestellt.

Zu jedem dieser Themengebiete gibt es ein Blatt mit maximal 10 Ankreuzaufgaben, die (wenn nicht anders angegeben) jeweils mit 1 Punkt bewertet werden und ein weiteres Blatt mit zusätzlichen Fragen mit insgesamt 10 Punkten.

**Hinweis: Bei den Multiple-Choice-Fragen ist immer genau nur eine Antwort richtig!**

Ein paar Tage vor oder nach diesem Test erhalten Sie noch Zusatzfragen zu einem sechsten Themengebiet. Die dort erzielten Punkte (max. 10) können Sie zusätzlich einbringen!

Lauf. Nr.	Themengebiet	Erreichbare Punkte	Erreichte Punkte	Note
1	Kontroll- und Datenstrukturen	20		
2	Struktogramme	20		
3	Methoden	20		
4	Objektorientierte Programmierung (OOP) (statisch)	20		
5	Unified Modeling Language (UML)	20		
Summe		100		
Extra 6	Objektorientierte Programmierung (OOP) (dynamisch)	10		
<b>Summe</b>		<b>„100“</b>		

Berlin, den \_\_\_\_\_

(Ehlert)

Notenspiegel:

	1	2	3	4	5	6	Ø
OG61							
OG62							

Name:	Klasse:	Datum:	Blatt Nr.: 2 / 2 lfd. Nr.:
-------	---------	--------	----------------------------

**1.1 Kontroll- und Datenstrukturen:**

1	Durch den <b>Datentyp</b> einer Variablen wird festgelegt, welche(n) ...	<input checked="" type="checkbox"/> Werte die Variable annehmen kann <input type="checkbox"/> Namen die Variable hat <input type="checkbox"/> Adresse die Variable im Speicher hat <input type="checkbox"/> Wert die Variable hat
2	Gegeben sind die beiden Quelltextzeilen: <pre>final double PI = 3.14; PI = 3.1415;</pre> Welche Aussage trifft zu?	<input type="checkbox"/> Compilerfehler in Zeile 1 <input checked="" type="checkbox"/> Compilerfehler in Zeile 2 <input type="checkbox"/> PI ist eine Variable <input type="checkbox"/> PI muss klein geschrieben werden
3	Welche Werte kann eine Variable vom <b>Datentyp</b> <code>byte</code> annehmen?	<input type="checkbox"/> 0 bis 255 <input type="checkbox"/> 0 bis 65535 <input checked="" type="checkbox"/> -128 bis 127 <input type="checkbox"/> Der Wert ist hardwareabhängig
4	Gegeben sind die beiden Quelltextzeilen: <pre>int a = 7; short b = a;</pre> Welche Aussage trifft zu?	<input checked="" type="checkbox"/> Compilerfehler in Zeile 2 <input type="checkbox"/> Beide Variablen benötigen den gleichen Speicherplatz <input type="checkbox"/> a belegt 16 Bit <input type="checkbox"/> b belegt 8 Bit
5	Gegeben ist der folgende Vergleich: <pre>if (wahrheit == false) {}</pre> Die <b>verkürzte Schreibweise</b> ist wie folgt:	<input type="checkbox"/> <code>if (wahrheit) {}</code> <input type="checkbox"/> <code>if (wahrheit &lt;&gt; true) {}</code> <input checked="" type="checkbox"/> <code>if (!wahrheit) {}</code> <input type="checkbox"/> <code>if (wahrheit!) {}</code>
6	Gegeben ist die folgende Variablendeklaration: <pre>String s = "Test";</pre> Welche weitere Quelltextzeile führt zu einem <b>Compilerfehler</b> ?	<input type="checkbox"/> <code>if (s == "Test") {}</code> <input type="checkbox"/> <code>if (s == "Hallo") {}</code> <input type="checkbox"/> <code>if (s.equals("Test")) {}</code> <input checked="" type="checkbox"/> keiner dieser Quelltextzeilen
7	<pre>int[] x = new int[25];</pre> Welche Aussage trifft nach der Ausführung der Programmzeile zu?	<input checked="" type="checkbox"/> <code>x.length</code> hat den Wert 25 <input type="checkbox"/> <code>x[0]</code> hat den Wert null <input type="checkbox"/> <code>x[25]</code> hat den Wert 0 <input type="checkbox"/> <code>x[24]</code> ist nicht definiert
8	Welche <b>Kontrollstruktur</b> gibt es?	<input type="checkbox"/> dreiseitige Auswahl <input type="checkbox"/> for-Auswahl <input type="checkbox"/> if-Schleife <input checked="" type="checkbox"/> Sequenz (Folge)
9	Welche Antwort enthält <b>keine</b> Endlosschleife?	<input type="checkbox"/> <code>for ( ; ; ) {}</code> <input checked="" type="checkbox"/> <code>do { } while (false);</code> <input type="checkbox"/> <code>while (true) { }</code> <input type="checkbox"/> <code>int y = 7; while (y == 7) { }</code>
10	<pre>for (int i = 1; i &lt; 10; i++)     if (i &lt; 5) {         a++;         b++;     } // if</pre> Welche <b>Kontrollstruktur</b> findet sich wieder?	<input type="checkbox"/> fußgesteuerte Schleife <input checked="" type="checkbox"/> einseitige Auswahl <input type="checkbox"/> zweiseitige Auswahl <input type="checkbox"/> Fallunterscheidung

Name:

Klasse:

Datum:

Blatt Nr.: 3/3 lfd. Nr.:

1.2 Kontroll- und Datenstrukturen:

1. Gegeben ist die folgende lückenhafte Variablendeklaration:

4P

```
char      a = 'a';
float     b = 7.0f;
boolean   c = (5 == 7);      4 * 1P
String    d = "Hallo";
```

Ergänzen Sie den Quelltext in den 4 Lücken!

2. Gegeben ist der folgende Quelltextausschnitt:


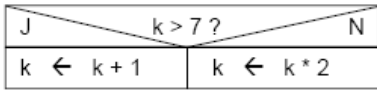


6P

```
int a = 7;
char c = 'a';    // oder 'b' oder 'x'
switch (c) {
    case 'a':
        while (a >= 1)
            a = a - 2;
            a = 3 * a;
            break;
    case 'b':
        for (int i = 10; i > 5; i--)
            a = a + 3;
            break;
    default:
        a = a % 3;
        break;
} // switch
if (a > 15)
    a--;
else
    a++;
System.out.println("a: " + a);
```

Welcher Wert wird für a ausgegeben, bei den folgenden Werten für die Variable c?

c	a
'a'	-2
'b'	21
'x'	2

3 \* 2P

		<b>Lernerfolgskontrolle</b> <b>(inkl. Löser)</b>		<b>Informatik</b> Einführungsphase	
Name:		Klasse:		Datum:	
				Blatt Nr.: 4 / 4 lfd. Nr.:	
<b>2.1 Struktogramme:</b>					
1	 <p>Innerhalb der <b>Selektionen</b> ist dies eine ...</p>			<input type="radio"/> einseitige Auswahl <input checked="" type="radio"/> zweiseitige Auswahl <input type="radio"/> Fallunterscheidung	
2	 <p>Innerhalb der <b>Iterationen</b> ist dies eine ...</p>			<input type="radio"/> kopfgesteuerte Schleife <input checked="" type="radio"/> fußgesteuerte Schleife <input type="radio"/> Endlosschleife	
3	 <p>Dies ist die <b>Darstellung</b> einer ...</p>			<input type="radio"/> Selektion (Auswahl) <input type="radio"/> Iteration (Wiederholung) <input checked="" type="radio"/> Sequenz (Folge)	
4	Gegeben ist die folgende Quelltextzeile: <pre>for (int i = 1; i &lt; 7; i++) {}</pre> Diese <b>Zählschleife</b> wird im Struktogramm mit Hilfe welchen Strukturelements dargestellt?			<input checked="" type="radio"/> kopfgesteuerte Schleife <input type="radio"/> fußgesteuerte Schleife <input type="radio"/> Endlosschleife	
5	<b>Variablendeklarationen</b> werden im Struktogramm ...			<input checked="" type="radio"/> nicht angegeben <input type="radio"/> angegeben <input type="radio"/> je nach Bedarf angegeben	
6 - 7	<pre>a ← 3 * 8 b ← a / 4 solange (b &gt; 1)   a ← a - 2   b ← b - 1</pre>			In der Variablen a steht am Ende des Algorithmus der Wert <input type="radio"/> 10 <input type="radio"/> 12 <input checked="" type="radio"/> 14 <input type="radio"/> 16 <input type="radio"/> keine der angegebenen Werte <b>2P</b>	
8 - 10	<pre>i ← (2 + 3) * 3 i ← i - 2 J i &lt; 12 ? N i ← i - 5 i ← i + 1 solange (i &gt; 8)</pre>			Wie oft wird der Schleifenkörper durchlaufen? <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> unendlich oft <b>3P</b>	
Ehler 2007		<a href="mailto:ehler@oszint.de">ehler@oszint.de</a>		<a href="http://www.oszint.de">www.oszint.de</a>	

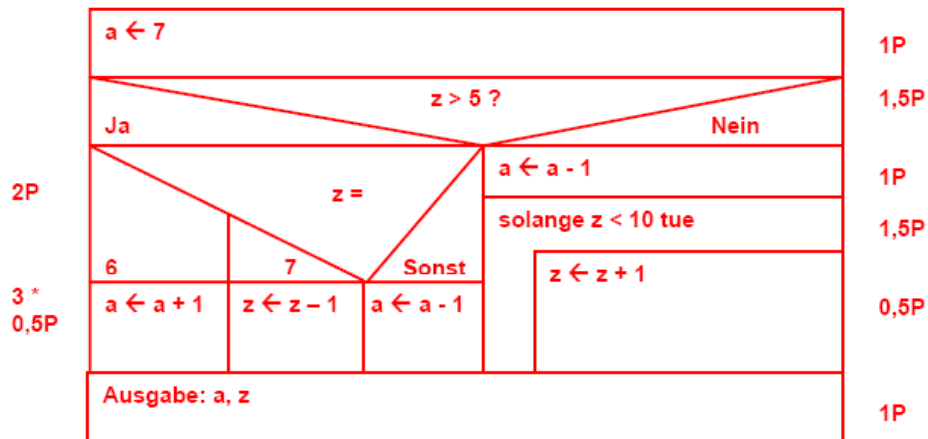
**2.2 Struktogramme:**


1. Gegeben ist die folgende Methode:

10P

```
public static void berechneIrgendwas(int z) {
    int a;
    a = 7;
    if (z > 5) {
        switch(z) {
            case 6: a += 1; break;
            case 7: z--; break;
            default: a -= 1; break;
        }
    }
    else {
        a = a - 1;
        while (z < 10)
            z++;
    }
    System.out.println("a: " + a + " z: " + z);
}
```

**Zeichnen** Sie das **Struktogramm** dieser Methode! Bitte nehmen Sie Eintragungen im Struktogramm relativ **programmiersprachenunabhängig** vor.



		<b>Lernerfolgskontrolle</b> <b>(inkl. Löser)</b>		<b>Informatik</b> Einführungsphase	
Name:		Klasse:		Datum:	
				Blatt Nr.: 6 / 6 lfd. Nr.:	
<b>3.1 Methoden:</b>					
1	Beim <b>Aufruf einer Methode</b> im Anwendungsprogramm findet man <b>immer</b> ...	<input type="radio"/> aktuelle Parameter <input type="radio"/> formale Parameter <input checked="" type="radio"/> eine runde Klammer auf und zu <input type="radio"/> eine Zuweisung			
2	In welcher Quelltextzeile findet sich <b>kein Methodenaufruf</b> ?	<input type="radio"/> name1 (); <input type="radio"/> a = name2 (b, c, d) ; <input type="radio"/> name3 (e) ; <input checked="" type="radio"/> f = name4 ;			
3	Welche Aussage trifft für die folgende <b>Quelltextzeile</b> zu: if (isGeldAlle ()) {}	<input checked="" type="radio"/> Die Methode hat einen Boolean-Rückgabewert <input type="radio"/> Die Methode hat keinen Rückgabewert <input type="radio"/> Der Methode fehlt ein Parameter <input type="radio"/> Sie führt zu einem Compilerfehler			
4	Bei der <b>Deklaration einer Methode</b> findet man im Methodenkopf <b>immer</b> ...	<input type="radio"/> einen formaler Parameter <input type="radio"/> einen aktuellen Parameter <input type="radio"/> den Zugriffsspezifizierer "public" <input checked="" type="radio"/> einen Methodennamen			
5	Gegeben ist der folgende Quelltextauszug: double a; a = berechneUmfang (4.7); Welche Aussage ist richtig?	<input checked="" type="radio"/> der aktuelle Parameter ist ein Double-Wert <input type="radio"/> die Methode kann nicht statisch sein <input type="radio"/> die Methode hat keinen Rückgabewert <input type="radio"/> vor dem Wert 4.7 fehlt der Datentyp double			
6	Die <b>statische Methode</b> mit dem Kopf: public static void tueWas (int z) steht im Anwendungsprogramm "über" der main-Methode. Wie kann diese Methode in der main-Methode aufgerufen werden?	<input type="radio"/> tueWas (int i) ; <input checked="" type="radio"/> tueWas (7) ; <input type="radio"/> int i = 8 ; int j = tueWas (i) ; <input type="radio"/> tueWas () ;			
7	<b>Methoden in Klassen</b> ...	<input type="radio"/> haben nie einen Rückgabewert <input type="radio"/> haben immer Parameter <input checked="" type="radio"/> sind in der Regel öffentlich <input type="radio"/> werden in der Regel groß geschrieben			
8	<b>Konstruktoren</b> und <b>Verwaltungsoperationen</b> („Setter“ und „Getter“) sind ...	<input checked="" type="radio"/> spezielle Methoden <input type="radio"/> keine Methoden <input type="radio"/> alles Methoden ohne Parameter <input type="radio"/> alles Methoden ohne Rückgabewert			
9	In der Regel werden in der OOP im Anwendungsprogramm <b>Methoden auf Objekten</b> wie folgt aufgerufen:	<input checked="" type="radio"/> objektname.methodenname () <input type="radio"/> Klassenname.methodenname () <input type="radio"/> methodenname () <input type="radio"/> Programmname.methodenname ()			
10	Welche Aussage über <b>Klassenmethoden</b> trifft zu?	<input type="radio"/> Sie besitzen immer ein Klassenattribut <input type="radio"/> Sie haben nie einen Rückgabewert <input checked="" type="radio"/> Aufruf erfolgt in der Regel mit: Klassenname.methodenname () <input type="radio"/> Aufruf erfolgt mit: methodenname ()			
Ehler 2007		<a href="mailto:ehler@oszimt.de">ehler@oszimt.de</a>		www.oszimt.de	

Name:

Klasse:

Datum:

Blatt Nr.: 7 / 7 lfd. Nr.:

### 3.2 Methoden:

1. Gegeben sind die folgenden Quelltextzeilen des Anwendungsprogramms, die in der main-Methode stehen: **3P**

```
...
int[] zahlenfeld = new int[10];
initialisiereFeld(zahlenfeld);
kleinsteZahl = ermittleKleinsteZahl(zahlenfeld);
...
```

Wie lautet der **Methodenkopf** der statischen Methode „ermittleKleinsteZahl (...)“?

**Hinweis:** Nur **eine Quelltextzeile** ist gefragt!

```
0,5P      1P                      1P   0,5P
public static int ermittleKleinsteZahl(int[] feld)
```

2. In der Klasse „Unsinn“ gibt es die folgende Methode: **4P**

```
public void tueIrgendwas(String s, double f, boolean b) {
    ...
}
```

In einem Anwendungsprogramm wird in der Main-Methode das Objekt „meinUnsinn“ erzeugt.

Rufen Sie auf diesem Objekt die Methode „tueIrgendwas(...)“ mit aktuellen Parametern (Übergabe von konstanten Werten) so auf, dass es keine Compilerfehler gibt!

**Hinweis:** Eine Quelltextzeile ist gefragt!

```
0,5P      0,5P      1P      1P   1P
meinUnsinn.tueIrgendwas("Hallo",4.7,true);
```

3. In der Klasse „Formelsammlung“ soll eine Methode „quadriere“ implementiert werden, die das Quadrat der übergebenen Double-Zahl zurückgibt. Geben Sie den **3P**

**Quelltext der Methode „quadriere“** an (Methodenkopf und Implementierung)!

**Hinweis:** Zwei bis drei Quelltextzeilen können genügen!

```
1P                      1P
public double quadriere(double zahl) {
    return zahl*zahl; 1P
}
```

Name:

Klasse:

Datum:

Blatt Nr.: 8 / 8 lfd. Nr.:

**4.1 Objektorientierte Programmierung** (statisch):

1	Welche Aussage über <b>Klassen</b> ist richtig?	<input type="radio"/> Eine Klasse ist in einem bestimmten Zustand <input checked="" type="radio"/> Eine Klasse ist der Bauplan für Objekte <input type="radio"/> Eine Klasse ist die Summe der Objekte <input type="radio"/> Eine Klasse hat immer Attribute
2	Wozu braucht man die <b>Verwaltungsmethoden</b> („Setter“ und „Getter“)?	<input checked="" type="radio"/> Um auf die gekapselten Attribute zugreifen zu können <input type="radio"/> Weil diese immer deklariert werden müssen <input type="radio"/> Weil die Attribute öffentlich sind <input type="radio"/> Weil die Methoden öffentlich sind
3	Wozu dienen <b>Konstruktoren</b> ?	<input checked="" type="radio"/> Um ein Objekt zu initialisieren <input type="radio"/> Um eine Vererbungshierarchie zu ermöglichen <input type="radio"/> Um eine Klasse zu initialisieren <input type="radio"/> Um ein Klasse deklarieren zu können
4	Welche Aussage über <b>Objekte</b> ist richtig?	<input type="radio"/> Von jeder Klasse gibt es mindestens ein Objekt <input type="radio"/> Objekte können sich nicht mit anderen Objekten austauschen <input checked="" type="radio"/> Objekte befinden sich in einem Zustand <input type="radio"/> Objekte haben keinen Zugriff auf sich selbst
5	Welche Verwaltungsmethode bzw. welcher Konstruktor liefert in jedem Fall einen <b>Rückgabewert</b> ?	<input type="radio"/> parameterloser Konstruktor <input type="radio"/> parametrisierter Konstruktor <input checked="" type="radio"/> „Getter“ <input type="radio"/> „Setter“
6	Welche Methode bzw. welcher Konstruktor hat mindestens einen <b>Parameter</b> ?	<input type="radio"/> parameterloser Konstruktor <input type="radio"/> „Getter“ (Get-Methode) <input checked="" type="radio"/> „Setter“ (Set-Methode)
7	Welche Aussage zu <b>Attributen</b> ist richtig?	<input checked="" type="radio"/> Attribute sind vergleichbar mit Variablen <input type="radio"/> Ein Objekt deklariert Attribute <input type="radio"/> In einer Klasse haben die Attribute Werte <input type="radio"/> Attribute haben keinen Datentyp
8	Welche Aussage ist richtig? Eine <b>abstrakte Klasse</b> ...	<input type="radio"/> enthält immer eine abstrakte Methode <input checked="" type="radio"/> kann nicht instanziiert werden <input type="radio"/> ist ein Interface <input type="radio"/> hat keine Attribute
9	Eine Oberklasse hat <b>2 Attribute</b> . Wie viele Attributswerte hat ein Objekt der Unterklasse?	<input type="radio"/> maximal 2 <input type="radio"/> genau 2 <input checked="" type="radio"/> mindestens 2 <input type="radio"/> kann man so nicht sagen
10	Welche Aussage ist richtig? Eine <b>abstrakte Methode</b> in einer Oberklasse ...	<input type="radio"/> ist verboten <input type="radio"/> muss in der Oberklasse implementiert werden <input checked="" type="radio"/> macht die Oberklasse automatisch zu einer abstrakten Klasse <input type="radio"/> enthält keine Parameter



Name:

Klasse:

Datum:

Blatt Nr.: 9 / 9 lfd. Nr.:

**4.2 Objektorientierte Programmierung** (statisch):

1. Die Klasse „Reisebuero“ hat u.a. das Attribut „anzahlKunden“ vom Datentyp „int“. **3P**  
Wie lautet der vollständige Quelltext der „**Setter**“-**Verwaltungsmethode**?

```

0,5P 0,5P 0,5P 0,5P
public void setAnzahlKunden(int anzahlKunden) {
    this.anzahlKunden = anzahlKunden; 1P
}

```

2. Die Klasse „Mensch“ hat das Attribut „vorname“ vom Datentyp „String“. **3P**  
Wie lautet der vollständige Quelltext der „**Getter**“-**Verwaltungsmethode**?

```

0,5P 1P 0,5P
public String getVorname() {
    return this.vorname; 1P
}

```

3. Die Klasse „Fahrrad“ hat die Attribute „groesse“ (Datentyp int) und „hersteller“ (Datentyp String). **2P**  
Wie lautet der **Methodenkopf** des vollparametrisierten Konstruktors?

```

0,5P 0,5P 0,5P 0,5P
public Fahrrad(int groesse, String hersteller)

```

4. Wie lautet die Quelltextzeile im Anwendungsprogramm zum **Erzeugen des Objektes** „meinAuto“? **2P**

```

0,5P 0,5P 0,5P 0,5P
Auto meinAuto = new Auto();

```

Name:	Klasse:	Datum:	Blatt Nr.: 10 / 10 lfd. Nr.:
-------	---------	--------	------------------------------

**5.1 Unified Modeling Language (UML):**

1	<div style="border: 1px solid black; width: 100px; height: 20px; margin: 0 auto; text-align: center; padding: 2px;"><b>einKunde</b></div> <p>Dies ist die UML-Darstellung eines/einer ...</p>	<input type="radio"/> Klasse <input checked="" type="radio"/> Objekts <input type="radio"/> Attributs <input type="radio"/> Methode																
2	<div style="border: 1px solid black; width: 100px; height: 20px; margin: 0 auto; text-align: center; padding: 2px;"><b>Haus</b></div> <p>Dies ist die UML-Darstellung eines/einer ...</p>	<input checked="" type="radio"/> Klasse <input type="radio"/> Objekts <input type="radio"/> Attributs <input type="radio"/> Methode																
3	<p>Wie wird im UML-OOD-Klassendiagramm die <b>Verwaltungsoperation („Getter“)</b> der Klasse „Mensch“ für das Attribut „alter“ dargestellt?</p>	<input type="radio"/> - getAlter(void): int <input type="radio"/> - getAlter(int): void <input checked="" type="radio"/> + getAlter(void): int <input type="radio"/> + getAlter(int): void																
4	<p>Wie wird im UML-OOD-Klassendiagramm die <b>Verwaltungsoperation („Setter“)</b> der Klasse „Mitarbeiter“ für das Attribut „gehalt“ dargestellt?</p>	<input type="radio"/> - setGehalt(void): float <input type="radio"/> - setGehalt(float): void <input type="radio"/> + setGehalt(void): float <input checked="" type="radio"/> + setGehalt(float): void																
5	<p>Wie wird im UML-OOD-Klassendiagramm der <b>parameterlose Konstruktor</b> der Klasse „Auto“ dargestellt?</p>	<input checked="" type="radio"/> + Auto() <input type="radio"/> + Auto(): int <input type="radio"/> + Auto(): void <input type="radio"/> + Auto(void): void																
6	<p><b>Methoden</b> werden im UML-Objektdiagramm ...</p>	<input type="radio"/> angegeben <input checked="" type="radio"/> nicht angegeben <input type="radio"/> je nach Bedarf angegeben																
7	<p>Wie ist die gebräuchlichste Verteilung der <b>Zugriffsspezifizierer</b> im UML-OOD-Klassendiagramm?</p>	<table style="width:100%; border: none;"> <tr> <td style="width:50%; vertical-align: top;"> <input type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>- ...</td></tr> <tr><td>- ...</td></tr> </table> </td> <td style="width:50%; vertical-align: top;"> <input type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>+ ...</td></tr> <tr><td>+ ...</td></tr> </table> </td> </tr> <tr> <td style="vertical-align: top;"> <input checked="" type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>- ...</td></tr> <tr><td>+ ...</td></tr> </table> </td> <td style="vertical-align: top;"> <input type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>+ ...</td></tr> <tr><td>- ...</td></tr> </table> </td> </tr> </table>	<input type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>- ...</td></tr> <tr><td>- ...</td></tr> </table>	Klassenname	- ...	- ...	<input type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>+ ...</td></tr> <tr><td>+ ...</td></tr> </table>	Klassenname	+ ...	+ ...	<input checked="" type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>- ...</td></tr> <tr><td>+ ...</td></tr> </table>	Klassenname	- ...	+ ...	<input type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>+ ...</td></tr> <tr><td>- ...</td></tr> </table>	Klassenname	+ ...	- ...
<input type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>- ...</td></tr> <tr><td>- ...</td></tr> </table>	Klassenname	- ...	- ...	<input type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>+ ...</td></tr> <tr><td>+ ...</td></tr> </table>	Klassenname	+ ...	+ ...											
Klassenname																		
- ...																		
- ...																		
Klassenname																		
+ ...																		
+ ...																		
<input checked="" type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>- ...</td></tr> <tr><td>+ ...</td></tr> </table>	Klassenname	- ...	+ ...	<input type="radio"/> <table border="1" style="width: 80px; height: 40px; border-collapse: collapse; text-align: center; font-size: 8px;"> <tr><td>Klassenname</td></tr> <tr><td>+ ...</td></tr> <tr><td>- ...</td></tr> </table>	Klassenname	+ ...	- ...											
Klassenname																		
- ...																		
+ ...																		
Klassenname																		
+ ...																		
- ...																		
8	<p>Wie wird im UML-OOD-Klassendiagramm das <b>gekapselte Attribut</b> „alter“ dargestellt?</p>	<input checked="" type="radio"/> - alter: int <input type="radio"/> - int: alter <input type="radio"/> + alter: int <input type="radio"/> + int: alter																
9	<p>Wie wird im UML-Objektdiagramm das <b>Attribut</b> „temperatur“ dargestellt?</p>	<input type="radio"/> 37.4: temperatur <input type="radio"/> temperatur: 37.4 <input type="radio"/> 37.4 = temperatur <input checked="" type="radio"/> temperatur = 37.4																
10	<p>Welcher <b>„Vererbungs Pfeil“</b> ist im UML-Klassendiagramm anstelle der gestrichelten Linie vorzusehen?</p> <div style="margin: 10px 0;"> <table style="width: 100%; border: none; text-align: center;"> <tr> <td style="width: 25%; border: 1px solid black; padding: 5px;">Oberklasse</td> <td style="width: 25%; border: 1px solid black; padding: 5px;">Unterklasse</td> </tr> <tr> <td colspan="2" style="text-align: center;"> </td> </tr> </table> </div>	Oberklasse	Unterklasse			<table style="width: 100%; border: none; text-align: center;"> <tr> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input checked="" type="radio"/></td> <td><input type="radio"/></td> </tr> <tr> <td>↓</td> <td>↑</td> <td>↑</td> <td>↓</td> </tr> </table>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	↓	↑	↑	↓				
Oberklasse	Unterklasse																	
<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>															
↓	↑	↑	↓															

Name:

Klasse:

Datum:

Blatt Nr.: 11 / 11 lfd. Nr.:

**5.2 Unified Modeling Language (UML):**

1. Die Klasse „Surfbrett“ hat die Attribute „laenge“ (Datentyp double) und „hersteller“ (Datentyp String). **7P**

Zeichnen Sie das vollständige **UML-OOD-Klassendiagramm** inklusive dem parameterlosen und dem vollparametrisierten Konstruktor und allen Verwaltungsoperationen („Setter“ und „Getter“)!

<b>Surfbrett</b>	<b>0,5P</b>
- laenge: double	0,5P
- hersteller: String	0,5P
+ Surfbrett()	0,5P
+ Surfbrett(double,String)	1P
+ setLaenge(double): void	1P
+ getLaenge(): double	1P
+ setHersteller(String): void	1P
+ getHersteller(): String	1P

2. Ein Schüler sagt: „Die Katze ist 3 Jahre alt und sie ist schwarz.“ **3P**  
Zeichnen Sie das **UML-Objektdiagramm** dieser Katze!

<u>dieKatze: Katze</u>	1P
alter = 3	1P
farbe = schwarz	1P

Name:	Klasse:	Datum:	Blatt Nr.: 12 / 12 lfd. Nr.:
-------	---------	--------	------------------------------

**6. Objektorientierte Programmierung** (dynamisch):

Gegeben ist der folgende Quelltext (zur Übersichtlichkeit wurden die Klassen `Mensch` und `Fahrrad` in das Anwendungsprogramm integriert):

```
public class DynamischeOOP {

    public static class Mensch {

        private String name;
        private Mensch ehpartner;
        private int anzahlFahrraeder;
        private Fahrrad[] fahrraeder;

        public Mensch() {
            this("Janine");
        }

        public Mensch(String name) {
            this.name = name;
            this.anzahlFahrraeder = 0;
            fahrraeder = new Fahrrad[100];
        }

        public void setName(String name) {
            this.name = name;
        }

        public String getName() {
            return this.name;
        }

        public int getAnzahlFahrraeder() {
            return this.anzahlFahrraeder;
        }

        public void heirate(Mensch einMensch) {
            ehpartner = einMensch;
        }

        public void kaufe(Fahrrad einFahrrad) {
            this.fahrraeder[anzahlFahrraeder] = einFahrrad;
            this.anzahlFahrraeder++;
        }

        public Mensch getEhpartner() {
            return this.ehpartner;
        }

        public Fahrrad getFahrrad(int i) {
            return fahrraeder[i];
        }

    } // Klasse Mensch
}
```

Name:

Klasse:

Datum:

Blatt Nr.: 13 / 13 lfd. Nr.:

```
public static class Fahrrad {

    private String farbe;

    public Fahrrad() {
        this.farbe = ("unbekannt");
    }

    public Fahrrad(String farbe) {
        this.farbe = farbe;
    }

    public void setFarbe(String farbe) {
        this.farbe = farbe;
    }

    public String getFarbe() {
        return this.farbe;
    }

} // Klasse Fahrrad

public static void main(String[] args) {

    Mensch ersterMensch = new Mensch("Klaus");
    Mensch zweiterMensch = new Mensch();
    Mensch dritterMensch = new Mensch();
    Mensch vierterMensch = new Mensch();

    zweiterMensch.setName("Harald");
    dritterMensch.setName("Susi");

    Fahrrad erstesFahrrad = new Fahrrad("rot");
    Fahrrad zweitesFahrrad = new Fahrrad();
    Fahrrad drittesFahrrad = new Fahrrad("blau");

    ersterMensch.kaufe(zweitesFahrrad);
    dritterMensch.kaufe(erstesFahrrad);
    dritterMensch.kaufe(drittesFahrrad);

    zweiterMensch.heirate(dritterMensch);
    dritterMensch.heirate(zweiterMensch);

    zweitesFahrrad.setFarbe("gruen");
    drittesFahrrad.setFarbe("gelb");

    System.out.println(ersterMensch.getName());
    System.out.println(zweiterMensch.getEhepartner().getName());
    int anzahl = dritterMensch.getAnzahlFahrraeder();
    for (int i = 0; i < anzahl; i++)
        System.out.println(i+1 + ". Fahrrad: " + dritterMensch.getFahrrad(i).getFarbe());
    System.out.println(vierterMensch.getName());

} // main

} // Klasse DynamischeOOP
```

Name:

Klasse:

Datum:

Blatt Nr.: 14 / 14 lfd. Nr.:

1. **Wie viele Objekte** werden zur Laufzeit durch die main-Methode erzeugt? **2P**

**7 (2P)**

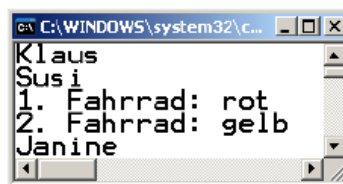
2. **Wie viele Methodenaufrufe auf Objekten** erfolgen zur Laufzeit in der main-Methode? **3P**

Achtung:

- Die Häufigkeit des Durchlaufes des Schleifenkörpers der Zählschleife ist vom Wert der Variablen `anzahl1` abhängig (0..100) und somit auch die Anzahl der Methodenaufrufe durch die Zählschleife!
- Bitte nicht die Konstruktoraufrufe mitzählen!

**18 (3P, ab 15 2P, ab 13 1P)**

3. Wie lautet genau die Bildschirmausgabe? **5P**



```
C:\WINDOWS\system32\c...  
Klaus  
Susi  
1. Fahrrad: rot  
2. Fahrrad: gelb  
Janine
```

**5 \* 1P**

## G) Ergebnisse des Vergleichstests (Vorstudie 2006/2007)

OG61 (OOP-First): Gesamtergebnis

OG61	1.Thema	2. Thema	3. Thema	4. Thema	5. Thema	6. Thema	Summe	Prozentual
6101	11,0	12,0	13,0	14,0	19,5	2,0	71,5	65,0
6102	9,0	13,0	12,5	14,5	14,5	3,0	66,5	60,5
6103	9,0	12,0	10,0	6,5	9,5	4,0	51,0	46,4
6104	7,0	6,5	8,0	17,0	14,0	4,0	56,5	51,4
6105	14,0	11,0	12,0	16,5	16,5	6,5	76,5	69,5
6106	11,0	11,0	12,5	15,5	19,5	4,0	73,5	66,8
6107	13,0	17,5	11,0	16,0	18,0	3,0	78,5	71,4
6108	13,0	14,0	16,0	14,5	13,0	5,0	75,5	68,6
6109	17,0	19,5	19,0	17,0	18,0	8,5	99,0	90,0
6110	9,0	5,0	10,0	13,5	11,5	5,5	54,5	49,5
6111								
6112	11,0	16,0	6,5	12,0	12,5	5,0	63,0	57,3
6113	20,0	19,5	15,5	19,5	18,5	9,0	102,0	92,7
6114	16,0	16,0	9,0	16,5	12,0	6,0	75,5	68,6
6115	6,0	4,0	6,0	9,0	8,5	2,0	35,5	32,3
6116	11,0	12,0	8,5	7,5	6,5	5,5	51,0	46,4
6117								
6118								
6119	12,0	16,0	15,0	18,0	19,0	6,0	86,0	78,2
6120	14,0	16,0	18,0	18,0	17,5	8,0	91,5	83,2
6121	9,0	5,5	7,0	6,0	7,5	3,5	38,5	35,0
6122	15,0	19,0	15,5	17,0	17,0	6,0	89,5	81,4
6123	13,0	17,0	14,0	15,5	15,0	5,0	79,5	72,3
6124	14,0	13,0	10,0	11,5	12,0	2,5	63,0	57,3
6125	16,0	16,0	17,5	20,0	15,0	7,0	91,5	83,2
6126	4,0	9,5	5,0	13,5	10,0	3,5	45,5	41,4
6127	16,0	20,0	14,0	16,0	20,0	6,0	92,0	83,6
<b>Durchschnitt</b>	<b>12,1</b>	<b>13,4</b>	<b>11,9</b>	<b>14,4</b>	<b>14,4</b>	<b>5,0</b>	<b>71,1</b>	<b>64,7</b>
<b>Prozentual</b>	<b>60</b>	<b>67</b>	<b>59</b>	<b>72</b>	<b>72</b>	<b>50</b>	<b>65</b>	<b>65</b>

Die Prozentzahlen vom 1. und 3. Thema (60% bzw. 59%) sind von Excel aufgrund der Durchschnittswerte mit vielen Nachkommastellen berechnet worden. In der Arbeit (vgl. Abbildung 90) werden die Prozentwerte auf die ausgewiesenen Durchschnittswerte mit einer Nachkommastelle (12,1 bzw. 11,9) bezogen, daraus der Prozentwert berechnet (60,5 bzw. 59,5) und dann aufgerundet auf 61 bzw. 60!

OG61 (OOP-First): Teilergebnisse inkl. Itemschwierigkeiten

OG61	1.1.1	1.1.2	1.1.3	1.1.4	1.1.5	1.1.6	1.1.7	1.1.8	1.1.9	1.1.10	1.2.1	1.2.2	Summe
Max. Punkte	1	1	1	1	1	1	1	1	1	1	4	6	20
6101	1	0	1	1	1	1	1	1	1	0	3	0	11
6102	1	0	1	1	1	0	1	1	0	1	2	0	9
6103	1	0	1	1	1	1	1	1	0	0	2	0	9
6104	1	0	1	1	1	0	1	0	0	1	1	0	7
6105	1	1	1	1	1	1	1	0	1	0	4	2	14
6106	1	0	1	1	1	1	1	1	1	1	2	0	11
6107	1	0	1	1	1	1	1	1	0	1	3	2	13
6108	1	1	1	1	1	1	1	1	0	0	3	2	13
6109	1	1	1	1	1	1	1	0	1	1	4	4	17
6110	1	1	1	1	1	1	0	0	0	0	3	0	9
6111													
6112	1	1	1	1	1	1	1	1	0	0	3	0	11
6113	1	1	1	1	1	1	1	1	1	1	4	6	20
6114	1	0	1	1	1	1	1	1	1	0	4	4	16
6115	1	0	0	1	1	0	1	1	0	0	1	0	6
6116	1	1	1	1	1	0	1	1	1	0	3	0	11
6117													
6118													
6119	1	0	1	1	1	1	1	1	1	1	3	0	12
6120	1	1	1	1	1	1	1	1	1	1	4	0	14
6121	1	0	1	1	1	1	1	0	1	0	2	0	9
6122	1	0	1	1	1	1	0	1	1	1	3	4	15
6123	1	1	1	1	1	1	1	0	1	0	3	2	13
6124	1	1	1	1	1	1	1	1	0	0	4	2	14
6125	1	1	1	1	1	1	1	0	1	0	4	4	16
6126	1	0	1	0	1	0	0	0	0	0	1	0	4
6127	1	1	1	1	1	1	1	1	1	0	3	4	16
Durchschnitt	1,0	0,5	1,0	1,0	1,0	0,8	0,9	0,7	0,6	0,4	2,9	1,5	12,1
Prozentual	100	50	96	96	100	79	88	67	58	38	72	25	60

Pi	1,00	0,50	0,96	0,96	1,00	0,79	0,88	0,67	0,58	0,38	0,72	0,25
----	------	------	------	------	------	------	------	------	------	------	------	------

2.1.1	2.1.2	2.1.3	2.1.4	2.1.5	2.1.6-7	2.1.8-10	2.2	Summe
1	1	1	1	1	2	3	10	20
0,0	1,0	1,0	0,0	1,0	0,0	0,0	9,0	12
0,0	1,0	1,0	1,0	1,0	0,0	3,0	6,0	13
0,0	1,0	0,0	1,0	0,0	0,0	3,0	7,0	12
0,0	1,0	1,0	0,0	1,0	0,0	3,0	0,5	6,5
0,0	1,0	1,0	0,0	1,0	0,0	0,0	8,0	11
0,0	1,0	1,0	1,0	1,0	2,0	3,0	2,0	11
1,0	1,0	1,0	1,0	1,0	2,0	3,0	7,5	17,5
0,0	1,0	1,0	1,0	0,0	0,0	3,0	8,0	14
1,0	1,0	1,0	1,0	1,0	2,0	3,0	9,5	19,5
0,0	1,0	1,0	1,0	0,0	2,0	0,0	0,0	5
0,0	1,0	1,0	1,0	1,0	2,0	3,0	7,0	16
1,0	1,0	1,0	1,0	1,0	2,0	3,0	9,5	19,5
0,0	1,0	1,0	1,0	1,0	2,0	0,0	10,0	16
0,0	1,0	1,0	1,0	1,0	0,0	0,0	0,0	4
0,0	1,0	1,0	1,0	1,0	2,0	0,0	6,0	12
1,0	1,0	1,0	0,0	1,0	2,0	3,0	7,0	16
1,0	1,0	1,0	1,0	1,0	2,0	0,0	9,0	16
0,0	1,0	1,0	0,0	1,0	0,0	0,0	2,5	5,5
1,0	1,0	1,0	1,0	1,0	2,0	3,0	9,0	19
0,0	1,0	1,0	1,0	1,0	2,0	3,0	8,0	17
0,0	1,0	1,0	1,0	0,0	2,0	3,0	5,0	13
0,0	1,0	1,0	1,0	1,0	2,0	0,0	10,0	16
0,0	1,0	1,0	1,0	0,0	0,0	3,0	3,5	9,5
1,0	1,0	1,0	1,0	1,0	2,0	3,0	10,0	20
0,3	1,0	1,0	0,8	0,8	1,3	1,9	6,4	13,4
29	100	96	79	79	63	63	64	67

0,29	1,00	0,96	0,79	0,79	0,63	0,63	0,64
------	------	------	------	------	------	------	------



3.1.1	3.1.2	3.1.3	3.1.4	3.1.5	3.1.6	3.1.7	3.1.8	3.1.9	3.1.10	3.2.1	3.2.2	3.2.3	Summe
1	1	1	1	1	1	1	1	1	1	3	4	3	20
0,0	1,0	1,0	0,0	0,0	1,0	1,0	0,0	1,0	1,0	3,0	1,0	3,0	13
0,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	0,0	1,5	12,5
0,0	1,0	0,0	0,0	0,0	0,0	1,0	1,0	1,0	1,0	2,0	1,0	2,0	10
0,0	1,0	0,0	0,0	1,0	0,0	1,0	0,0	1,0	1,0	1,0	0,0	2,0	8
0,0	0,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	0,5	4,0	2,5	12
1,0	1,0	0,0	1,0	0,0	0,0	1,0	1,0	1,0	1,0	3,0	0,5	2,0	12,5
0,0	1,0	1,0	1,0	0,0	0,0	0,0	1,0	1,0	1,0	3,0	0,0	2,0	11
1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	3,0	16
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	2,0	4,0	3,0	19
0,0	1,0	1,0	0,0	0,0	0,0	1,0	1,0	1,0	0,0	1,0	3,0	1,0	10
0,0	1,0	0,0	0,0	1,0	0,0	0,0	0,0	1,0	0,0	1,5	0,0	2,0	6,5
1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	0,0	3,5	3,0	15,5
0,0	1,0	0,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	9
0,0	0,0	1,0	0,0	0,0	0,0	1,0	0,0	1,0	1,0	1,0	0,0	1,0	6
1,0	1,0	1,0	1,0	0,0	0,0	1,0	1,0	1,0	0,0	1,0	0,5	0,0	8,5
0,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	3,0	1,0	3,0	15
1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	2,0	4,0	3,0	18
0,0	1,0	0,0	0,0	1,0	0,0	0,0	1,0	1,0	0,0	1,0	0,0	2,0	7
1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,5	3,5	1,5	15,5
1,0	1,0	0,0	1,0	0,0	1,0	1,0	0,0	1,0	0,0	1,5	3,5	3,0	14
0,0	1,0	1,0	1,0	1,0	0,0	0,0	1,0	1,0	0,0	1,5	1,0	1,5	10
0,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	2,5	4,0	3,0	17,5
0,0	0,0	0,0	0,0	0,0	0,0	1,0	1,0	0,0	1,0	0,0	0,5	1,5	5
1,0	1,0	0,0	1,0	1,0	0,0	0,0	1,0	1,0	0,0	1,0	4,0	3,0	14
0,4	0,9	0,5	0,5	0,5	0,5	0,8	0,8	1,0	0,7	1,6	1,8	2,1	11,9
38	88	46	50	54	46	79	79	96	67	53	45	70	59
0,38	0,88	0,46	0,50	0,54	0,46	0,79	0,79	0,96	0,67	0,53	0,45	0,70	

4.1.1	4.1.2	4.1.3	4.1.4	4.1.5	4.1.6	4.1.7	4.1.8	4.1.9	4.1.10	4.2.1	4.2.2	4.2.3	4.2.4	Summe
1	1	1	1	1	1	1	1	1	1	3	3	2	2	20
1,0	1,0	1,0	0,0	0,0	1,0	0,0	0,0	1,0	0,0	2,0	3,0	2,0	2,0	14
1,0	1,0	1,0	0,0	0,0	1,0	0,0	0,0	1,0	0,0	3,0	3,0	2,0	1,5	14,5
1,0	1,0	0,0	0,0	0,0	0,0	0,0	1,0	1,0	0,0	1,0	0,0	0,0	1,5	6,5
1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	2,0	3,0	2,0	2,0	17
1,0	1,0	1,0	0,0	1,0	1,0	0,0	1,0	0,0	1,0	3,0	3,0	2,0	1,5	16,5
1,0	1,0	0,0	0,0	1,0	1,0	1,0	0,0	1,0	1,0	2,0	3,0	2,0	1,5	15,5
1,0	1,0	0,0	0,0	0,0	1,0	0,0	1,0	1,0	1,0	3,0	3,0	2,0	2,0	16
0,0	1,0	0,0	1,0	1,0	1,0	0,0	0,0	1,0	1,0	3,0	3,0	2,0	0,5	14,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	3,0	3,0	1,5	0,5	17
1,0	1,0	0,0	0,0	1,0	1,0	0,0	0,0	1,0	0,0	2,0	3,0	2,0	1,5	13,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	2,0	0,0	0,0	12
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	3,0	2,0	1,5	19,5
0,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	3,0	3,0	2,0	1,5	16,5
1,0	0,0	0,0	0,0	1,0	0,0	0,0	1,0	0,0	0,0	1,0	3,0	1,5	0,5	9
1,0	0,0	1,0	0,0	0,0	1,0	0,0	0,0	1,0	0,0	1,0	0,0	1,5	1,0	7,5
1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	3,0	3,0	2,0	2,0	18
1,0	1,0	1,0	0,0	1,0	1,0	0,0	1,0	1,0	1,0	3,0	3,0	2,0	2,0	18
1,0	1,0	1,0	0,0	1,0	0,0	0,0	0,0	0,0	1,0	0,5	0,5	0,0	0,0	6
1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	0,0	2,5	2,5	2,0	2,0	17
1,0	1,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	2,0	2,5	2,0	2,0	15,5
1,0	1,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	2,0	1,0	0,5	11,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	3,0	2,0	2,0	20
1,0	1,0	1,0	0,0	1,0	1,0	0,0	1,0	0,0	1,0	2,5	3,0	1,0	0,0	13,5
1,0	0,0	1,0	0,0	1,0	1,0	1,0	0,0	0,0	1,0	3,0	3,0	2,0	2,0	16
0,9	0,9	0,7	0,3	0,8	0,9	0,5	0,7	0,8	0,4	2,2	2,5	1,6	1,3	14,4
92	88	67	25	79	88	46	67	79	42	74	84	80	66	72
0,92	0,88	0,67	0,25	0,79	0,88	0,46	0,67	0,79	0,42	0,74	0,84	0,80	0,66	

5.1.1	5.1.2	5.1.3	5.1.4	5.1.5	5.1.6	5.1.7	5.1.8	5.1.9	5.1.10	5.2.1	5.2.2	Summe
1	1	1	1	1	1	1	1	1	1	7	3	20
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	6,5	3,0	19,5
0,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	5,5	2,0	14,5
0,0	1,0	0,0	1,0	0,0	0,0	1,0	1,0	0,0	1,0	4,0	0,5	9,5
1,0	1,0	0,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	4,0	2,0	14
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	6,5	0,0	16,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	2,5	19,5
1,0	1,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	3,0	18
1,0	0,0	1,0	1,0	1,0	1,0	0,0	0,0	1,0	0,0	5,5	1,5	13
1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	6,0	3,0	18
0,0	0,0	1,0	1,0	1,0	0,0	1,0	1,0	0,0	1,0	5,5	0,0	11,5
1,0	1,0	0,0	1,0	1,0	0,0	1,0	0,0	1,0	1,0	3,5	2,0	12,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	6,5	2,0	18,5
0,0	0,0	0,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	4,0	2,0	12
0,0	1,0	1,0	1,0	0,0	0,0	0,0	0,0	1,0	1,0	3,5	0,0	8,5
0,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	0,0	0,5	6,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	6,5	2,5	19
1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	6,5	2,0	17,5
1,0	1,0	0,0	1,0	0,0	0,0	1,0	0,0	0,0	1,0	2,5	0,0	7,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	6,0	2,0	17
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	0,0	6,0	1,0	15
1,0	1,0	0,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	4,0	0,0	12
0,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	0,0	1,0	6,5	1,5	15
0,0	1,0	1,0	0,0	1,0	0,0	1,0	1,0	1,0	0,0	3,0	1,0	10
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	3,0	20
0,7	0,8	0,6	0,9	0,9	0,5	0,9	0,9	0,7	0,9	5,1	1,5	14,4
67	79	63	92	88	50	88	88	71	88	73	51	72
0,67	0,79	0,63	0,92	0,88	0,50	0,88	0,88	0,71	0,88	0,73	0,51	

6.1	6.2	6.3	Summe	Summe	Prozentual
2	3	5	10		
0,5	0,0	1,5	2	71,5	65,0
2,0	0,0	1,0	3	66,5	60,5
2,0	0,0	2,0	4	51,0	46,4
2,0	0,0	2,0	4	56,5	51,4
2,0	2,0	2,5	6,5	76,5	69,5
2,0	2,0	0,0	4	73,5	66,8
2,0	0,0	1,0	3	78,5	71,4
2,0	0,0	3,0	5	75,5	68,6
2,0	2,0	4,5	8,5	99,0	90,0
0,5	2,0	3,0	5,5	54,5	49,5
2,0	2,0	1,0	5	63,0	57,3
2,0	3,0	4,0	9	102,0	92,7
2,0	1,0	3,0	6	75,5	68,6
2,0	0,0	0,0	2	35,5	32,3
2,0	2,0	1,5	5,5	51,0	46,4
2,0	2,0	2,0	6	86,0	78,2
2,0	2,0	4,0	8	91,5	83,2
0,5	2,0	1,0	3,5	38,5	35,0
2,0	2,0	2,0	6	89,5	81,4
2,0	2,0	1,0	5	79,5	72,3
0,5	0,0	2,0	2,5	63,0	57,3
2,0	2,0	3,0	7	91,5	83,2
2,0	0,0	1,5	3,5	45,5	41,4
2,0	2,0	2,0	6	92,0	83,6
1,8	1,3	2,0	5,0	71,1	64,7
88	42	40	50	65	65

OG61
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
Durchschnitt
Prozentual

0,88	0,42	0,40	Durchschnitt	0,69	Pi
------	------	------	--------------	------	----

OG62 (OOP-Later): Gesamtergebnis

OG62	1.Thema	2. Thema	3. Thema	4. Thema	5. Thema	6. Thema	Summe	Prozentual
6201	11,0	19,0	20,0	20,0	20,0	8,0	98,0	89,1
6202	15,0	19,5	17,5	17,0	19,5	9,0	97,5	88,6
6203	13,0	16,5	14,5	16,0	17,5	7,0	84,5	76,8
6204	9,0	4,0	6,5	11,5	13,5	5,0	49,5	45,0
6205	8,0	7,0	3,0	6,5	8,5	4,0	37,0	33,6
6206	13,0	11,5	13,5	17,5	17,5	6,0	79,0	71,8
6207	17,0	15,0	18,0	19,5	17,0	8,0	94,5	85,9
6208	7,0	3,0	10,0	16,5	18,0	6,0	60,5	55,0
6209	9,0	10,0	15,0	17,0	15,5	6,0	72,5	65,9
6210	17,0	19,5	20,0	20,0	20,0	10,0	106,5	96,8
6211								
6212	10,0	15,0	9,5	15,5	17,5	8,0	75,5	68,6
6213	11,0	10,5	9,0	17,0	18,0	6,0	71,5	65,0
6214	12,0	13,5	11,5	17,0	15,5	5,0	74,5	67,7
6215	18,0	19,5	20,0	19,0	20,0	8,0	104,5	95,0
6216	8,0	13,0	8,5	15,5	20,0	7,5	72,5	65,9
6217	14,0	18,0	18,5	18,5	19,0	8,0	96,0	87,3
6218	11,5	16,0	18,5	20,0	19,0	8,0	93,0	84,5
6219	6,0	10,5	12,0	15,0	17,0	6,0	66,5	60,5
6220	13,0	16,0	18,0	20,0	20,0	9,0	96,0	87,3
6221	11,0	8,5	8,0	11,5	16,0	6,0	61,0	55,5
6222	14,0	17,0	11,5	18,5	20,0	4,5	85,5	77,7
6223	12,0	7,0	9,5	7,5	10,5	5,5	52,0	47,3
6224								
6225	8,0	8,0	7,5	8,0	8,5	4,0	44,0	40,0
6226	11,0	7,0	9,0	15,5	16,5	5,0	64,0	58,2
6227	12,0	9,5	6,0	14,0	18,0	8,5	68,0	61,8
6228	16,0	17,5	14,0	13,0	19,0	7,0	86,5	78,6
<b>Durchschnitt</b>	<b>11,8</b>	<b>12,8</b>	<b>12,7</b>	<b>15,7</b>	<b>17,0</b>	<b>6,7</b>	<b>76,6</b>	<b>69,6</b>
<b>Prozentual</b>	<b>59</b>	<b>64</b>	<b>63</b>	<b>78</b>	<b>85</b>	<b>67</b>	<b>70</b>	<b>70</b>

Die Prozentzahlen vom 3. und 4. Thema (63% bzw. 78%) sind von Excel aufgrund der Durchschnittswerte mit vielen Nachkommastellen berechnet worden. In der Arbeit (vgl. Abbildung 90) werden die Prozentwerte auf die ausgewiesenen Durchschnittswerte mit einer Nachkommastelle (12,7 bzw. 15,7) bezogen, daraus der Prozentwert berechnet (63,5 bzw. 78,5) und dann aufgerundet auf 64 bzw. 79!

OG62 (OOP-Later): Teilergebnisse inkl. Itemschwierigkeiten

OG62	1.1.1	1.1.2	1.1.3	1.1.4	1.1.5	1.1.6	1.1.7	1.1.8	1.1.9	1.1.10	1.2.1	1.2.2	Summe
max.Punkte	1	1	1	1	1	1	1	1	1	1	4	6	20
6201	1	0	1	1	1	1	1	1	0	1	3	0	11
6202	1	0	1	1	1	1	1	1	1	1	4	2	15
6203	1	0	0	1	1	1	0	1	1	1	2	4	13
6204	1	1	1	1	1	0	1	1	0	0	2	0	9
6205	0	1	1	0	1	0	0	1	0	0	2	2	8
6206	1	1	0	0	1	0	1	1	0	1	3	4	13
6207	1	1	1	0	1	1	1	1	1	1	4	4	17
6208	1	0	1	0	1	0	1	1	0	0	2	0	7
6209	1	0	1	1	1	0	0	1	0	1	1	2	9
6210	1	1	0	1	1	1	1	1	1	1	4	4	17
6211													
6212	1	1	0	1	1	0	0	1	1	1	3	0	10
6213	0	1	1	1	1	1	0	1	1	1	3	0	11
6214	1	1	0	0	1	0	0	1	0	1	3	4	12
6215	1	1	1	1	1	1	1	1	0	1	3	6	18
6216	1	0	1	1	0	0	0	1	0	1	3	0	8
6217	1	1	1	1	1	1	0	1	0	1	4	2	14
6218	1	1	0	1	1	0	0	1	0	1	3,5	2	11,5
6219	0	0	0	1	1	0	1	1	0	0	2	0	6
6220	1	1	1	0	1	1	0	1	1	1	3	2	13
6221	1	0	0	0	1	1	1	1	0	1	3	2	11
6222	1	1	1	0	1	1	1	1	0	1	2	4	14
6223	1	1	1	1	1	1	1	1	0	1	3	0	12
6224													
6225	1	0	0	0	1	0	0	0	0	1	3	2	8
6226	1	0	1	0	1	1	0	0	0	1	4	2	11
6227	1	1	1	1	1	1	0	1	0	1	4	0	12
6228	1	0	1	1	1	0	0	1	0	1	4	6	16
Durchschnitt	0,9	0,6	0,7	0,6	1,0	0,5	0,5	0,9	0,3	0,8	3,0	2,1	11,8
Prozentual	88	58	65	62	96	54	46	92	27	85	75	35	59

Pi	0,88	0,58	0,65	0,62	0,96	0,54	0,46	0,92	0,27	0,85	0,75	0,35
----	------	------	------	------	------	------	------	------	------	------	------	------

2.1.1	2.1.2	2.1.3	2.1.4	2.1.5	2.1.6-7	2.1.8-10	2.2	Summe
1	1	1	1	1	2	3	10	20
1,0	1,0	1,0	1,0	1,0	2,0	3,0	9,0	19
1,0	1,0	1,0	1,0	1,0	2,0	3,0	9,5	19,5
1,0	1,0	1,0	1,0	1,0	2,0	3,0	6,5	16,5
1,0	1,0	1,0	1,0	0,0	0,0	0,0	0,0	4
1,0	1,0	1,0	0,0	1,0	0,0	3,0	0,0	7
1,0	1,0	1,0	0,0	1,0	2,0	0,0	5,5	11,5
1,0	1,0	1,0	1,0	0,0	2,0	0,0	9,0	15
1,0	0,0	1,0	0,0	1,0	0,0	0,0	0,0	3
1,0	1,0	1,0	1,0	0,0	0,0	0,0	6,0	10
1,0	1,0	1,0	1,0	1,0	2,0	3,0	9,5	19,5
1,0	1,0	1,0	1,0	0,0	0,0	3,0	8,0	15
1,0	1,0	1,0	1,0	1,0	0,0	0,0	5,5	10,5
1,0	0,0	1,0	1,0	0,0	2,0	0,0	8,5	13,5
1,0	1,0	1,0	1,0	1,0	2,0	3,0	9,5	19,5
1,0	1,0	1,0	1,0	1,0	0,0	0,0	8,0	13
1,0	1,0	1,0	1,0	0,0	2,0	3,0	9,0	18
1,0	1,0	1,0	1,0	1,0	0,0	3,0	8,0	16
1,0	1,0	1,0	1,0	1,0	0,0	0,0	5,5	10,5
1,0	1,0	1,0	1,0	0,0	2,0	0,0	10,0	16
1,0	1,0	0,0	1,0	1,0	2,0	0,0	2,5	8,5
1,0	1,0	1,0	1,0	0,0	2,0	3,0	8,0	17
1,0	1,0	1,0	1,0	1,0	2,0	0,0	0,0	7
0,0	1,0	1,0	1,0	1,0	0,0	3,0	1,0	8
0,0	1,0	0,0	1,0	1,0	0,0	0,0	4,0	7
1,0	1,0	1,0	0,0	1,0	0,0	0,0	5,5	9,5
1,0	1,0	1,0	1,0	1,0	2,0	3,0	7,5	17,5
0,9	0,9	0,9	0,8	0,7	1,1	1,4	6,0	12,8
92	92	92	85	69	54	46	60	64
0,92	0,92	0,92	0,85	0,69	0,54	0,46	0,60	

3.1.1	3.1.2	3.1.3	3.1.4	3.1.5	3.1.6	3.1.7	3.1.8	3.1.9	3.1.10	3.2.1	3.2.2	3.2.3	Summe
1	1	1	1	1	1	1	1	1	1	3	4	3	20
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	4,0	3,0	20
1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	2,0	4,0	2,5	17,5
0,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,5	3,0	2,0	14,5
1,0	1,0	0,0	1,0	1,0	0,0	0,0	1,0	0,0	0,0	0,0	0,5	1,0	6,5
0,0	1,0	0,0	1,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,0	3
1,0	1,0	0,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	2,5	2,0	13,5
1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	2,0	4,0	3,0	18
1,0	1,0	0,0	0,0	0,0	0,0	1,0	1,0	1,0	1,0	0,0	4,0	0,0	10
1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	4,0	2,0	15
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	4,0	3,0	20
0,0	1,0	0,0	1,0	1,0	0,0	0,0	1,0	1,0	1,0	2,5	1,0	0,0	9,5
0,0	1,0	1,0	0,0	1,0	0,0	1,0	1,0	0,0	1,0	2,0	1,0	0,0	9
1,0	1,0	1,0	1,0	0,0	0,0	1,0	1,0	1,0	1,0	1,5	1,0	1,0	11,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	4,0	3,0	20
0,0	1,0	0,0	0,0	0,0	0,0	0,0	1,0	1,0	1,0	1,5	1,0	2,0	8,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	4,0	1,5	18,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	3,0	3,5	3,0	18,5
1,0	1,0	1,0	0,0	1,0	0,0	1,0	1,0	1,0	0,0	2,0	1,0	2,0	12
1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	3,5	2,5	18
1,0	1,0	0,0	1,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	0,0	8
0,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,5	11,5
0,0	1,0	0,0	1,0	0,0	0,0	0,0	1,0	1,0	1,0	2,0	1,0	1,5	9,5
0,0	0,0	0,0	0,0	1,0	0,0	1,0	1,0	1,0	0,0	2,5	0,0	1,0	7,5
1,0	1,0	0,0	1,0	1,0	0,0	0,0	1,0	1,0	1,0	1,0	0,0	1,0	9
0,0	0,0	1,0	0,0	1,0	0,0	0,0	1,0	1,0	0,0	2,0	0,0	0,0	6
1,0	1,0	0,0	1,0	1,0	1,0	1,0	0,0	0,0	0,0	2,0	3,5	2,5	14
0,7	0,9	0,5	0,7	0,8	0,4	0,7	1,0	0,8	0,7	1,8	2,1	1,6	12,7
65	92	54	65	77	42	73	96	85	69	58	53	53	63
0,65	0,92	0,54	0,65	0,77	0,42	0,73	0,96	0,85	0,69	0,58	0,53	0,53	

4.1.1	4.1.2	4.1.3	4.1.4	4.1.5	4.1.6	4.1.7	4.1.8	4.1.9	4.1.10	4.2.1	4.2.2	4.2.3	4.2.4	Summe
1	1	1	1	1	1	1	1	1	1	3	3	2	2	20
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	3,0	2,0	2,0	20
0,0	1,0	1,0	0,0	1,0	1,0	1,0	0,0	1,0	1,0	3,0	3,0	2,0	2,0	17
0,0	1,0	1,0	0,0	1,0	1,0	0,0	1,0	1,0	0,0	3,0	3,0	2,0	2,0	16
1,0	1,0	1,0	0,0	1,0	1,0	0,0	0,0	1,0	1,0	1,0	0,5	2,0	1,0	11,5
1,0	0,0	1,0	1,0	1,0	0,0	1,0	1,0	0,0	0,0	0,0	0,0	0,0	0,5	6,5
1,0	1,0	1,0	1,0	1,0	1,0	0,0	0,0	1,0	1,0	2,5	3,0	2,0	2,0	17,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	2,5	2,0	2,0	19,5
1,0	1,0	1,0	0,0	1,0	1,0	0,0	0,0	1,0	1,0	3,0	3,0	2,0	1,5	16,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	0,0	1,0	3,0	3,0	1,0	2,0	17
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	3,0	2,0	2,0	20
1,0	1,0	0,0	1,0	1,0	1,0	0,0	0,0	0,0	1,0	3,0	3,0	1,5	2,0	15,5
1,0	0,0	1,0	1,0	0,0	1,0	1,0	0,0	1,0	1,0	3,0	3,0	2,0	2,0	17
1,0	0,0	1,0	1,0	1,0	1,0	1,0	0,0	0,0	1,0	3,0	3,0	2,0	2,0	17
1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	3,0	3,0	2,0	2,0	19
1,0	1,0	0,0	0,0	0,0	1,0	1,0	0,0	1,0	1,0	2,5	3,0	2,0	2,0	15,5
1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	3,0	2,0	1,5	18,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	3,0	2,0	2,0	20
1,0	0,0	1,0	0,0	0,0	1,0	0,0	0,0	1,0	1,0	3,0	3,0	2,0	2,0	15
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	3,0	3,0	2,0	2,0	20
1,0	1,0	0,0	0,0	1,0	1,0	0,0	0,0	1,0	1,0	1,5	0,5	2,0	1,5	11,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	3,0	3,0	2,0	1,5	18,5
0,0	1,0	1,0	1,0	0,0	0,0	0,0	0,0	1,0	0,0	0,5	0,5	1,0	1,5	7,5
1,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	1,0	1,5	1,0	2,0	0,5	8
0,0	1,0	1,0	0,0	1,0	1,0	0,0	0,0	1,0	1,0	2,5	3,0	2,0	2,0	15,5
1,0	1,0	1,0	1,0	0,0	0,0	0,0	0,0	0,0	1,0	3,0	3,0	1,5	1,5	14
1,0	1,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	2,5	0,0	1,5	13
0,8	0,8	0,8	0,6	0,8	0,8	0,6	0,4	0,8	0,9	2,5	2,5	1,7	1,7	15,7
85	81	81	58	77	85	58	38	77	88	82	83	87	86	78
0,85	0,81	0,81	0,58	0,77	0,85	0,58	0,38	0,77	0,88	0,82	0,83	0,87	0,86	

5.1.1	5.1.2	5.1.3	5.1.4	5.1.5	5.1.6	5.1.7	5.1.8	5.1.9	5.1.10	5.2.1	5.2.2	Summe
1	1	1	1	1	1	1	1	1	1	7	3	20
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	3,0	20
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	2,5	19,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	7,0	1,5	17,5
1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	3,5	1,0	13,5
1,0	1,0	0,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	0,5	1,0	8,5
1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	7,0	1,5	17,5
1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	0,0	1,0	7,0	2,0	17
1,0	1,0	1,0	1,0	1,0	0,0	0,0	1,0	1,0	1,0	7,0	3,0	18
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	0,0	1,0	5,0	15,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	3,0	20
1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	6,0	2,5	17,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	5,5	2,5	18
1,0	1,0	0,0	0,0	1,0	1,0	1,0	1,0	1,0	1,0	5,0	2,5	15,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	3,0	20
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	3,0	20
1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	7,0	3,0	19
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	7,0	3,0	19
1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	7,0	1,0	17
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	3,0	20
1,0	1,0	1,0	0,0	1,0	1,0	1,0	1,0	0,0	0,0	6,5	2,5	16
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	7,0	3,0	20
1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	0,0	1,0	2,5	0,0	10,5
1,0	1,0	0,0	0,0	0,0	0,0	1,0	1,0	1,0	1,0	2,0	0,5	8,5
1,0	1,0	1,0	1,0	1,0	1,0	0,0	1,0	1,0	1,0	6,5	1,0	16,5
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,0	0,0	7,0	3,0	18
1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	6,0	3,0	19
1,0	1,0	0,9	0,8	1,0	0,7	0,9	0,8	0,7	1,0	5,9	2,2	17,0
100	100	88	85	96	69	92	85	73	96	85	74	85
1,00	1,00	0,88	0,85	0,96	0,69	0,92	0,85	0,73	0,96	0,85	0,74	

6.1	6.2	6.3	Summe	Summe	Prozentual
2	3	5	10		
2,0	2,0	4,0	8	98,0	89,1
2,0	3,0	4,0	9	97,5	88,6
2,0	1,0	4,0	7	84,5	76,8
2,0	0,0	3,0	5	49,5	45,0
2,0	0,0	2,0	4	37,0	33,6
2,0	0,0	4,0	6	79,0	71,8
2,0	2,0	4,0	8	94,5	85,9
2,0	0,0	4,0	6	60,5	55,0
2,0	3,0	1,0	6	72,5	65,9
2,0	3,0	5,0	10	106,5	96,8
2,0	2,0	4,0	8	75,5	68,6
2,0	1,0	3,0	6	71,5	65,0
2,0	0,0	3,0	5	74,5	67,7
2,0	2,0	4,0	8	104,5	95,0
2,0	2,0	3,5	7,5	72,5	65,9
2,0	2,0	4,0	8	96,0	87,3
2,0	2,0	4,0	8	93,0	84,5
2,0	1,0	3,0	6	66,5	60,5
2,0	2,0	5,0	9	96,0	87,3
2,0	1,0	3,0	6	61,0	55,5
2,0	1,0	1,5	4,5	85,5	77,7
2,0	0,0	3,5	5,5	52,0	47,3
2,0	0,0	2,0	4	44,0	40,0
2,0	2,0	1,0	5	64,0	58,2
2,0	3,0	3,5	8,5	68,0	61,8
2,0	2,0	3,0	7	86,5	78,6
2,0	1,4	3,3	6,7	76,6	69,6
100	47	66	67	70	70

OG62
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
Durchschnitt
Prozentual

1,00	0,47	0,66	Durchschnitt	0,74	Pi
------	------	------	--------------	------	----

## H) Promotionsvorschlag (Februar 2008)



### Vorstellung: Promotionsvorhaben



Albrecht Ehler, 15741 Bestensee, Böcklinstr. 21a, ehler@oszimt.de

Datum: 04.02.2008

#### Empirische Studie:

Emotional-motivationales Erleben des Unterrichtsgeschehens  
und (signifikante) Unterschiede im Lernerfolg  
im Informatik-Anfängerunterricht  
in Abhängigkeit der zeitlichen Reihenfolge  
der einzelnen Themen  
(OOP-First bzw. OOP-Later)

**Zusammenfassung.** Unter den vielen fachdidaktischen Vorschlägen zum Unterrichten der Objektorientierten Programmierung (OOP) werden bei dieser empirischen Untersuchung zwei Wege verglichen, der sog. OOP-First-Einstieg mit dem OOP-Later-Einstieg. Dabei werden zwei Schulklassen ein Jahr lang in zweierlei Hinsicht untersucht:

1. Wie ist das subjektive Erleben der Schüler bezogen auf den Informatik-Unterricht?
2. Welche objektiven Unterschiede sind bei den Lernerfolgen festzustellen?

#### 1. Vorstellung

**Meine Person.** Ich bin 49 Jahre alt, Diplom-Ingenieur Elektrotechnik und unterrichte als Lehrer (Studiendirektor) seit fast 20 Jahren die Fächer Elektrotechnik, Sozialkunde und Informatik. Die letzten 6 Jahre bin ich ausschließlich in den Fächern Informatik in der gymnasialen Oberstufe und Anwendungssysteme (Ausbildungsberuf Fachinformatiker) in der Berufsschule eingesetzt. An meiner Schule leite ich den Fachbereich Informationstechnik.

**Die Schule.** Das Oberstufenzentrum Informations- und Medizintechnik ist die größte IT-Schule Berlins ([www.oszimt.de](http://www.oszimt.de)). Sie beherbergt unterschiedliche Bildungsgänge unter einem Dach, wie z.B. die Berufsschule und die gymnasiale Oberstufe.

**Der Bildungsgang.** Die gymnasiale Oberstufe beginnt mit der 11. Klasse (Einführungsphase). Dort haben die Schüler ein 6stündiges Fach Technik, welches sich zu gleichen Teilen aus der Elektrotechnik und der Informatik zusammensetzt.

**Die Schüler.** Untersucht werden sollen zwei Parallelklassen der gymnasialen Oberstufe (Einführungsphase). Die Schüler kommen überwiegend von Realschulen und sind eher leistungsschwach.

#### 2. Informatik-Anfangsunterricht

**Vorgehensweisen** [Eh07]. Man findet in deutschen Schulen und Hochschulen oft zwei unterschiedliche Strategien um objektorientierte Programmierung im Kontext von „Programmierung für Anfänger“ zu unterrichten.

Bei der einen Strategie wird gleich mit den Begrifflichkeiten der OO, OOA, OOD, OOP und/oder OOM begonnen. Dabei ergeben sich für den Lehrer oft Fragen wie diese:

- Was sind die besten Themen der ersten Stunden? [MPW06]
- Welches sind das beste Unterrichtskonzept und das beste IDE-Tool? [Sch03]
- Welcher OOP-Einstieg ist für die Schüler am leichtesten? [BK06]
- Was sind die besten OOP-Übungen für Anfänger? [Ta06]

Bei der anderen Strategie werden erst prozedurale Themen und dann in einem weiteren Lernabschnitt OO-Themen unterrichtet. Dabei ergeben sich für den Lehrer oft Fragen wie diese:

- Wann bietet sich der Umstieg von den prozeduralen auf die OO-Themen an? [Nø06]
- Welche Programmiersprache bietet die Möglichkeit, sowohl objektorientiert als auch nicht objektorientiert (prozedural) zu programmieren? [Th05]

**Themen.** Obwohl beide Wege auf den ersten Blick sehr unterschiedlich sind, enthalten sie doch im Endeffekt die gleichen Themen, z.B. die Implementierung von Klassen und die Erzeugung von Objekten, genauso wie die Themen Daten- und Steuerstrukturen.

Nur die Sichtweise scheint eine andere zu sein:

1. Das OO-Paradigma steht von Anfang an im Mittelpunkt des Anfänger-Programmier-Unterrichts, die Themenfolge ist oft wie folgt:

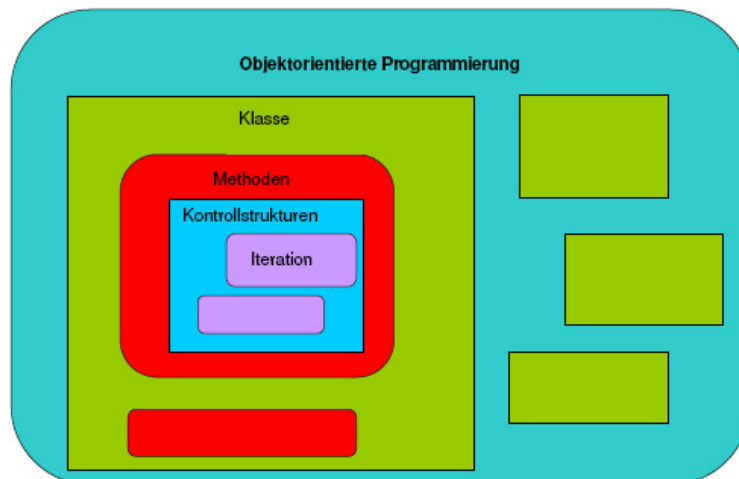
- Klassen und Objekte
- Attribute und Methoden
- Vererbung
- Assoziation

2. Das objektorientierte Paradigma baut auf das prozedurale Paradigma auf, die Themenfolge ist oft wie folgt:

- Variablen und Konstanten, einfache Datentypen
- Kontrollstrukturen: Folgen, Schleifen, Selektionen
- Prozeduren
- Klassen und Objekte, ...

Die Themen, die auf den ersten Blick so unterschiedlich wirken, ergänzen sich in Wirklichkeit, nur die zeitliche Abfolge ist eine andere!

Bei der ersten Sichtweise, wenn der Objektbegriff und der Klassenbegriff geklärt sind, kommt irgendwann der Zeitpunkt wo im Unterricht z.B. auf Methoden eingegangen wird. Wie sollen aber Methoden implementieren ohne Kontrollstrukturen (z.B. die Iteration):

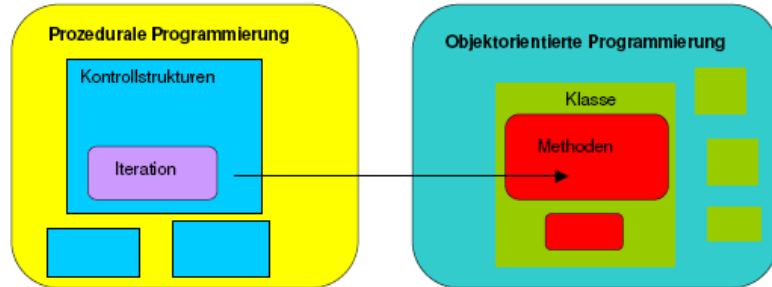


**Darstellung1: Thema Iteration beim OOP-First-Ansatz**

Ähnlich ist es beim Thema Attribute, es impliziert das Thema fundamentale Datentypen!



Bei einer Koexistenz zwischen dem prozeduralen Paradigma und dem objektorientierten Paradigma wird in der objektorientierten Programmierung auf vorhandenes Wissen aus der prozeduralen Programmierung zurück gegriffen:



Darstellung2: Thema Iteration beim OOP-Later-Ansatz

Hierbei gibt es für den Unterrichtenden Freiheitsgrade: Wann ist der Punkt gekommen um von der prozeduralen Sichtweise auf die objektorientierte Sichtweise umzusteigen?

**Begriffsbestimmung.** Ich möchte im folgenden den Begriff **OOP-First** verwenden, wenn im Anfangsunterricht gleich mit dem Begriff Objekt bzw. Klasse eingestiegen wird. Dagegen setze ich den Begriff **OOP-Later**, wenn ein Teil der prozeduralen Themen vor dem Einstieg in die OO-Themen erfolgt. Das „OOP“ statt dem „OO“ bzw. „O“ verwende ich deshalb, weil unsere Schüler mehr als 50% der Zeit mit der (Java-) Programmierung beschäftigt sind.

**Probleme.** Unsere Schüler tun sich schwer im Informatik-Anfangsunterricht, den wir auch nach dem Motto „Objects First“ bzw. „OOP-First“ gestalten. Aber wie sagt Koll. Spolwig [SP05] so schön: Probleme im OOP-Anfangsunterricht lassen eher Mängel in der didaktisch-methodischen Aufbereitung vermuten.

Also habe ich im Jahr 2005 drei Fortbildungen zur „OOP im Anfängerunterricht“ besucht:

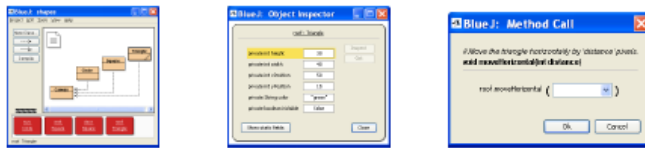
- Niki-Einstieg bei Koll. Dietz / Dr. Kokavec
- Fujaba-Einstieg (OOM) bei Prof. Schulte / Diethelm
- BlueJ-Einstieg bei Koll. Dietz / Punkenburg

Ich habe drei vollständig verschiedene Ansätze kennen gelernt, die aber leider m. E. nicht überzeugend waren im Hinblick auf die Probleme meiner Schüler!

### 3. Thesen

**Frage.** Ich möchte Koll. Spolwig entgegenen: „Haben wir vielleicht ein allgemeines Vermittlungsproblem mit der „OOP-First“-Strategie im Anfangsunterricht, welches sich auch durch noch so viele Tools höchstens abfedern, aber leider nicht beseitigen lässt?“

**Beispiel (Hinführung zur These1).** Als Beispiel soll hier der oft genutzte Einstieg in die OOP mit dem BlueJ-Tool aufgeführt werden:



Darstellung3: BlueJ-Tool [BK06]

Dieses erste BlueJ-Beispiel enthält ein Menge Fachtermini: Klassen, Objekte, Zugriffsspezifizierer, Methoden, Parameter und Attribute.

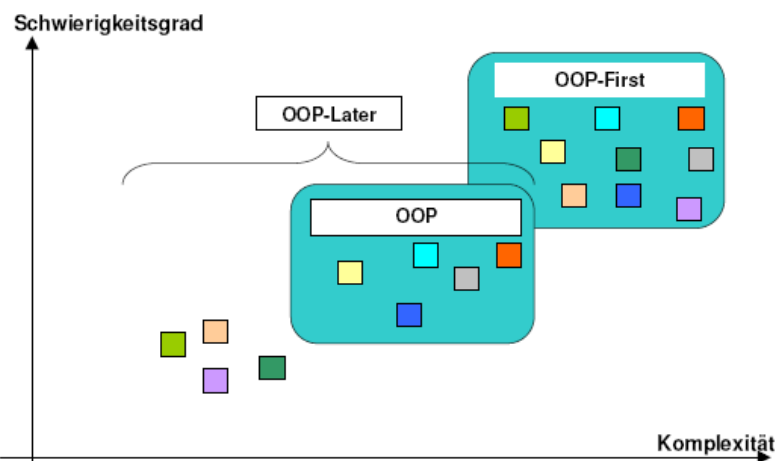
Wenn diese Entwicklungsumgebung für den OOP-First-Einstieg genutzt wird, wissen die Schüler noch nichts über Datentypen, Variablen, Konstanten, Schleifen und Verzweigungen. Der Lehrer wird dieses Beispiel anders präsentieren als ein Lehrer, der über einen Objects-Later-Weg (z.B. mit einem regulären textorientierten Java-Editor) bei dem Übergang zur objektorientierten Programmierung dieses Beispiel einsetzt und schon viele Begrifflichkeiten bei den Schülern voraussetzen kann.

Und die Schüler mit ihrem unterschiedlichen Vorwissen werden m. E. das gleiche Thema mit dem gleichen Tool unterschiedlich erleben (z.B. die Schwere des Themas).

**These1.** Das Thema OOP in der ersten „Anfangsstunde“ wird von den Schülern schwerer erlebt, als wenn es erst nach einer „Einführung in die prozedurale Programmierung“ erfolgt.

**Hinführung zur These2.** Die von der allgemeinen Lerntheorie herstammenden didaktischen Prinzipien „Vom Einfachen zum Schweren“, „Neue Informationen mit altem Wissen in Verbindung setzen“ und „Nicht zu viele neue Informationen auf einmal“ sollte auch bei der „OOP im Anfängerunterricht“ eingehalten werden. Dies gilt umso mehr, je leistungsschwächer die Schüler sind! Es spricht somit m. E. einiges dafür, vor der Einführung in die OOP erst einmal grundlegendes Wissen wie Datentypen, Selektionen und Iterationen zu vermitteln, da dann der Unterricht vom Leichten zum Schweren führt, auf Bekanntes aufgebaut wird und nicht zu viele neue Dinge auf einmal unterrichtet werden!

**These2.** Ein Unterrichts-Einstieg mit „Programmieren im Kleinen“ ohne OOP-Bezug steigert den „Wohlfühlfaktor“ im Unterricht und somit auch die Motivation bzw. die Erfolgserlebnisse des einzelnen (leistungsschwachen) Schülers und verbessert somit auch die Lernergebnisse! Parallel dazu wird der „Schwierigkeitsgrad“ einiger Themen verringert.



Darstellung4: Veranschaulichung der Thesen

**Darstellung.** Sie soll die Thesen veranschaulichen: Der „OOP-First“-Einstieg mit seinen 9 Themen befindet sich auf einem relativ hohen Schwierigkeits- und Komplexitätsgrad. Lagert man nun 4 prozedurale Themen aus und zieht sie zeitlich im Unterricht an den Anfang, dann gestaltet sich der spätere OOP-Einstieg für die Schüler leichter.

„**Antithese**“. Durch die OO-Begrifflichkeiten von der ersten Stunde an wird das OO-Paradigma viel besser von den Schülern verinnerlicht.

Durch die Vielzahl der Begrifflichkeiten (inkl. der prozeduralen Begrifflichkeiten) werden diese unterschiedlichen Themen viel öfters (im Sinne eines Spiralmodells) bzw. immer wieder angesprochen und somit besser gelehrt.

„**Synthese**“. Beide Vorgehen haben spezielle Vor- und Nachteile. Diese sollten herausgearbeitet werden!

**Hinführung zur These3.** Wenn man sich den OOP-First- und den OOP-Later-Ansatz im Hinblick auf die einzelnen Themen anguckt, finden sich starke Überschneidungen!

Im Endeffekt lassen sich gleichartige Themen herausarbeiten:

1. Einstieg in die OOP (z.B. mit Hilfe des BlueJ-Tools)
- 2a. Einstieg in eine textorientierte IDE (z.B. der JavaEditor)
- 2b. Variablen, Konstanten, Datentypen
3. Klasse und Objekt
4. Kontrollstruktur: Selektion
5. Methoden
6. Kontrollstruktur: Iteration
7. Komplexere Datentypen: Arrays(Felder) und Strings (Zeichenkette)
8. Vererbung
9. Assoziation

**These3.** Im Endeffekt unterscheidet sich der OOP-First- von dem OOP-Later-Ansatz nur in einer unterschiedlichen Reihenfolge der Themen!

#### 4. Allgemein: Forschungsdesign

Diese empirische Studie als wissenschaftlicher Beitrag soll dabei helfen, ein praktisches relevantes Problem besser als bisher zu lösen. [PR05]

Das Forschungsdesign ist dabei ein kausales Design mit dem Ziel, Zusammenhänge zwischen Variablen aufzudecken. Dazu wird hier ein Experiment durchgeführt. [SR07]

Das Feldexperiment wird an einer kleinen und leicht zu erreichenden Gesamtheit durchgeführt, sprich meinen Schülern. [SHE05]

Über eine geschichtete Stichprobenauswahl werden einzelne Schüler-Merkmale (wie Geschlecht, Alter, etc.) schon von vornherein in ihrem richtigen Verhältnis im Sample repräsentiert. [Hö05]

Die verschiedenen Fragebögen werden aus den Forschungsfragen entwickelt und überarbeitet bzw. validiert (Vorstudie). [Hö02]

Für die so gewonnenen Ergebnisse aus der explorativen Studie kann nicht unbedingt eine Allgemeingültigkeit in Anspruch genommen werden, da aufgrund der Stichprobengröße und einzelner unterschiedlicher Einflussfaktoren eine Generalisierung teilweise schwierig erscheint. [WiDe07]

#### 5. Speziell: Studiendurchführung, Studienidee und Studienziel [Eh07]

**Studiendurchführung.** Zwei Schulklassen der Einführungsphase der gymnasialen Oberstufe des Oberstufenzentrums Informations- und Medizintechnik werden nach Geschlecht, Alter (ca. 17 Jahre), Vorbildung (in der Regel Realschulabschluss), Mathematik- und Deutsch-Note gleichartig zusammengesetzt und ein Schuljahr lang 3 Stunden (à 45 Minuten) pro Woche in Informatik beschult. Die Vorstudie lief im Schuljahr 2006/2007, die Hauptstudie läuft zur Zeit, also im Schuljahr 2007/2008.

Die Themen sind gleichartig, nur die Reihenfolge der Themen ist eine andere!

Die OOP-First-Klasse startet gleich mit OOP-Themen bzw. OOP-Fachtermini:

1	Einführung in die OOP (mit BlueJ)
2	Einführung in den Editor, Datentypen, Variablen, Kontrollstruktur: Sequenz (inkl. Struktogramm)
3	IDE: Klassenimplementierung und Objekterzeugung
4	Kontrollstruktur: Selektion (inkl. Struktogramm)
5	Methoden
6	Kontrollstruktur: Iteration (inkl. Struktogramm)
7	Arrays und Strings
8	Vererbung
9	Assoziationen inkl. Klassenattribute und -methoden

**Darstellung5: Themenreihenfolge (OOP-First)**

Die OOP-Later-Klasse fokussiert sich erst einmal (ca. ½ Schuljahr) auf prozedurale Themen inkl. Daten- und Kontrollstrukturen:

1	Einführung in den Editor, Datentypen, Variablen, Kontrollstruktur: Sequenz (inkl. Struktogramm)
2	Kontrollstruktur: Selektion (inkl. Struktogramm)
3	Kontrollstruktur: Iteration (inkl. Struktogramm)
4	Arrays und Strings
6/5*	Einführung in die OOP (mit BlueJ)
7/6*	IDE: Klassenimplementierung und Objekterzeugung
5/7*	Methoden
8	Vererbung
9	Assoziationen inkl. Klassenattribute und -methoden

**Darstellung6: Themenreihenfolge (OOP-Later)**

- linke Zahl: Reihenfolge Vorstudie, rechte Zahl: Hauptstudie

Bei der letzten Darstellung erkennt man schon, dass Lehren aus der Vorstudie für die Hauptstudie gezogen wurden: Das Thema „Methoden“ wird jetzt nicht mehr vor dem OOP-Einstieg, sondern erst danach unterrichtet.

**Studienidee.** Die Studie beobachtet zwei Aspekte im Schüler-Lernprozess.

**Die subjektive Seite:** Wie empfinden die Schüler den Unterricht, die einzelnen Themen und ihren eigenen Lernprozess. Wie schwierig werden einzelne Themen empfunden, wie ist der Interessens- und Motivationsverlauf bezogen auf ein ganzes Unterrichtsjahr. Diese Seite ist wichtig, da das emotional-motivationale Erleben des Unterrichtsgeschehens und die Intensität und Qualität der Motivation den Lernerfolg mitbestimmen. [Se05]

Konkret: Wenn ein Thema von den Schülern als schwer empfunden wird, können sich Denk- und Lernblockaden entwickeln. Wenn man erreichen könnte, dass das subjektive „Schwere-Erleben“ gemindert wird, dann könnte man evt. für einen größeren Lernerfolg sorgen.

**Die objektive Seite:** Gibt es nach einem Jahr signifikante Unterschiede im Lernerfolg der Schüler der beiden Klassen? Kann man diese aufschlüsseln nach einzelnen Themengebieten?

Für beide Aspekte werden Daten erhoben [LR98]. Zuerst wird aber am Anfang des Schuljahres überprüft, ob beide Klassen in ihrer Zusammensetzung, was die Computer-Vorkenntnisse anbelangt, gleichartig sind. Diese Eingangsbefragung hat z.B. die folgende Frage:

12. Ich habe folgende Kenntnisse im Schreiben von Computer-Programmen (z.B. mit Basic, Pascal, Java)  
 sehr fundiert  fundiert  durchschnittlich  wenig  keine

Für die subjektiven Fragestellungen gibt es nach jedem der neun Thema (also ca. alle 3 Wochen) einen Fragebogen für jeden Schüler. Einige Fragen des Schülerfragebogen beziehen sich allgemein auf das private Umfeld und auf die Schule, z.B.

1. *Mir geht es zur Zeit in meinem privaten Umfeld eher ...*

*sehr gut (1) (2) (3) (4) (5) sehr schlecht*

2. *Ich fühle mich zur Zeit in der Schule eher ...*

*sehr wohl (1) (2) (3) (4) (5) sehr unwohl*

Andere Fragen beziehen sich allgemein auf das Fach Informatik, z.B.:

8. *Ich fühle mich allgemein beim Unterricht im Fach Informatik eher ...*

*sehr wohl (1) (2) (3) (4) (5) sehr unwohl*

Die Hälfte aller Fragen beziehen sich aber auf das subjektive Erleben des Unterrichts des letzten Themas, wie z.B.:

12. *Den Stoff des letzten Themas empfand ich als ...*

*sehr leicht (1) (2) (3) (4) (5) sehr schwer*

15. *Ich fühlte mich speziell beim letzten Thema eher ...*

*sehr wohl (1) (2) (3) (4) (5) sehr unwohl*

Mit diesen Fragen lassen sich z.B. die These1 und die These2 überprüfen!

Der Lehrer füllt zur gleichen Zeit, also auch nach jedem der neun Themen, einen Lehrerfragebogen aus, der teilweise „gespiegelte“ Fragen enthält, wie z.B.

3. *Ich fühle mich allgemein beim Unterricht in dieser Klasse eher ...*

*sehr wohl (1) (2) (3) (4) (5) sehr unwohl*

4. *Den Stoff des letzten Themas halte ich eigentlich für ...*

*sehr leicht (1) (2) (3) (4) (5) sehr schwer*

Für die objektive Seite des Lerngeschehens wird in beiden Klassen nach einem Jahr der selbe Vergleichstest über zwei Stunden, also 120 Minuten geschrieben:

Nr.	Themengebiet	Punkte
1	Daten- und Kontrollstrukturen	20
2	Struktogramme	20
3	Methoden	20
4	Statische OOP	20
5	UML-Diagramme	20
6	Dynamische OOP	10

Jedes der ersten 5 Themengebiete enthält einen Multiple-Choice-Teil (10 Fragen à 1 Pkt.):

3	Welche Werte kann eine Variable vom Datentyp byte annehmen?	<input type="radio"/> 0 bis 255 <input type="radio"/> 0 bis 65535 <input type="radio"/> -128 bis 127 <input type="radio"/> Die Werte sind hardwareabhängig
---	---	---

Und einen Teil mit ca. zwei bis vier offeneren Fragen, z.B.:

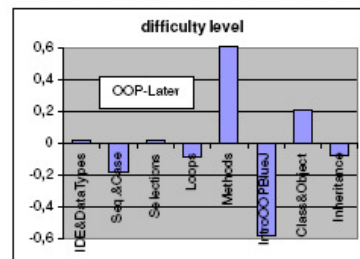
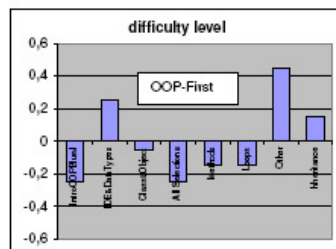
3. In der Klasse „Formelsammlung“ soll eine Methode „quadriere“ implementiert werden, die das Quadrat der übergebenen Double-Zahl zurückgibt. Geben Sie den Quelltext der Methode „quadriere“ an (Methodenkopf und Implementierung)! 3P

Das sechste Themengebiet ist eine Zusatzaufgabe ohne Multiple-Choice-Fragen.

**Studienziel.** Das Ziel ist nicht einen „Königsweg“ zu finden, sondern die Unterschiede und speziellen Schwierigkeiten der beiden Vorgehensweisen herauszuarbeiten, um bei beiden Wegen die Lernerfolge zu optimieren. Wenn erst einmal einzelne Stolpersteine erkannt sind (z.B. Schwierigkeiten bei den Themen-Übergängen), dann kann darüber diskutiert werden, welche Änderungsmöglichkeiten es im unterrichtlichen Geschehen geben könnte. Auch können allgemeine Prinzipien der Didaktik mit herangezogen werden, z.B. dass es im Unterricht oft besser ist, vom „Leichten zum Schweren“ zu gehen. Dann müsste gefragt werden, ob die Themen der OOP bzw. auch die der OOM sich nicht auf einem höheren „Lernschwierigkeitsniveau“ befinden als z.B. die Themen Daten- und Kontrollstrukturen. Wenn ja, wäre dies eine mögliche Begründung für einen OOP-Later-Einstieg. Die Fragebögen könnten (evt. leicht modifiziert) in Zukunft von jedem Lehrer für seinen speziellen Unterricht verwendet werden und Rückschlüsse auf spezifische Stärken und Schwächen seines Unterrichts geben!

### 6. Erste Ergebnisse [Eh07]

Ein erster Probelauf (Vorstudie) im Schuljahr 2006/2007 hat interessante Unterschiede beim subjektiven Erleben des Unterrichtsgeschehens erbracht, z.B. ist das sog. „Schwere-Empfinden“ der Themen teilweise in den beiden Klassen sehr unterschiedlich:



#### Darstellung7 [ES07]: „Schwere-Empfinden“ in Abhängigkeit der Themen

Der Durchschnittswert (Nulllinie) der „Themen-Schwere“ ist bei beiden Klassen bezogen auf alle neun Themen fast gleich: 2,65 bzw. 2,59. Also hat sich die These nicht bestätigt, dass durch einen OOP-Later-Einstieg das subjektive Schwere-Empfinden reduziert werden kann.

Der OOP-First-Klasse ist z.B. der Umstieg vom BlueJ-Tool auf den textorientierten Java-Editor schwerer gefallen als der Parallelklasse, die diesen Editor ab der ersten Stunde benutzt hat. Dies ist aber nicht signifikant ( $z = 1,4$  und  $p = 0,162$ ).

Dafür empfand die OOP-Later-Klasse das Thema Methoden als sehr schwer, was signifikant ist ( $z = 2,69$  und  $p = 0,007$ ).

Die objektiven Lernergebnisse (ermittelt durch den Vergleichstest am Ende des Schuljahrs) waren bei der OOP-Later-Klasse teilweise signifikant besser, gerade im Hinblick auf die OOP-Themen:

Nr.	Themengebiet	Ergebnis (OOP-First)	Ergebnis (OOP-Later)
1	Daten- und Kontrollstrukturen	61%	59%
2	Methoden	67%	64%
3	Struktogramme	60%	64%
4	„Statische“ OOP	72%	79%
5	UML-Klassen- und -Objektdiagramme	72%	85%
6	„Dynamische“ OOP	50%	67%
	<b>Gesamt</b>	<b>65%</b>	<b>70%</b>

#### Darstellung8: Objektive Unterschiede in den Lernergebnissen

Der UML-Unterschied ist signifikant ( $F = 1,5$  und  $t = 2,48$  und  $p = 0,017$ ), der Dynamische-OOP-Unterschied ist sehr signifikant ( $F = 1,4$  und  $t = 3,35$  und  $p = 0,0016$ ).

Auffällig war auch, dass die OOP-First-Klasse mit den Themen schneller vorangekommen ist, so dass ein weiteres Thema (kurzes OOP-Projekt) behandelt wurde. Dies kann ein Vorteil des OOP-First-Wegs sein, evt. aber auch die Erklärung dafür, warum die Schüler schlechter abgeschnitten haben! Vielleicht „verführt“ der OOP-First-Einstieg zu einem schnelleren Vorgehen. Eine andere Erklärung ist, dass Schüler das zuletzt Gelernte am Besten abrufen können. Dies waren bei der OOP-Later-Klasse ja die OOP-Themen!

#### 7. Probleme

Es gibt einige Probleme im Hinblick auf die Verallgemeinerung der Aussagen.

**Stichprobengröße.** Die untersuchten Gruppen (Stichprobengrößen) sind relativ klein. Allerdings fließt die Anzahl  $n$  der Schüler in die statistischen Formeln ein und wird daher berücksichtigt.

**Stichprobenauswahl.** Über die geschichtete Stichprobenauswahl werden zwar einzelne Schüler-Merkmale (wie Geschlecht, Alter, Vorbildung etc.) von vornherein in ihrem richtigen Verhältnis im Sample repräsentiert, aber sind die Lerngruppen bezogen auf ihre informatischen Fähigkeiten wirklich gleich? Der Nachweis der Gleichheit erfolgt auf zweierlei Wegen: zum einen über die Eingangsbefragung über die Computer-Vorkenntnisse. Zum anderen wird in der zur Zeit laufenden Hauptstudie der selbe Mathematiklehrer in den beiden Klassen eingesetzt und über die resultierenden zwei Mathematik-Durchschnittsnoten geprüft, ob die Klassen, was die naturwissenschaftliche Begabung anbelangt, wirklich gleich zusammengesetzt waren.

**Störgrößen.** Welche Art von Störgrößen gibt es im Hinblick auf das kausale Forschungsdesign, welches zum Ziel hat, den Einfluss der Eingangs-Variablen „Themenreihenfolge“ auf die Ausgangs-Variablen „Subjektives Erleben“ und „Objektiver Lernerfolg“ aufzuzeigen. Das Problem wird eher nicht in der Stichprobenauswahl (siehe oben) sondern im unterschiedlichen Lehrereinsatz liegen, da beide Klassen von unterschiedlichen Informatiklehrern beschult werden. Dies ist aber bewusst erfolgt, damit der Vorwurf nicht lauten kann, dass die Ergebnisse durch mein Lehrerverhalten in eine von mir wie auch immer gewünschte Richtung gehen. Der in der OOP-First-Klasse eingesetzte Lehrer besitzt langjährige Berufserfahrung und eine hohe Informatik-Lehrer-Kompetenz und ist daher auch zur Zeit Fachseminarleiter Informatik. Er ist ein großer Befürworter des „OOP-First“-Ansatzes.

Die Störgröße „Unterschiedlicher Lehrer“ hat zwei Dimensionen: Die persönliche Seite und das methodisch-didaktische Vorgehen des Lehrers.

Einige Fragen zum Lehrer und zur Lernatmosphäre in den Schülerfragebögen sollen dabei die Gleichartigkeit der persönlichen Seite der beiden Lehrer nachweisen:

7. Den unterrichtenden Lehrer (Herr F. oder Herr E.) finde ich ...

keine Aussage (0)  
sehr gut (1) (2) (3) (4) (5) sehr schlecht

17. Die Darstellung und Veranschaulichung der Lerninhalte des letzten Themas waren ...

sehr verständlich (1) (2) (3) (4) (5) sehr unverständlich

18. Die Arbeitsatmosphäre in der Klasse beim letzten Thema war eher ...

sehr gut (1) (2) (3) (4) (5) sehr schlecht

19. Die Unterstützung des Lehrers bei meinen Lernversuchen beim letzten Thema war ...

keine Aussage (0)  
sehr gut (1) (2) (3) (4) (5) sehr schlecht

Der methodisch-didaktische Störgrößenanteil soll durch die sehr guten Absprachen zwischen mir und meinem Kollegen reduziert werden:

- Vergleichbare Übungsaufgaben und Übungsphasen
- Ähnlich aufgebaute Arbeitsblätter
- Vergleichbarer Medieneinsatz
- Vergleichbare Methodik
- Gleiche Werkzeuge (BlueJ und JavaEditor)

Als Beleg wird u. a. von jeweils einem Schüler aus beiden Klassen der gesamte Ordner-Inhalt im Anhang der Studie zu finden sein.

**Signifikanz der Ergebnisse.** Es wird drei Arten von Aussagen geben:

1. Signifikante Aussagen, wie z.B.: „OOP-Later-Schüler haben bessere Lernerfolge in der „dynamischen OOP“ als OOP-First-Schüler!“

(Hinweis: Signifikante Aussagen müssen nicht richtig sein, z.B. könnten die verschiedenen Lehrer die Ursache sein.)

2. Nicht signifikante Aussagen, die trotzdem breite Zustimmung finden werden, wie z.B.: „Der Einstieg in die OOP mit BlueJ fällt OOP-Later-Schülern leichter als OOP-First-Schülern!“

(Hinweis: Nicht signifikante Aussagen können trotzdem richtig sein)

3. Nicht signifikante Aussagen, die interessant sind und einer weiteren Beobachtung bedürfen, wie z.B.: „Mit dem OOP-First-Einstieg wird (kann) in der Regel schneller unterrichtet (werden)!“

## 8. Resümee

### Die subjektive Seite:

- Die Studie zeigt u. a. das emotional-motivationale Erleben von Informatik-Unterricht für Schüler bezogen auf ein ganzes Schuljahr (auch im Kontext zum allgemeinen schulischen und privaten Umfeld).
- Zu neun unterschiedlichen Zeitpunkten wird jeweils eine Momentaufnahme erstellt, die auch abhängig ist vom zuletzt unterrichteten Thema. Es werden dabei u. a. Schwierigkeiten bei den Themenübergängen herausgearbeitet, genauso wie subjektives „Schwere-Empfinden“ oder „Wohlfühlfaktor“.
- Der Fragebogen kann (mit ganz leichten Veränderungen) von jedem Lehrer in jedem Fach eingesetzt werden und ermöglicht dem Lehrer, ausgehend von der Analyse des emotional-motivationalen Erlebens der Schüler, für eine Verbesserung dieses Erlebens zu sorgen.



#### Die objektive Seite:

- Die Studie zeigt u. a. den Lernerfolg der Schüler nach einem Jahr in Abhängigkeit von der Vorgehensweise im Kontext zu speziellen Themengebieten.
- Diese Ergebnisse können zu einer Reflektion des Lehrers führen, warum es zu unterschiedlichen Lernerfolgen in einzelnen Themengebieten gekommen ist und damit zu einer Verbesserung zukünftigen Unterrichts führen.
- Der große Vergleichstest (könnte mit Modifikationen) von jedem Informatik-Lehrer eingesetzt werden.


#### Referenzen

- [BK06] Barnes D.J., Kölling M.: BlueJ - Objects First with Java A Practical Introduction  
Prentice Hall / Pearson Education, 2006 ISBN 0-13-197-629X  
<http://www.bluej.org/>
- [Eh07] Ehlert A., Stechert P. (Hrsg.): Informatische Bildung in der Wissensgesellschaft  
S.17: Studie: Objects-First- und Objects-Later-Einstieg  
Universi-Verlag, ISBN: 978-3-936533-23-1
- [ES07] Ehlert A., Schulte C.: Learners Views on Objects-First and Objects-Later -Results of an Exploratory Study  
Beitrag für die ECOOP2007-Konferenz  
<http://www.cs.umu.se/~jubo/Meetings/ECOOP07/Submissions/EhlertSchulte.pdf>
- [Hö02] Höpflinger F.: Stichworte zu Forschungsplan und Forschungsdesign, 2002  
<http://www.hoepflinger.com/fhtop/fhmethod1A.html>
- [Hö05] Höpflinger F.: Stichprobenauswahl und Samplingverfahren,2005  
<http://www.hoepflinger.com/fhtop/fhmethod1F.html>
- [LR98] Lienert G., Raatz U.: Testaufbau und Testanalyse, 1998  
BELTZ, PVU ISBN-13: 978-3621274241
- [MJ2001] Mitchell M. L., Jolley J. M.: Research Design Explained, Clarion University of Pennsylvania, 4th ed.,2001  
<http://de.wikipedia.org/wiki/Forschungsdesign>
- [MPW06] McLaughlin B., Pollice G., West D.: Head First Object-Oriented Analysis and Design  
November 2006 ISBN: 0-596-00867-8
- [Nø06] Nørmark K.: From structured programming to object-oriented programming  
Aalborg University 1/29/2006  
[oldwww.cs.aau.dk/~nørmark/ooop-06/html/notes/intro-ooop\\_themes-from-structured-prog-to-ooop-sect.html](http://oldwww.cs.aau.dk/~nørmark/ooop-06/html/notes/intro-ooop_themes-from-structured-prog-to-ooop-sect.html)
- [Po05] Pollice G.: Beyond an object-first approach Professor of Practice  
Worcester Polytechnic Institute 10/15/2005  
<http://www-128.ibm.com/developerworks/rational/library/oct05/pollice/index.html>
- [Pr05] Prechelt L.: Promotion am Fachbereich Mathematik und Informatik  
<https://www.mi.fu-berlin.de/w/Mi/PromotionsVerfahren>
- [Sch03] Schulte C.: Lehr-Lernprozesse im Informatik-Anfängerunterricht  
Universität Paderborn, Oktober 2003  
<http://www.informatikdidaktik.de/Examensarbeiten/Schulte2003.pdf>
- [SHE05] Schnell H., Hill P. B., & Esser E. (2005): Methoden der empirischen Sozialforschung (S. 211-263).  
Oldenbourg-Verlag, ISBN-13: 978-3486250435
- [Se05] Sembill D., Emotional-motivationales Lernen in berufsbildenden Lernprozessen  
Otto-Friedrich-Universität Bamberg 13.01.2005  
[http://www.ewi-psy.fu-berlin.de/einrichtungen/arbeitsbereiche/allg\\_paedagogik/media/iz11/06\\_sembill.pdf](http://www.ewi-psy.fu-berlin.de/einrichtungen/arbeitsbereiche/allg_paedagogik/media/iz11/06_sembill.pdf)
- [Sp05] Spolwig S.: Methodische Probleme mit OOP im Anfangsunterricht  
[http://www.schule.de/schulen/oszhd1/gymnasium/faecher/informatik/didaktik/ooop\\_probleme.htm](http://www.schule.de/schulen/oszhd1/gymnasium/faecher/informatik/didaktik/ooop_probleme.htm)
- [SR07] Srnka K. J., Reisinger H., Heindler M.: Marketing-Lexikon (Stichwort: Forschungsdesign)  
<http://www.wiwi-treff.de/home/mllexikon.php?mpage=be@forschdesign.htm>
- [Ta06] Tang L.S.: Some Effective OOP Examples  
Department of Mathematics and Computer Science  
Western New England College Springfield, MA 01119, 2006  
<http://www.cs.umu.se/~jubo/Meetings/ECOOP06/Submissions/10-Tang.pdf>
- [Th05] The scripts developer network / forum Teaching C# to Junior High Pupil  
11/15/2005  
<http://www.thescripts.com/forum/thread237389.html>
- [WiEn07] Wikipedia, the free encyclopedia, May 2007-07-07  
Object-oriented comparison Structured Programming  
[http://en.wikipedia.org/wiki/Structured\\_programming](http://en.wikipedia.org/wiki/Structured_programming)
- [WiDe07] Wikipedia, die freie Enzyklopädie, Dez. 2007  
Forschungsdesign  
<http://de.wikipedia.org/wiki/Forschungsdesign>

I) Auswahl der Stichproben (Schülerdaten, Hauptstudie 2007/2008)

Nr.	Nati	m/w	Geb.-	Schulart	Mat	Det	OG71	OG72	Geb71	Math71	De71	Geb72	Math72	De72
1	deut	w	1990	Real	2	2	1	0	1990	2	2	0	0	0
2	deut	m	1990	Real	2	2	1	0	1990	2	2	0	0	0
3	deut	m	1990	Gym	3	3	0	1	0	0	0	1990	3	3
4	deut	m	1989	Gym	4	4	1	0	1989	4	4	0	0	0
5	deut	m	1990	Gesamt	3	3	1	0	1990	3	3	0	0	0
6	deut	m	1990	Real	2	2	1	0	1990	2	2	0	0	0
7	deut	m	1989	Real	2	3	1	0	1989	2	3	0	0	0
8	deut	m	1988	Gym	2	2	1	0	1988	2	2	0	0	0
9	deut	m	1989	Real	3	3	1	0	1989	3	3	0	0	0
10	deut	m	1990	OSZ-W	3	4	1	0	1990	3	4	0	0	0
11	deut	m	1987	OSZ-W	5	4	1	0	1987	5	4	0	0	0
12	ukra	m	1990	Real	3	2	0	1	0	0	0	1990	3	2
13	deut	w	1990	Real	2	2	0	1	0	0	0	1990	2	2
14	deut	m	1990	Gym	1	1	1	0	1990	1	1	0	0	0
15	deut	m	1989	Real	3	3	1	0	1989	3	3	0	0	0
16	deut	m	1987	Real	2	2	1	0	1987	2	2	0	0	0
17	deut	m	1990	Gesamt	2	2	0	1	0	0	0	1990	2	2
18	deut	m	1990	Gym	5	4	0	1	0	0	0	1990	5	4
19	deut	m	1990	Gym	3	3	0	1	0	0	0	1990	3	3
20	deut	m	1990	Real	2	2	1	0	1990	2	2	0	0	0
21	deut	m	1990	Real	2	3	0	1	0	0	0	1990	2	3
22	deut	m	1991	Real	2	3	0	1	0	0	0	1991	2	3
23	deut	m	1990	Gesamt	1	2	0	1	0	0	0	1990	1	2
24	deut	m	1989	Gym	1	2	0	1	0	0	0	1989	1	2
25	deut	m	1990	Gesamt	2	2	0	1	0	0	0	1990	2	2
26	deut	m	1989	Real	3	3	0	1	0	0	0	1989	3	3
27	deut	m	1989	OSZ-W	5	4	0	1	0	0	0	1989	5	4
28	deut	m	1989	Gesamt	3	3	1	0	1989	3	3	0	0	0
29	deut	m	1990	Real	2	1	1	0	1990	2	1	0	0	0
30	deut	m	1990	Gesamt	2	3	0	1	0	0	0	1990	2	3
31	deut	m	1991	Gym	2	3	1	0	1991	2	3	0	0	0
32	deut	m	1990	Gesamt	2	3	0	1	0	0	0	1990	2	3
33	deut	m	1991	Real	3	2	1	0	1991	3	2	0	0	0
34	deut	m	1990	Real	2	2	1	0	1990	2	2	0	0	0
35	deut	m	1988	Gym	4	5	0	1	0	0	0	1988	4	5
36	deut	m	1990	Real	3	3	0	1	0	0	0	1990	3	3
37	deut	m	1990	Real	3	3	0	1	0	0	0	1990	3	3
38	deut	m	1990	Real	2	3	0	1	0	0	0	1990	2	3
39	deut	m	1989	Gym	3	4	1	0	1989	3	4	0	0	0
40	deut	w	1991	Gesamt	4	3	1	0	1991	4	3	0	0	0
41	deut	m	1989	Real	3	2	0	1	0	0	0	1989	3	2
42	deut	m	1989	Real	3	3	1	0	1989	3	3	0	0	0
43	deut	w	1991	Real	3	3	0	1	0	0	0	1991	3	3
44	deut	m	1987	Real	2	2	0	1	0	0	0	1987	2	2
<b>Durchschnitt</b>							<b>22</b>	<b>22</b>	<b>1989,45</b>	<b>2,64</b>	<b>2,64</b>	<b>1989,68</b>	<b>2,64</b>	<b>2,82</b>

J) Eingangsbefragung bezüglich der Computerkenntnisse (Hauptstudie 2007/2008)

	Eingangsbefragung: <b>Computer-Vorkenntnisse</b>	Wissenschaftliche Untersuchung zu OOP-First bzw. OOP-Later im Informatik-Anfängerunterricht
---	---	---


	Ihr Code
Erster Buchstabe des Vornamens der Mutter	
Zweiter Buchstabe des Vornamens der Mutter	
Erster Buchstabe des Vornamens des Vaters	
Zweiter Buchstabe des Vornamens des Vaters	

1. Ich verfüge zu Hause über  
 einen eigenen Computer    einen fremden Computer (z.B. den der Eltern)    keinen Computer
2. Ich verfüge zu Hause über einen Internetanschluss  
 ja    nein
3. Ich verfüge zu Hause über eine Spielekonsole  
 ja    nein
4. Ich hatte bereits Informatik-Unterricht (Informationstechnische Grundbildung, ITG, etc.)  
 ja    nein
5. Ich downloade (Musik, Filme, Programme etc.) vom Internet  
 sehr viel    viel    durchschnittlich    eher weniger    sehr wenig
6. Ich recherchiere im Internet  
 sehr viel    viel    durchschnittlich    eher weniger    sehr wenig
7. Ich spiele am Computer oder mit der Spielekonsole  
 sehr viel    viel    durchschnittlich    eher weniger    sehr wenig
8. Ich habe folgende Kenntnisse mit einem Textverarbeitungsprogramm (z.B. WORD)  
 sehr fundiert    fundiert    durchschnittlich    wenig    keine
9. Ich habe folgende Kenntnisse mit einem Kalkulationsprogramm (z.B. EXCEL)  
 sehr fundiert    fundiert    durchschnittlich    wenig    keine
10. Ich habe folgende Kenntnisse mit einem Präsentationsprogramm (z.B. POWERPOINT)  
 sehr fundiert    fundiert    durchschnittlich    wenig    keine
11. Ich habe folgende Kenntnisse mit einem Datenbankprogramm (z.B. ACCESS)  
 sehr fundiert    fundiert    durchschnittlich    wenig    keine
12. Ich habe folgende Kenntnisse im Schreiben von Computer-Programmen (z.B. mit Basic, Pascal, Java)  
 sehr fundiert    fundiert    durchschnittlich    wenig    keine
13. Ich habe folgende Kenntnisse in der Erstellung von Webseiten (z.B. mit HTML, PHP, Javascript)  
 sehr fundiert    fundiert    durchschnittlich    wenig    keine
14. Ich freue mich auf den Informatikunterricht  
 sehr    eher mehr    durchschnittlich    eher weniger    wenig

Francioni / Ehlert    Oberstufenzentrum Informations- und Medizintechnik    August 2007

K) Fragebogen zum subjektiven Erleben der Schüler (Hauptstudie 2007/2008)

	Themenfragebogen: Emotional-motivationales Erleben des Unterrichtsgeschehens	Wissenschaftliche Untersuchung zu OOP-First bzw. OOP-Later im Informatik-Anfängerunterricht
---	--	---

	Ihr Code	Datum:
Erster Buchstabe des Vornamens der <b>Mutter</b>		Klasse:
Zweiter Buchstabe des Vornamens der <b>Mutter</b>		Thema-Nr.:
Erster Buchstabe des Vornamens des <b>Vaters</b>		
Zweiter Buchstabe des Vornamens des <b>Vaters</b>		

----- allgemeine Fragen zur Schule -----

1. Mir geht es zur Zeit in meinem privaten Umfeld eher ...  
 sehr gut    (1)    (2)    (3)    (4)    (5)    sehr schlecht
2. Ich fühle mich zur Zeit in der Schule eher ...  
 sehr wohl    (1)    (2)    (3)    (4)    (5)    sehr unwohl
3. Mein Stress in der Schule ist zur Zeit eher ...  
 sehr niedrig    (1)    (2)    (3)    (4)    (5)    sehr hoch
4. Meine Motivation in der Schule gute Leistungen zu erbringen ist zur Zeit eher ...  
 sehr hoch    (1)    (2)    (3)    (4)    (5)    sehr niedrig
5. Ich empfinde die Anforderungen der Schule an mich zur Zeit eher als ...  
 sehr niedrig    (1)    (2)    (3)    (4)    (5)    sehr hoch

----- allgemeine Fragen zum Fach Informatik und zum Lehrer -----

6. Das Fach Informatik (innerhalb der Technik) finde ich ...  
 sehr gut    (1)    (2)    (3)    (4)    (5)    sehr schlecht
7. Den unterrichtenden Lehrer (Herr F. oder Herr E.) finde ich ...  
 keine Aussage    (0)  
 sehr gut    (1)    (2)    (3)    (4)    (5)    sehr schlecht
8. Ich fühle mich allgemein beim Unterricht im Fach Informatik eher ...  
 sehr wohl    (1)    (2)    (3)    (4)    (5)    sehr unwohl
9. Mein Stress im Fach Informatik ist eher ...  
 sehr niedrig    (1)    (2)    (3)    (4)    (5)    sehr hoch
10. Meine Motivation im Fach Informatik gute Leistungen zu erbringen ist zur Zeit ...  
 sehr hoch    (1)    (2)    (3)    (4)    (5)    sehr niedrig
11. Ich empfinde die Anforderungen des Faches Informatik an mich eher als ...  
 sehr niedrig    (1)    (2)    (3)    (4)    (5)    sehr hoch

Ehlerl                      Oberstufenzentrum Informations- und Medizintechnik                      September 2007

----- Fragen zum letzten Thema -----

**12. Den Stoff des letzten Themas empfand ich als ...**

sehr leicht (1) (2) (3) (4) (5) sehr schwer

**13. Unabhängig von der Schwere des Stoffes fand ich das letzte Thema ...**

sehr interessant (1) (2) (3) (4) (5) sehr langweilig

**14. Den Stoffumfang des letzten Themas empfand ich als ...**

sehr niedrig (1) (2) (3) (4) (5) sehr hoch

**15. Ich fühlte mich speziell beim letzten Thema eher ...**

sehr wohl (1) (2) (3) (4) (5) sehr unwohl

**16. Das letzte Thema ist meiner Meinung nach für die Informatik ...**

sehr wichtig (1) (2) (3) (4) (5) sehr unwichtig

**17. Die Darstellung und Veranschaulichung der Lerninhalte des letzten Themas waren ...**

sehr verständlich (1) (2) (3) (4) (5) sehr unverständlich

**18. Die Arbeitsatmosphäre in der Klasse beim letzten Thema war eher ...**

sehr gut (1) (2) (3) (4) (5) sehr schlecht

**19. Die Unterstützung des Lehrers bei meinen Lernversuchen beim letzten Thema war ...**

keine Aussage (0)  
sehr gut (1) (2) (3) (4) (5) sehr schlecht

**20. Mein Stress beim letzten Thema war eher ...**

sehr niedrig (1) (2) (3) (4) (5) sehr hoch

**21. Ich beherrsche jetzt das letzte Thema ...**

sehr gut (1) (2) (3) (4) (5) sehr schlecht

**22. Mein Wissenszugewinn war ...**

sehr hoch (1) (2) (3) (4) (5) sehr niedrig

**23. Mein Arbeitsaufwand beim letzten Thema war ...**

sehr gering (1) (2) (3) (4) (5) sehr hoch

**24. Der Wiedererkennungswert zu bereits behandelten Themen war ...**

sehr hoch (1) (2) (3) (4) (5) sehr niedrig

**25. Die Nützlichkeit des Wissens bereits behandelter Themen für das letzte Thema war ...**

sehr hoch (1) (2) (3) (4) (5) sehr niedrig

L) Daten zum subjektiven Erleben der Schüler (Hauptstudie 2007/2008)

Klasse: OG71 Datum: 27.09.2007  
 Lfd. Nummer: 01 Thema-Nummer: 01  
 Thema: Einführung in die OOP mit BlueJ

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
1	ANDA	3	3	2	3	2	1	3	2	1	3	2	1	3	3	3	3	2	2	3	1	1	5	3			
2	ANFR	1	2	4	3	4	2	1	2	3	2	4	3	1	3	2	3	3	2	1	2	2	1	4			
3	ANPE	2	1	4	2	4	1	1	1	3	1	4	2	1	4	1	1	1	1	1	2	2	2	3			
4	ARST	1	2	3	2	3	1	1	1	2	1	2	2	2	1	2	1	1	2	1	2	2	2	3			
5	CLUL	2	2	4	2	5	1	2	2	4	2	4	2	3	4	2	1	1	1	2	2	2	1	2			
6	COVO	3	3	4	2	4	1		1	2	1	4	2	2	1	2	3	2	1	2	2	2	2	2			
7	DAHA	3	3	2	1	2	2	2	2	3	2	3	3	3	3	3	2	2	2	2	2	3	4	3			
8	EWAD	4	3	2	3	3	1	2	2	2	3	3	3	1	3	2	1	2	2	3	1	2	1	3			
9	JAKR	4	4	2	4	3	3	2	3	2	3	3	4	2	2	4	1	4	2	3	1	3	3	2			
10	JETH	2	2	3	2	3	2	2	2	3	2	3	2	4	2	3	1	2	2	1	3	2	1	3			
11	LIFR																										
12	LINO	4	3	3	3	5	1	1	2	2		3	1	1	3	1	2	1	1	1	2	2	1	2			
13	MARE	2	2	2	2	2	2	1	2	2	2	2	3	2	2	3	1	2	2	2	2	2	3	2			
14	NAGO	3	3	3	3	3	3	3	3	4	4	3	3	3	3	1	2	3	3	3	3	3	3	3			
15	NUYU	2	2	3	2	3	1	1	1	2	2	3	2	1	3	1	1	1	3	1	3	2	1	4			
16	PERA	4	2	3	2	3	3	2	3	3	3	4	4	3	3	4	2	4	3	3	3	4	3	3			
17	PETH	2	3	3	2	2	2		2	2	2	3	1	4	3	1	3	2	3		1	1	5	2			
18	REDE	3	4	2	5	3	4		4	4	4	3	3	2	3	4	2	4	3		2	4	4	3			
19	SIJO	1	2	1	2	2	2	1	2	1	2	2	3	2	3	2	2	3	2	2	2	2	3	3			
20	SIPE																										
21	SIVO	3	3	3	2	3	3		3	3	3	3	4	3	3	3	3	2	2		3	3	2	4			
22	SUNO	2	2	2	2	2	1	1	1	1	1	2	2	1	2	1	2	1	2	2	2	2	1	1	1		
Durchschnitt:		2,6	2,6	2,8	2,5	3,1	1,9	1,6	2,1	2,5	2,3	3,0	2,5	2,2	2,7	2,4	1,8	2,1	2,1	1,9	2,1	2,3	2,4	2,8			

Klasse: OG71 Datum: 01.11.2007  
 Lfd. Nummer: 02 Thema-Nummer: 02  
 Thema: Java-Editor, Ein- und Ausgabe, Datentypen, Variablen, Konstanten, Sequenz

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
1	ANDA	2	3	2	3	2	1	2		1	1	2	1	2	4	3	3	2	1	3	2	1	1	4	2	3	1
2	ANFR	2	3	4	2	4	2	1	2	3	2	4	3	1	3	2	2	2	2	2	2	3	3	1	4	2	2
3	ANPE	4	3	4	2	3	1	5	4	4	1	3	3	1	3	1	1	1	1	5	3	4	1	3	3	3	
4	ARST	2	2	3	1	3	1	2	3	3	1	3	3	2	3	3	1	3	2	2	3	3	2	3	3	3	
5	CLUL	3	3	4	4	4	2	2	3	5	2	4	5	2	5	4	1	3	2	3	4	3	1	4	3	2	
6	COVO																										
7	DAHA	5	4	2	1	2	3	1	2	5	3	5	3	3	5	3	3	3	2	2	3	3	1	4	4	3	
8	EWAD	3	3	3	3	4	2	2	2	3	3	3	3	1	3	3	1	3	3	3	3	3	1	3	3	2	
9	JAKR	4	3	3	4	3	3	2	3	2	3	4	3	2	3	3	2	2	1	3	3	3	2	2	4	3	
10	JETH	2	2	4	1	3	1	1	2	3	1	3	3	2	4	2	2	3	2	1	3	2	1	4	2	2	
11	LIFR	2	2	3	2	3	1	1		2	2	3	2	1	3	2	2	2	2	1	2	2	2	3	3	3	
12	LINO	4	3	4	3	4	2	1	2	3	4	3	2	1	3	2	2	2	1	1	3	2	1	4	1	1	
13	MARE	4	3	3	2	3	1	2	2	3	2	3	2	3	2	3	2	2	2	2	2	3	2	4	3	3	
14	NAGO	3	3	3	3	3	3	4	3	4	2	3	3	3	3	2	2	3	3	4	3	2	2	3	3	3	
15	NUYU	1	2	3	1	2	1	1	1	2	1	2	1	1	2	1	1	1	3	2	2	1	1	3	3	1	
16	PERA	2	2	3	3	3	3	4	4	4	3	4	4	3	3	4	2	3	3	2	4	4	2	3	4	3	
17	PETH	4	3	2	2	2	2	2	3	3	2	3	2	2	3	3	3	2	2	2	3	2	2	3	3	2	
18	REDE	4	4	4	4	3	4		4	5	4	4	4	2	3	4	2	5	3		4	4	4	3	3	4	
19	SIJO	2	1	2	3	2	2	1	2	2	3	3	3	2	3	2	2	1	2	2	3	2	2	3	3	2	
20	SIPE	3	3	4	3	4	2	3	3	4	2	4	2	2	4	3	3	4	2	4	3	2	2	3	5	4	
21	SIVO	3	3	3	3	2	3		3	3	3	3	3	2	3	3	3	2	3		3	3	2	3	2	3	
22	SUNO	2	2	2	1	2	1	1	1	1	1	2	1	2	2	2	2	2	3	2	2	2	1	1	1	3	2
Durchschnitt:		2,9	2,7	3,1	2,4	2,9	2,0	2,0	2,5	3,1	2,2	3,2	2,8	2,0	3,2	2,6	2,0	2,4	2,2	2,4	2,9	2,5	1,8	3,1	3,0	2,5	

Klasse: OG71 Datum: 15.11.2007  
 Lfd. Nummer: 03 Thema-Nummer: 03  
 Thema: Klassenimplementierung und Objekterzeugung

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANDA	2	2	3	3	3	1	2	2	2	3	2	1	2	3	2	2	2	2	2	1	1	4	2	2	2
2	ANFR	3	2	2	3	4	1	1	1	3	2	4	4	1	3	3	1	2	2	1	3	4	2	4	3	2
3	ANPE	2	2	4	2	4	1	1	1	3	2	3	3	1	3	1	1	1	2	1	3	2	1	3	3	2
4	ARST	2	2	3	2	3	2	3	3	3	2	4	4	2	3	3	1	3	2	2	3	4	2	3	2	2
5	CLUL	2	4	3	2	4	2	1	2	3	5	5	4	3	4	2	1	3	1	1	5	5	4	2	1	2
6	COVO	4	2	3	3	3	1	2	2	2	1	4	3	2	3	3	1	4	2	2	4	3	2	3	2	2
7	DAHA	3	4	3	2	2	3	2	3	2	1	2	4	2	5	3	3	3	4	1	2	3	1	2	4	2
8	EWAD	3	2	3	3	3	1	2	2	3	2	3	2	1	3	2	1	2	2	3	2	3	2	4	3	3
9	JAKR	4	4	2	4	2	1	2	2	2	3	2	2	2	3	3	1	2	1	3	2	2	2	3	4	2
10	JETH	1	2	3	2	3	3	1	2	3	2	3	3	2	4	3	2	2	2	1	3	3	2	3	3	1
11	LIFR	1	2	3	2	3	1	1	1	2	1	2	2	1	3	2	2	2	2	1	3	2	2	3	2	2
12	LINO	5	4	4	3	4	1	2	2	3	2	3	3	1	4	2	1	2	2	1	4	3	1	4	1	2
13	MARE	4	3	3	3	3	2	2	2	3	2	3	3	2	3	3	2	2	2	2	3	3	2	3	2	2
14	NAGO	3	3	3	4	3	3	3	3	4	4	3	3	3	3	3	5	4	3	4	3	3	2	3	2	2
15	NUYU	3	3	4	3	3	1	2	1	2	1	2	1	1	2	1	2	1	4	3	2	1	1	3	3	3
16	PERA	3	2	3	2	3	3	3	3	4	1	4	3	2	3	3	2	3	5	3	3	4	2	3	3	3
17	PETH	3	3	3	2	3	3	3	4	3	2	4	3	2	3	4	2	3	4		3	4	3	3	2	1
18	REDE	2	3	3	3	3	3	2	2	3	2	3	3	2	2	2	3	2	2	2	3	3	2	3	2	2
19	SIJO																									
20	SIPE	3	3	4	2	3	2	3	2	4	2	4	4	2	3	3	2	3	2	2	4	3	3	4	4	1
21	SIVO	3	3	3	3	3	3		3	3	2	3	4	3	3	3	3	2	3		3	2	2	3	2	1
22	SUNO	3	3	3	2	2	1	1	2	2	2	2	2	2	1	2	2	2	3	3	3	2	2	2	2	3
Durchschnitt:		2,8	2,8	3,1	2,6	3,0	1,9	2,0	2,1	2,8	2,1	3,1	2,9	1,9	3,0	2,5	1,9	2,4	2,5	2,0	3,0	2,9	2,1	3,0	2,5	2,0

Klasse: OG71 Datum: 07.02.2008  
 Lfd. Nummer: 05 Thema-Nummer: 04  
 Thema: Methoden

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANDA	3	3	3	3	2	1	2	2	4	3	2	2	3	3	2	2	2	3	2	4	2	3	4	3	3
2	ANFR	3	3	3	2	4	1	2	1	2	2	4	3	2	2	3	2	2	2	2	3	3	2	3	2	2
3	ANPE	1	2	3	3	4	1	1	2	3	2	3	3	1	3	1	1	2	3	1	3	2	1	3	2	1
4	ARST	1	2	3	2	3	1	3	2	3	1	3	2	2	2	2	1	2	2	2	3	2	1	3	3	3
5	CLUL	1	2	3	3	4	2	2	2	3	2	4	4	2	4	2	1	3	3	2	4	2	1	4	2	1
6	COVO	4	3	3	4	4	2	2	2	3	3	3	2	4	2	1	2	2	3	3	2	2	3	4	2	2
7	DAHA	5	5	1	4	5	3	2	3	2	4	2	3	2	4	4	1	4	4	2	2	3	4	3	4	3
8	EWAD	2	2	3	2	3	1	2	2	2	2	3	2	2	3	2	2	2	2	3	2	2	2	3	3	2
9	JAKR	3	3	3	3	3	2	3	2	3	3	3	3	1	3	3	1	2	3	3	3	2	2	4	2	1
10	JETH	2	2	2	2	4	1	1	2	3	2	2	2	3	2	2	1	1	2	2	3	2	3	4	2	2
11	LIFR	1	3	2	1	3	1	1	1	3	2	3	2	2	1	1	1	1	1	1	3	3	2	2	3	3
12	LINO	4	4	4	4	3	2	2	4	4	3	5	3	3	3	4	2	3	2	4	3	4	3	2	2	2
13	MARE	5	3	2	4	3	2	2	2	2	3	3	3	2	3	2	2	2	2	2	2	2	3	3	3	2
14	NAGO	3	3	3	2	3	3	2	3	2	2	3	3	3	3	3	2	3	3	2	3	3	3	3	3	2
15	NUYU	3	2	3	1	2	1	1	1	1	1	3	2	1	3	1	1	1	3	2	1	1	3	1	3	1
16	PERA	2	2	3	3	3	4	3	5	4	2	4	4	3	3	4	3	3	2	4	3	4	3	3	2	3
17	PETH	1	3	2	2	2	2	2	2	2	2	2	1	2	3	1	1	3	1		2	1	3	2	2	1
18	REDE	2	3	2	3	2	2	2	2	4	3	4	3	3	3	3	3	3	3	2	3	3	3	2	2	2
19	SIJO	2	2	2	3	3	2	2	1	2	2	3	2	3	1	2	2	2	3	2	1	2	3	2	3	2
20	SIPE	4	3	4	2	3	2	2	3	2	3	3	2	3	2	2	2	2	3	2	3	3	2	3	2	2
21	SIVO	3	3	2	3	3	2		3	3	2	3	3	2	4	3	2	2	3		3	3	1	3	2	2
22	SUNO	2	2	2	2	2	2	2	2	2	2	2	1	2	1	2	1	2	3	2	2	1	3	1	3	1
Durchschnitt:		2,6	2,7	2,6	2,6	3,1	1,8	2,0	2,2	2,7	2,3	3,0	2,6	2,2	2,8	2,3	1,6	2,2	2,5	2,3	2,7	2,4	2,4	2,8	2,6	2,0

Klasse: OG71 Datum: 13.12.2007  
 Lfd. Nummer: 04 Thema-Nummer: 05  
 Thema: Selektionen

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANDA	2	2	3	2	3	1	1	2	2	2	3	2	3	3	3	2	2	3	2	3	2	3	3	2	2
2	ANFR	3	2	4	1	2	2	1	2	2	2	3	3	1	3	2	2	2	2	1	3	2	3	3	2	2
3	ANFE	2	2	4	2	3	1	1	2	4	2	4	3	2	3	2	1	1	3	1	3	3	1	3	2	2
4	ARST	2	3	3	2	3	2	3	3	4	2	4	3	2	3	3	1	3	3	2	4	3	2	3	2	3
5	CLUL	2	4	5	5	5	3	1	3	3	4	5	5	2	4	4	1	3	5	5	5	4	3	2	1	5
6	COVO	3	3	2	3	4	1	3	2	3	3	3	2	4	2	3	3	2	2	3	3	2	3	1	2	2
7	DAHA	3	3	4	2	2	2	1	3	3	2	4	4	3	4	3	4	3	2	2	2	2	3	3	4	3
8	EWAD	2	2	2	2	3	1	2	1	2	2	3	3	1	3	3	2	3	2	3	3	3	2	4	3	2
9	JAKR	3	3	2	3	2	3	3	3	3	3	3	4	2	3	3	2	3	3	3	3	4	3	3	3	3
10	JETH	2	2	3	1	3	2	2	2	3	1	3	3	3	3	3	2	2	2	3	2	2	3	3	2	2
11	LIFR	2	2	2	2	3	1	1	1	1	1	3	3	1	3	1	1	1	1	1	1	1	1	2	2	2
12	LINO	4	4	4	3	5	2	2	3	4	4	3	3	3	4	4	2	3	3	3	3	3	2	4	2	2
13	MARE																									
14	NAGO	3	3	3	4	3	3	3	3	4	2	3	3	3	3	3	1	2	3	2	3	2	2	2	2	2
15	NUYU	3	3	3	3	2	1	1	1	2	1	2	1	2	1	2	1	3	2	1	1	1	3	1	1	1
16	PERA	3	2	2	4	3	4	3	5	4	3	4	3	3	3	4	2	3	3	3	3	3	3	3	3	3
17	PETH	2	3	3	2	3	2		4	3	2	3	2	3	3	2	2	3	3		3	2	2	3	2	2
18	REDE	2	2	2	3	3	3	1	3	3	4	4	3	2	3	3	3	2	2	2	3	4	3	2	2	2
19	SIJO	2	1	1	2	3	1	2	1	2	1	2	2	2	3	3	2	2	3	1	2	2	2	3	3	2
20	SIFE	4	3	3	2	3	2	2	2	3	2	4	2	2	4	3	2	2	2	2	2	3	2	3	4	2
21	SIVO	3	3	3	3	3	2		3	3	2	3	2	3	2	3	2	2	3		3	2	2	2	2	1
22	SUNO	2	2	3	3	2	2	2	2	1	2	2	2	2	2	2	2	2	3	2	1	2	2	2	2	2
Durchschnitt:		2,6	2,6	2,9	2,6	3,0	2,0	1,8	2,4	2,8	2,2	3,2	2,8	2,3	3,0	2,8	1,9	2,3	2,6	2,2	2,7	2,5	2,4	2,6	2,3	2,2

Klasse: OG71 Datum: 07.03.2008  
 Lfd. Nummer: 06 Thema-Nummer: 06  
 Thema: Iterationen

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANDA	2	2	3	3	4	2	2	2	2	3	3	2	2	3	2	2	2	3	3	2	2	3	3	2	2
2	ANFR																									
3	ANFE	4	3	3	2	3	1	1	1	3	2	3	3	2	3	3	1	4	2	2	3	2	1	2	3	2
4	ARST	2	2	3	2	2	2	3	2	3	2	3	3	2	3	3	2	3	2	3	3	3	2	3	3	2
5	CLUL	1	3	5	2	5	2	1	3	4	2	5	4	2	4	3	2	3	2	2	4	3	1	4	2	2
6	COVO	3	3	3	3	4	2	2	2	4	3	2	3	3	3	3	3	3	3	3	3	4	5	1	4	4
7	DAHA	3	4	4	1	4	3	2	4	5	1	4	5	1	4	5	1	3	5	2	4	1	1	5	5	4
8	EWAD	2	4	4	2	3	1	3	2	3	2	3	2	1	3	2	2	2	2	3	3	3	2	4	3	2
9	JAKR	3	3	2	3	2	2	2	3	2	3	2	3	4	2	3	2	3	3	3	2	3	3	2	2	2
10	JETH	2	2	3	2	3	2	2	3	3	2	3	2	2	3	2	2	3	3	3	3	2	2	3	3	1
11	LIFR	2	2	2	2	3	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
12	LINO	5	4	4	4	4	2	2	3	5	4	4	4	2	3	4	1	3	2	3	5	4	3	2	1	1
13	MARE	5	3	3	3	3	2	2	2	2	2	2	2	2	3	3	2	3	2	3	2	3	3	3	2	2
14	NAGO	3	3	3	3	3	2	3	3	3	2	3	3	3	3	3	2	3	3	3	3	2	3	3	2	2
15	NUYU	1	1	2	2	3	1	1	1	1	1	2	2	1	2	1	1	1	3	1	1	1	3	2	3	1
16	PERA	3	3	3	3	3	2	3	3	3	3	3	4	3	3	4	2	3	3	3	3	4	3	3	3	2
17	PETH	2	3	2	2	3	2		2	3	2	2	2	2	3	3	2	2	3		3	2	3	2	2	2
18	REDE	3	3	4	2	3	3	3	3	4	4	3		3	3	3	3	3	3	4	3	4	3	3	3	3
19	SIJO	2	2	3	2	3	1	2	2	1	2	2	2	2	3	2	2	2	2	3	2	2	2	3	2	2
20	SIFE	4	3	4	2	4	2	2	3	4	2	3	4	2	4	3	3	3	4	2	3	3	2	4	4	4
21	SIVO	3	3	3	3	3	3		3	3	3	3	3	3	3	3	3	3	3		3	3	3	3	3	3
22	SUNO	3	3	3	3	2	2	2	2	2	3	3	2	2	2	2	2	2	3	3	3	2	2	2	2	2
Durchschnitt:		2,8	2,8	3,1	2,4	3,2	1,9	2,1	2,4	2,9	2,4	2,9	2,9	2,2	3,0	2,9	2,1	2,7	2,9	2,7	3,0	2,7	2,5	2,9	2,7	2,3



Klasse: OG71 Datum: 24.04.2008  
 Lfd. Nummer: 07 Thema-Nummer: 07  
 Thema: Arrays und Strings

CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25		
1 ANDA	2	3	3	3	3	2	2	2	2	3	2	3	4	3	3	2	2	3	2	3	3	3	3	3	2	2	
2 ANFR	4	3	4	4	5	1	1	2	3	2	4	4	3	3	3	2	2	3	2	3	3	2	3	2	2	2	
3 ANPE	2	3	4	2	4	2	2	2	3	2	3	4	2	3	2	2	3	3	1	3	3	2	3	3	3	3	
4 ARST	1	2	3	2	3	1	2	2	3	2	3	3	1	2	2	1	2	2	2	3	2	2	2	2	3	2	
5 CLUL	2	3	4	3	4	3	2	3	4	2	4	4	2	4	4	1	3	2	2	4	3	2	4	2	4	2	3
6 COVO	3	2	4	3	4	1	2	1	4	2	4	2	1	4	2	1	2	3	3	2	3	2	2	1	2	2	
7 DAHA	5	4	5	2	5	2	2	3	4	1	4	5	3	5	2	1	4	4	2	4	3	1	4	4	5	5	
8 EWAD	3	2	3	2	3	1	2	2	3	2	3	3	2	3	2	2	2	3	3	3	2	2	3	3	3	3	
9 JAKR	2	3	2	3	3	2	2	4	3	2	2	2	1	2	2	3	2	3	3	2	2	2	2	2	2	3	
10 JETH	2	2	1	2	3	2	2	2	2	1	3	3	1	3	2	1	2	2	2	2	2	2	2	3	3	1	
11 LIFR	2	2	3	2	3	1	1	2	2	2	3	2	1	3	3	3	3	3	2	3	3	2	3	3	2	2	
12 LINO	4	4	5	4	4	2	2	4	4	3	4	4	3	4	4	2	3	3	3	4	5	2	2	1	1	1	
13 MARE	5	3	3	3	3	2	2	3	3	3	3	3	3	3	3	3	3	2	2	3	3	3	3	3	2	3	
14 NAGO	3	3	3	3	3	3	3	3	3	2	3	3	3	4	3	5	3	4	3	3	3	2	3	4	4	4	
15 NUYU	3	2	2	3	3	1	2	1	1	2	3	4	2	4	3	1	4	2	3	2	2	2	3	2	2	2	
16 PERA	1	2	2	3	3	2	2	2	3	3	4	2	2	3	3	2	3	3	3	3	2	2	2	2	3	3	
17 PETH	1	2	3	3	3	2	2	2	3	2	2	2	3	2	2	3	3	2	2	2	2	2	2	2	3	2	
18 REDE	3	3	4	3	4	2	1	3	4	2	4	3	3	3	2	2	3	2	2	4	3	2	4	3	3	3	
19 SIJO	2	2	3	3	2	1	1	2	2	3	2	3	2	4	2	1	2	3	3	2	2	3	4	3	2	2	
20 SIPE	4	3	4	2	4	3	2	2	4	2	4	4	1	3	2	2	2	4	2	4	3	3	4	4	4	4	
21 SIVO	3	3	2	2	3	2		3	3	2	3	3	2	4	3	2	4	3		4	3	2	2	2	2	2	
22 SUNO	2	2	3	3	2	1	1	2	1	3	1	2	1	2	2	1	2	3	2	2	2	2	1	2	2	2	
Durchschnitt:	2,7	2,6	3,2	2,7	3,4	1,8	1,8	2,4	2,9	2,2	3,1	3,1	2,0	3,3	2,5	1,9	2,7	2,8	2,4	3,0	2,6	2,2	2,8	2,5	2,5	2,5	

Klasse: OG71 Datum: 22.05.2008  
 Lfd. Nummer: 08 Thema-Nummer: 08  
 Thema: Vererbung

CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
1 ANDA	2	2	2	3	2	2	1	2	3	3	2	3	2	2	3	3	3	2	2	2	2	3	2	3	2	2
2 ANFR	1	2	4	3	4	2	1	1	3	2	4	4	1	2	3	1	2	2	2	3	3	2	4	2	2	2
3 ANPE	2	3	4	2	4	1	1	2	2	2	3	3	2	3	3	2	2	2	1	2	2	2	3	3	2	2
4 ARST																										
5 CLUL	2	2	4	2	4	3	2	3	5	3	4	4	3	4	3	1	3	2	2	4	3	2	4	3	3	3
6 COVO	4	2	4	3	3	1	2	1	2	2	2	2	2	3	3	2	2	3	3	2	2	1	2	3	2	2
7 DAHA	4	3	5	2	5	1	2	3	4	2	4	2	4	4	3	1	3	5	3	5	3	5	3	5	3	2
8 EWAD	4	2	3	2	3	1	3	2	3	2	3	3	2	3	3	2	2	3	3	3	2	2	3	3	3	3
9 JAKR	2	2	3	3	4	1	2	2	2	2	3	2	2	2	2	1	1	1	1	3	2	1	3	3	3	3
10 JETH																										
11 LIFR	2	2	3	2	3	1	1	1	3	2	3	2	1	3	2	1	1	1	3	2	2	2	3	2	1	1
12 LINO																										
13 MARE	4	3	3	3	3	2	2	2	2	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
14 NAGO	3	3	3	3	3	2	2	3	3	2	3	2	2	3	2	2	2	3	2	2	2	2	2	3	2	2
15 NUYU	3	3	2	3	3	1	1	1	2	1	3	2	2	2	2	1	1	3	2	2	1	1	1	3	1	1
16 PERA																										
17 PETH	2	3	2	3	3	2		2	2	2	3	3	2	3	3	2	2	2		3	2	3	2	3	2	2
18 REDE	2	3	4	3	3	2	2	3	4	3	4	2	2	2	3	2	2	2	2	2	3	2	3	3	3	2
19 SIJO	2	2	3	3	3	1	1	2	2	3	3	3	2	3	3	2	2	3	2	3	2	4	3	2	2	2
20 SIPE	4	3	4	2	3	2	2	2	4	3	3	2	2	3	2	2	3	3	2	2	2	1	4	4	4	2
21 SIVO	3	3	4	3	2	1		2	2	4	2	2	3	2	2	1	1	3		2	2	2	3	2	1	1
22 SUNO	2	2	2	3	2	2	2	2	1	3	2	1	1	2	2	1	2	3	2	1	2	1	1	3	1	1
Durchschnitt:	2,7	2,5	3,3	2,7	3,2	1,6	1,7	2,0	2,7	2,4	3,0	2,4	2,1	2,7	2,6	1,6	2,0	2,5	2,1	2,6	2,1	2,1	2,8	2,8	1,9	1,9

Klasse: OG71 Datum: 24.06.2008  
 Lfd. Nummer: 09 Thema-Nummer: 09  
 Thema: Assoziation

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANDA	3	4	4	4	3	2	2	2	2	3	2	2	2	3	3	2	2	3	2	2	2	3	3	3	2
2	ANFR	2	3	4	4	4	2	1	2	2	2	4	4	2	3	3	1	2	2	2	3	4	2	3	1	2
3	ANFE	4	3	4	2	4	1	1	2	2	2	2	3	2	3	2	2	2	3	1	2	3	1	2	4	2
4	ARST	2	2	2	2	3	2	2	2	3	2	3	3	3	3	3	2	3	2	3	2	3	3	2	3	3
5	CLUL	1	3	4	2	4	2		2	4	2	4	4	3	4	3	1	3	2		3	3	3	3	2	2
6	COVO	2	3	4	2	4	1	2	2	4	3	3	2	2	2	2	1	4	3	4	4	3	3	2	1	2
7	DAHA	5	5	2	5	2	4	2	3	2	4	2	1	4	4	4	1	3	5	3	2	1	3	1	4	4
8	EWAD	3	3	3	2	3	2	3	2	3	2	3	2	3	3	2	3	2	2	3	3	3	2	2	3	3
9	JAKR	3	3	2	3	2	2	2	4	3	3	3	3	3	2	3	2	3	3	3	2	4	3	2	2	3
10	JETH	2	2	3	2	3	2	2	2	4	2	3	2	2	3	2	1	2	2	2	3	3	2	4	2	2
11	LIFR	3	2	2	3	2	3	2	2	3	2	3	2	2	3	2	2	1	3	2	2	3	2	3	3	2
12	LINO	4	4	3	4	4	2	3	4	4	2	4	3	3	3	3	2	3	3	4	4	4	3	3	2	2
13	MARE																									
14	NAGO	4	4	3	4	3	3	1	3	3	4	3	3	2	3	3	2	2	5	2	3	3	2	2	3	3
15	NUYU	4	3	3	3	3	2	2	2	3	3	3	4	4	2	4	2	2	3	2	2	3	2	2	4	2
16	PERA	3	3	2	2	2			3	3	3	4	4	3		3	3	4	3	4	3	4	4	3	2	3
17	PETH	2	2	2	2	3	2		2	2	2	2	3	2	3	2	3	3	3		2	2	3	2	2	2
18	REDE	3	3	4	4	4	2	2	3	3	2	3	3	2	3	3	3	2	3	3	3	3	2	3	2	3
19	SIJO	2	3	3	3	2	2	2	2	3	2	3	2	3	2	2	1	2	2	3	2	2	3	2	3	2
20	SIPE	3	3	4	2	4	3	2	2	4	2	4	2	3	3	2	2	3	4	2	3	2	3	4	4	2
21	SIVO	3	3	2	2	3	2		3	2	2	3	2	2	2	3	2	4	2		2	2	2	3	2	1
22	SUNO	2	3	3	3	2	2	1	2	1	3	1	2	1	2	2	1	2	3	3	3	2	2	1	2	3
Durchschnitt:		2,9	3,0	3,0	2,9	3,0	2,1	1,9	2,4	2,8	2,5	3,0	2,7	2,5	2,8	2,7	1,8	2,6	3,0	2,7	2,7	2,8	2,5	2,6	2,6	2,4

Klasse: OG72 Datum: 31.01.2008  
 Lfd. Nummer: 05 Thema-Nummer: 01  
 Thema: Einführung in die OOP mit BlueJ

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
1	ANHA	2	2	3	3	3	2	2	2	3	2	3	3	1	3	2	2	2	1	2	3	2	1	3	3	2	
2	ANST																										
3	BAJO	2	2	3	2	4	1	1	2	2	2	3	3	4	3	3	5	1	2	1	2	2	2	2	2	1	
4	BERE																										
5	BETH	2	2	2	2	3	1	2	2	3	1	4	2	1	3	2	2	2	1	2	3	2	2	3	2	2	
6	BIFR	1	1	2	1	3	1	1	1	1	1	1	1	1	3	1	3	1	1	1	1	1	1	2	3	1	1
7	BIHO	2	2	3	3	3	1	2	2	3	3	2	1	1	3	1	1	1	1	2	2	1	5	1	1	1	
8	BITH	2	2	3	2	3	2	2	3	3	2	3	2	2	2	2	3	2	2	2	3	2	3	3	3	3	
9	BIUW	2	2	3	3	2	1	1	1	2	1	2	1	1	2	2	2	1	2	2	2	1	1	2	2	1	
10	CARU	4	2	4	2	4	1	1	1	2	1	2	1	1	2	1	2	1	2	1	2	2	1	2	2	2	
11	CHGE	2	3	3	3	4	1	1	1	1	1	1	1	1	5	1	1	2	2	2	1	1	5	1	4	3	
12	CHMI	2	2	3	2	3	2	2	3	4	2	5	2	1	3	1	3	2	1	2	3	2	1	2	3	4	
13	DAUW	2	2	3	3	3	2	2	3	4	3	4	3	1	3	2	1	2	1	3	3	3	2	4	1	2	
14	INWO	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1	2	2	1	2	1	2	2	1	2	2	
15	JUKL	2	2	3	1	3	2	2	2	3	1	3	2	1	2	2	3	2	2	3	2	3	2	3	2	2	
16	LASA	1	2	2	1	3	1	1	2	2	1	3	2	2	3	2	1	1	2	2	2	1	3	2	3	2	
17	MAAC																										
18	REYV	3	3	4	3	3	3	3	4	4	3	3	2	2	3	3	2	3	2	3	3	2	3	3	2	2	
19	SADI	4	3	2	3	3	2		2	2	2	2	2	3	3	2	2	3		2	2	3	2	3	3	3	
20	SILO	1	1	2	2	2	1	1	2	2	1	3	3	1	4	2	1	2	1	2	3	2	1	3	3	2	
21	TAWA	2	1	1	2	3	1	2	2	2	2	4	3	1	3	2	3	1	1	1	2	3	1	3	2	1	
22	VEVA	3	3	2	3	2	2	2	2	2	3	4	2	1	3	2	2	1	1	4	2	3	2	1	2	2	
Durchschnitt:		2,1	2,1	2,6	2,3	2,9	1,5	1,7	2,1	2,4	1,7	2,8	1,9	1,4	2,8	1,8	2,2	1,6	1,5	2,1	2,2	1,9	2,2	2,3	2,3	2,0	

Klasse: OG72 Datum: 27.09.2007  
 Lfd. Nummer: 01 Thema-Nummer: 02  
 Thema: Java-Editor, Ein- und Ausgabe, Datentypen, Variablen, Konstanten, Sequenz

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANHA	1	2	3	2	3	3	2	2	4	2	3	3	2	3	3	2	4	1	4	3	3	2	3		
2	ANST	4	3	2	2	2	2	2	2	3	2	1	1	4	2	3	3	1	2	3	1	1	3	1		
3	BAJO	2	2	2	1	2	1	1	1	1	1	2	1	1	2	1	1	1	1	1	1	1	2	2	1	
4	BERE	1	2	3	2	3	1	2	2	2	1	3	2	2	3	3	2	2	1	2	2	2	2	3		
5	BETH	2	2	4	2	3	1	2	2	4	1	4	3	1	3	4	3	2	2	3	3	3	2	4		
6	BIFR	1	2	2	1	2	1	2	1	1	1	1	1	1	3	1	1	2	1	2	1	2	1	2		
7	BIHO	3	3	3	3	3	1	1	1	1	3	2	1	3	3	1	1	3	2	1	2	1	5	1		
8	BITH	2	3	3	2	3	3		4	4	2	4	3	2	2	3	3	3	2	2	3	3	2	3		
9	BIUW	3	2	4	3	4	2	2	2	2	2	3	2	2	3	2	1	2	3	2	3	1	2	3		
10	CARU	4	2	4	4	5	2	3	2	4	2	5	4	2	3	3	2	2	1	3	5	4	2	4		
11	CHGE	3	3	3	3	2	1	1	1	1	1	1	1	1	1	2	2	2	2	3	1	1	1	1		
12	CHMI	2	2	3	2	4	2	2	3	5	2	5	5	2	3	5	3	2	1	1	5	3	2	4		
13	DAUW	2	2	3	3	4	2	2	2	3	2	4	3	2	3	2	1	2	1	2	3	2	2	3		
14	INWO	2	2	3	2	2	1	2	1	2	2	3	2	3	3	3	2	3	2	2	2	2	1	2		
15	JUKL	1	3	3	1	3	3	3	3	3	1	4	3	2	3	3	1	2	2	3	3	3	2	3		
16	LASA	1	1	3	1	3	1	1	1	2	1	2	1	3	3	1	1	1	1	1	1	2	1	3	2	
17	MAAC																									
18	REYV	4	2	3	2	3	1	2	2	3	1	3	2	1	2	2	2	2	2	2	3	3	2	3		
19	SADI	4	3	3	4	2	2	2	2	2	2	2	2	2	3	3	2	2	3	3	2	2	3	2		
20	SILO	1	2	2	1	3	1	1	1	3	1	3	3	1	3	1	2	1	1	1	3	2	1	4		
21	TAWA	1	1	1	1	3	1	2	2	2	2	4	3	2	3	3	3	2	1	1	2	3	2	3		
22	VEVA	4	2	2	4	2	2	2	2	2	3	3	3	2	3	3	2	3	2	1	3	4	3	4		

Durchschnitt: 2,3 2,2 2,8 2,2 2,9 1,6 1,9 1,9 2,6 1,7 3,0 2,3 2,0 2,7 2,5 1,9 2,1 1,6 2,0 2,5 2,3 2,1 2,7

Klasse: OG72 Datum: 11.03.2008  
 Lfd. Nummer: 06 Thema-Nummer: 03  
 Thema: Klassenimplementierung und Objekterzeugung

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANHA																									
2	ANST																									
3	BAJO	1	2	4	3	3	1	2	2	2	3	4	4	2	4	3	2	1	2	2	4	3	1	4	5	3
4	BERE	2	2	2	2	3	1	2	2	3	2	3	3	2	4	3	1	2	2	2	3	3	2	3	2	2
5	BETH	2	4	4	3	3	2	2	2	3	2	3	4	2	4	3	2	3	2	2	3	4	2	3	3	3
6	BIFR	1	2	1	2	2	1	1	1	1	1	2	3	1	3	2	1	1	2	2	2	2	2	3	1	1
7	BIHO	2	2	2	4	3	1	1	2	2	3	3	2	1	3	2	1	1	1	1	2	1	4	4	1	1
8	BITH	2	3	2	2	3	2	2	2	3	2	3	2	2	3	3	2	2	1	2	2	2	2	3	3	3
9	BIUW	3	4	4	3	3	2	2	3	3	2	3	4	3	3	4	3	3	4	3	4	2	3	3	2	2
10	CARU	4	1	3	2	3	1	1	1	2	1	3	2	1	2	2	1	1	1	2	2	3	1	1	1	1
11	CHGE																									
12	CHMI	2	3	3	2	3	2	2	2	3	1	3	2	1	3	1	2	2	3	2	2	2	1	3	3	3
13	DAUW	2	2	3	2	3	1	2	1	4	1	3	2	1	3	2	1	2	3	1	2	1	1	4	2	2
14	INWO	2	2	3	2	3	2	2	2	2	2	2	3	2	3	2	1	2	2		2	3	1	2	4	3
15	JUKL																									
16	LASA	1	1	2	1	3	1	1	2	3	1	3	2	2	3	2	1	1	2	2	3	1	1	3	3	2
17	MAAC	2	2	2	2	3	3	3	3	2	2	3	2	2	4	2	1	2	3	3	2	2	2	4	3	3
18	REYV																									
19	SADI																									
20	SILO	1	1	1	1	2	1	1	1	2	2	3	2	2	3	2	2	2	1	1	2	2	2	3	3	3
21	TAWA																									
22	VEVA	3	2	2	4	3	3	3	4	3	4	4	4	2	3	4	2	3	2	2	3	5	2	2	2	1

Durchschnitt: 2,0 2,2 2,5 2,3 2,9 1,6 1,8 2,0 2,5 1,9 3,0 2,7 1,7 3,2 2,5 1,5 1,9 2,1 1,9 2,5 2,4 1,8 3,0 2,5 2,2

Klasse: OG72 Datum: 24.04.2008  
 Lfd. Nummer: 07 Thema-Nummer: 04  
 Thema: Methoden

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANHA																									
2	ANST	3	3	3	2	3	3	2	2	2	2	2	3	2	3	3	2	2	2	2	3	3	2	3	3	3
3	BAJO	2	2	2	2	4	1	2	2	3	3	3	3	2	3	3	1	1	1	1	3	2	1	2	2	2
4	BERE	4	3	3	3	3	2	2	2	2	3	3	2	2	3	3	1	2	2	2	2	2	1	3	2	1
5	BETH	1	3	3	3	3	2	2	2	4	2	3	3	2	3	3	2	2	2	3	3	3	3	3	2	2
6	BIFR																									
7	BIHO	2	2	2	4	3	1	1	1	1	3	2	1	1	3	2	1	1	1	1	2	1	3	2	1	1
8	BITH	2	2	3	2	2	2	2	2	3	2	3	3	2	3	3	2	2	2	2	3	3	2	3	2	2
9	BIUW	2	1	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	CARU	4	3	4	3	4	2	2	2	3	2	3	2	2	3	3	2	2	2	2	3	2	2	3	2	2
11	CHGE																									
12	CHMI	3	2	3	2	3	2	2	2	5	1	5	3	2	3	3	3	2	2	2	3	2	1	4	3	3
13	DAUW	2	2	2	2	3	1	2	2	3	1	4	3	1	3	2	1	2	3	2	3	2	1	5	2	1
14	INWO	1	2	2	2	3	2	2	2	1	2	2	3	3	3	3	1	2	2	2	2	3	2	2	3	2
15	JUKL	2	2	2	1	3	2	3	3	3	1	3	3	2	3	2	3	2	2	3	3	3	2	3	3	3
16	LASA	1	1	3	1	3	1	1	2	3	1	3	2	2	3	1	2	1	3	2	2	1	2	3	3	2
17	MAAC	1	2	3	2	3	2	2	3	3	2	3	3	2	2	3	2	2	2	3	2	3	3	3	2	2
18	REYV																									
19	SADI	4	3	2	3	3	2		3	2	3	2	2	2	3	3	2	2	3		2	2	3	3	3	3
20	SIL0	1	1	1	1	2	1	1	2	3	1	3	3	2	3	2	1	1	1	2	3	3	2	2	2	1
21	TAWA																									
22	VEVA	3	3	2	4	3	2	3	3	2	2	4	4	3	4	4	2	2	2	3	3	5	2	1	2	2

Durchschnitt: 2,2 2,2 2,5 2,2 2,9 1,7 1,9 2,1 2,6 1,9 2,9 2,6 1,9 2,9 2,6 1,7 1,7 1,9 2,1 2,5 2,4 1,9 2,7 2,2 1,9

Klasse: OG72 Datum: 30.10.2007  
 Lfd. Nummer: 02 Thema-Nummer: 05  
 Thema: Kontrollstruktur: Selektion

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANHA	1	2	4	2	3	2	3	3	3	2	4	3	2	3	3	2	3	1	2	3	2	1	3	3	2
2	ANST	3	3	4	3	3	2	2	2	2	3	1	1	4	2	2	2	2	3	3	1	1	5	4	1	1
3	BAJO	2	2	3	2	3	2	1	1	1	1	2	2	1	3	2	1	3	2	1	1	2	4	2	1	3
4	BERE	1	1	2	2	3	1	1	1	2	1	2	2	1	3	1	1	2	1	1	2	2	2	3	1	2
5	BETH	2	3	3	2	3	2	2	2	3	2	3	3	2	3	3	3	2	2	2	3	2	2	3	3	3
6	BIFR	1	2	2	1	2	1	1	1	1	1	1	1	1	3	1	1	2	1	2	1	1	1	1	2	3
7	BIHO	3	2	3	4	3	1	2	2	2	2	3	1	3	2	2	1	3	2	1	1	1	5	2	5	1
8	BITH																									
9	BIUW	1	3	3	2	4	1	2	2	2	2	3	3	2	2	2	1	2	3	2	2	2	2	2	3	2
10	CARU	4	1	3	1	2	1	2	2	3	1	3	2	1	3	2	2	1	1	2	2	2	1	2	2	2
11	CHGE																									
12	CHMI	2	3	2	2	3	2	2	2	5	4	4	3	3	3	3	2	2	2	1	3	2	2	2	3	5
13	DAUW	2	2	4	3	4	2	2	2	3	2	4	4	1	3	2	1	2	1	1	4	3	2	3	1	1
14	INWO																									
15	JUKL	1	2	3	1	3	2	2	2	4	1	4	3	2	3	3	1	2	2	1	4	3	1	4	2	1
16	LASA																									
17	MAAC	3	2	3	2	3	2	3	3	2	2	2	3	2	3	3	2	2	3	3	3	2	2	3	3	3
18	REYV																									
19	SADI	4	3	2	3	2	2		2	2	3	2	2	3	4	3	2	2	3		2	2	4	2	3	4
20	SIL0	1	2	3	1	3	1	1	1	2	1	3	2	1	3	2	2	1	1	2	3	2	2	3	2	2
21	TAWA	1	1	2	2	3	1	2	2	3	2	4	4	2	3	3	2	2	1	1	3	3	1	3	2	2
22	VEVA																									

Durchschnitt: 2,0 2,1 2,9 2,1 2,9 1,6 1,9 1,9 2,5 1,9 2,8 2,4 1,9 2,9 2,3 1,6 2,1 1,8 1,7 2,4 2,0 2,3 2,6 2,3 2,3

Klasse: OG72 Datum: 22.11.2007  
 Lfd. Nummer: 03 Thema-Nummer: 06  
 Thema: Kontrollstruktur: Iteration

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANHA	2	2	4	2	3	2	2	3	3	2	4	3	2	3	3	2	2	1	2	3	3	2	3	3	3
2	ANST	3	3	4	2	3	3	2	2	2	2	2	2	3	2	3	1	2	2	3	2	2	5	2	1	2
3	BAJO	1	2	3	2	3	1	1	1	2	1	2	2	1	4	2	1	2	3	2	2	1	2	2	1	1
4	BERE	1	1	2	2	3	2	2	1	2	2	3	3	2	2	2	1	1	2	1	2	2	1	2	2	2
5	BETH	2	3	4	2	4	2	2	2	3	2	4	3	2	3	2	2	2	2	2	3	3	2	3	3	3
6	BIFR	4	2	1	1	2	1	1	1	1	1	1	1	1	3	1	2	1	1	2	1	1	2	3	1	3
7	BIHO	3	3	4	5	4	2	1	2	4	4	4	2	1	3	2	1	3	2	2	3	1	5	3	3	1
8	BITH																									
9	BIUW	3	3	4	3	3	2	2	3	2	2	3	2	3	3	3	1	2	2	3	3	2	2	3	3	2
10	CARU	4	2	3	2	3	1	1	1	3	1	3	3	1	2	2	2	1	1	1	2	2	1	2	1	1
11	CHGE																									
12	CHMI																									
13	DAUW	2	2	4	2	4	2	2	2	3	2	4	4	2	3	2	1	3	2	2	4	2	2	3	2	2
14	INWO	1	2	2	1	3	2	2	2	2	2	3	2	2	3	3	1	2	2		2	3	2	2	2	1
15	JUKL	2	2	2	1	3	2	2	2	3	1	3	2	1	2	2	2	3	1	3	3	3	2	2	3	3
16	LASA	1	2	2	2	3	1	1	1	2	2	3	2	2	3	2	1	1	2	2	3	1	2	3	3	2
17	MAAC	3	2	2	3	3	2	2	3	2	2	3	2	2	3	3	3	3	2	3	2	3	3	3	2	2
18	REYV	4	3	3	3	2	3	3	4	5	4	4	4	3	3	4	2	4	3	4	5	3	3	4	3	3
19	SADI																									
20	SILO	1	1	3	2	3	1	1	2	3	1	3	3	2	3	2	1	2	1	1	3	2	2	3	3	2
21	TAWA	1	2	3	2	3	2	2	3	3	2	4	3	2	3	4	2	1	1	2	3	3	2	2	3	3
22	VEVA																									

Durchschnitt: 2,2 2,2 2,9 2,2 3,1 1,8 1,7 2,1 2,6 1,9 3,1 2,5 1,9 2,8 2,5 1,5 2,1 1,8 2,2 2,7 2,2 2,4 2,6 2,3 2,1

Klasse: OG72 Datum: 20.12.2007  
 Lfd. Nummer: 04 Thema-Nummer: 07  
 Thema: Array und Strings

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
1	ANHA	2	3	3	3	3	2	2	2	3	2	3	3	2	3	3	3	1	2	2	3	3	2	3	3	2	
2	ANST																										
3	BAJO	2	2	3	3	4	1	1	2	2	2	3	3	1	2	2	1	2	2	1	2	2	2	3	3	3	
4	BERE	2	2	2	2	3	2	2	2	2	2	3	2	1	2	2	1	1	2	2	2	3	2	3	1	1	
5	BETH	2	3	2	3	3	2	2	2	4	2	4	3	2	3	3	2	2	2	2	4	3	2	3	3	3	
6	BIFR	2	2	2	2	3	1	1	1	1	1	1	2	1	3	2	2	1	2	1	1	1	2	2	1	1	
7	BIHO	2	2	3	3	3	2	2	2	2	5	2	2	2	2	2	2	2	2	2	2	2	1	3	3	2	1
8	BITH	1	3	3	2	3	3	2	3	2	2	4	3	2	2	2	2	2	2	2	3	2	3	3	2	3	
9	BIUW	2	3	4	3	4	2	2	2	2	1	2	2	1	2	2	1	2	3	2	3	2	1	3	2	1	
10	CARU	2	1	2	1	3	1	1	1	2	1	3	1	1	2	1	2	1	1	2	3	2	1	3	1	2	
11	CHGE	2	2	3	2	3	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	1	4	1	2	2
12	CHMI	1	3	3	3	4	3	2	4	5	1	5	3	3	3	3	3	3	3	3	4	3	1	4	3	3	
13	DAUW	2	3	3	3	4	2	2	2	3	2	4	4	2	3	2	2	2	2	2	4	3	3	4	1	1	
14	INWO	2	2	2	3	3	1	2	1	1	2	2	2	2	2	2	1	2	2		2	2	1	1	2	3	
15	JUKL	2	2	2	1	2	2	2	4	1	4	3	2	2	2	2	2	2	2	1	2	2	1	4	3	3	
16	LASA	1	2	2	2	3	1	1	2	3	1	3	2	2	3	2	1	1	2	1	3	1	2	3	3	2	
17	MAAC	2	2	2	3	4	2	3	3	3	3	3	3	2	3	2	2	3	3	3	3	2	2	3	3	2	
18	REYV																										
19	SADI	4	3	2	3	3	2		2	3	3	3	3	3	2	3	2	2	2		2	3	2	2	2	2	
20	SILO	1	2	1	1	2	1	1	1	3	1	3	3	2	3	3	2	1	1	2	3	2	2	4	3	3	
21	TAWA																										
22	VEVA																										

Durchschnitt: 1,9 2,3 2,4 2,4 3,2 1,7 1,7 1,9 2,6 1,8 2,9 2,5 1,8 2,4 2,2 1,8 1,8 2,1 1,9 2,6 2,1 2,0 2,9 2,2 2,1

Klasse: OG72 Datum: 29.05.2008  
 Lfd. Nummer: 08 Thema-Nummer: 08  
 Thema: Vererbung

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANHA	3	3	4	3	3	2	1	3	4	3	3	4	3	3	3	3	2	2	2	3	4	3	3	2	2
2	ANST	4	3	4	2	3	3	2	3	3	2	3	3	2	3	4	2	2	3	3	3	3	3	2	4	3
3	BAJO	2	4	2	3	3	4	1	2	2	4	3	4	1	2	2	1	4	4	4	1	1	4	5	1	
4	BERE	4	3	3	2	3	2	2	3	3	2	3	3	2	3	3	1	2	2	2	3	2	2	3	1	1
5	BETH	2	3	3	3	3	2	2	2	3	2	3	3	2	3	2	2	2	2	2	3	2	2	3	2	3
6	BIFR	1	2	3	2	3	1	2	1	2	1	2	3	2	3	3	1	2	3	2	3	2	3	3	2	2
7	BIHO	1	2	2	4	3	1	1	1	2	4	2	2	1	2	1	1	1	3	1	2	1	5	3	1	1
8	BITH																									
9	BIUW	2	2	3	1	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	2	2	1	1
10	CARU	1	1	2	3	4	1	2	1	3	2	3	3	2	2	3	2	2	2	2	3	4	3	4	2	2
11	CHGE	4	3	3	3	4	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	4	1	4	3
12	CHMI	4	3	3	2	3	2	2	2	4	2	4	4	2	3	3	2	2	2	2	4	2	1	4	3	3
13	DAUW	2	2	3	2	3	1	1	2	3	1	3	3	1	3	2	1	2	3	2	3	2	1	5	2	1
14	INWO																									
15	JUKL	2	2	4	1	3	2	2	3	3	1	3	3	2	3	3	1	3	3	3	3	3	1	3	3	2
16	LASA	3	3	4	2	4	2	2	3	3	2	2	3	3	3	2	2	2	3	2	3	1	3	3	3	2
17	MAAC	2	3	3	2	3	2	2	2	3	2	3	2	2	3	2	2	3	3	3	2	2	2	3	2	2
18	REYV																									
19	SADI																									
20	SILO	1	1	2	2	2	1	1	2	3	2	3	3	1	3	2	1	2	2	2	3	2	2	3	2	2
21	TAWA	1	2	2	2	2	2	2	2	3	2	3	4	2	3	3	1	1	1	2	3	3	2	2	1	2
22	VEVA	3	2	1	4	3	2	2	3	4	3	4	4	2	4	4	2	2	1	3	3	5	2	1	2	2

Durchschnitt: 2,3 2,4 2,8 2,4 3,0 1,8 1,6 2,1 2,8 2,1 2,8 2,9 1,8 2,7 2,4 1,5 2,0 2,3 2,1 2,8 2,3 2,3 2,9 2,3 1,9


Klasse: OG72 Datum: 19.06.2008  
 Lfd. Nummer: 09 Thema-Nummer: 09  
 Thema: Assoziation

	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	ANHA	2	3	3	3	3	3	2	2	3	3	4	3	3	3	3	3	2	1	2	3	4	3	3	2	1
2	ANST	4	3	3	2	3	2	2	3	2	3	3	2	2	2	2	2	2	2	2	2	3	3	3	2	2
3	BAJO	2	2	2	4	3	1	2	1	2	4	3	3	4	2	2	1	1	1	1	1	2	1	1	1	1
4	BERE																									
5	BETH	2	2	4	3	4	2	2	2	4	2	4	4	2	3	3	2	3	2	2	4	3	2	4	2	3
6	BIFR	1	2	2	2	2	1	1	1	2	1	2	2	1	3	2	1	2	2	1	2	2	2	3	1	1
7	BIHO	1	3	3	5	2	1	1	1	1	5	2	1	1	4	2	1	1	2	1	2	2	4	3	1	1
8	BITH	2	3	4	2	3	2	2	3	3	2	3	3	3	4	3	2	3	2	2	3	3	3	3	3	2
9	BIUW	2	3	3	2	3	2	3	2	2	2	3	4	3	3	4	2	3	4	3	4	3	3	2	2	1
10	CARU	4	3	4	2	4	2	3	3	4	2	4	5	3	4	4	2	3	4	4	4	4	4	4	2	2
11	CHGE																									
12	CHMI	2	2	3	2	3	2	2	2	4	2	5	3	3	3	3	3	2	2	2	3	2	1	4	2	2
13	DAUW	2	2	3	3	3	2	2	3	4	2	4	5	3	5	3	2	3	3	2	4	4	3	3	2	2
14	INWO	1	2	3	3	2	2	2	2	2	2	2	3	2	3	3	1	2	2	2	2	2	1	2	2	1
15	JUKL	3	3	3	2	3	2	3	3	1	3	3	3	2	3	1	3	3	3	3	3	4	2	3	3	3
16	LASA	3	3	4	2	3	2	2	3	3	2	3	3	2	2	2	2	2	3	2	2	2	2	3	3	2
17	MAAC	3	2	3	4	3	1	2	2	3	2	3	4	3	2	4	2	4	3	2	3	3	3	3	2	2
18	REYV	5	4	5	3	4	3	3	4	5	2	5	3	2	3	4	3	3	2	4	4	4	3	4	2	2
19	SADI	4	3	3	4	2	2	2	2	3	3	3	2	3	3	2	2	2	2	2	3	3	2	3	2	2
20	SILO	1	1	2	3	3	1	1	2	3	2	4	3	2	2	2	1	2	1	1	3	3	2	2	2	1
21	TAWA	1	1	2	2	2	2	2	2	3	2	3	2	3	3	2	2	1	2	2	3	2	2	3	1	2
22	VEVA																									

Durchschnitt: 2,4 2,5 3,1 2,8 2,9 1,8 2,1 2,2 2,9 2,3 3,3 3,1 2,5 3,0 2,8 1,8 2,3 2,3 2,1 2,9 2,9 2,4 2,9 1,9 1,7

## M) Vergleichstest (Post-Test) inkl. Löser (Hauptstudie 2007/2008)

Hinweis: Ohne die roten Lösungshinweise ist dies das Original des Vergleichstests (Post-Test)!

	<b>Lernerfolgskontrolle 2008 (inkl. Löser)</b>	Informatik Einführungsphase
	Klasse:	Datum:
		Blatt Nr.: 1 / 1

	Ihr Code
<b>Erster Buchstabe des Vornamens der Mutter</b>	
<b>Zweiter Buchstabe des Vornamens der Mutter</b>	
<b>Erster Buchstabe des Vornamens des Vaters</b>	
<b>Zweiter Buchstabe des Vornamens des Vaters</b>	

In der folgenden Lernerfolgskontrolle werden Fragen zu 9 Themengebieten gestellt.

Zu jedem dieser Themengebiete gibt es in der Regel ein Blatt mit einer Mischung aus Ankreuzaufgaben und Fragen, die einer eigenen Lösung bedürfen. In jedem Themengebiet können 10 Punkte erreicht werden.

**Hinweis: Bei den Multiple-Choice-Fragen sind evt. beliebig viele Antworten (aber mindestens eine) richtig!** Fehlerhafte Kreuze führen zu Punktabzug.

Ein paar Tage vor oder nach diesem Test erhalten Sie noch Zusatzfragen zu einem zehnten und elften Themengebiet. Die dort erzielten Punkte werden zu den anderen Punkten hinzuaddiert!

Nr.	Themengebiet	Erreichbare Punkte	Erreichte Punkte
1	Datentypen und Steuerstruktur Sequenz	10	
2	Steuerstruktur: Selektion	10	
3	Steuerstruktur: Iteration	10	
4	Arrays und Strings	10	
5	Methoden	10	
6 / 7	Einführung in die OOP / Klasse und Objekt	20	
8	Vererbung	10	
9	Assoziation	10	
10	Zusatz: dynamische OOP	10	
11	Zusatz: Objektorientierte Modellierung (OOM)	10	
Summe		110	

Ehler 2008 [ehler@oszimt.de](mailto:ehler@oszimt.de) [www.oszimt.de](http://www.oszimt.de)

1. Datentypen und Steuerstruktur Sequenz

1.	Durch den <b>Datentyp</b> einer Variablen wird festgelegt, welche(n) ...	<input checked="" type="checkbox"/> Werte die Variable annehmen kann <input type="checkbox"/> Namen die Variable hat <input type="checkbox"/> Adresse die Variable im Speicher hat <input type="checkbox"/> Wert die Variable hat <input checked="" type="checkbox"/> Anzahl von Bits die Variable im Speicher belegt	1P
2.	Gegeben sind die beiden Quelltextzeilen: <pre>final double PI = 3; PI = 3.14;</pre> Welche Aussage(n) trifft/treffen zu?	<input type="checkbox"/> Compilerfehler in Zeile 1 <input checked="" type="checkbox"/> Compilerfehler in Zeile 2 <input type="checkbox"/> PI ist eine Variable <input type="checkbox"/> PI muss klein geschrieben werden <input type="checkbox"/> 3,14 statt 3.14	1P
3.	Welche Werte kann eine Variable vom <b>Datentyp byte</b> annehmen?	<input type="checkbox"/> 0 bis 255 <input type="checkbox"/> 0 bis 65535 <input checked="" type="checkbox"/> -128 bis 127 <input type="checkbox"/> -127 bis 128 <input type="checkbox"/> Der Wert ist hardwareabhängig	1P
4.	Gegeben sind die beiden Quelltextzeilen: <pre>int a = 7; short b = a;</pre> Welche Aussage(n) trifft/treffen zu?	<input checked="" type="checkbox"/> Es gibt einen Compilerfehler <input type="checkbox"/> Beide Variablen benötigen den gleichen Speicherplatz <input type="checkbox"/> a belegt 16 Bit <input type="checkbox"/> b belegt 8 Bit	1P

5. Gegeben ist die folgende lückenhafte **Variablendeklaration**:

3P

```
char    a = __'a'____;
float__ b = 7.0f;
boolean_ c = (5 == 7);
```

3 \* 1P

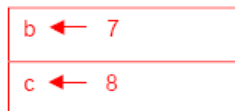
**Ergänzen** Sie den **Quelltext** in den 3 Lücken!

6. Gegeben sind die drei Quelltextzeilen:

```
int a, b;
b = 7;
c = 8;
```

**Zeichnen** Sie das dazugehörige **Struktogramm**!

3P



2 \* 1,5P



2. Steuerstruktur: Selektion

1.	Gegeben ist der folgende Vergleich: if (wahrheit == false) {} Die verkürzte(n) Schreibweise(n) ist/sind wie folgt:	<input type="radio"/> if (wahrheit) {} <input type="radio"/> if (wahrheit <> true) {} <input checked="" type="radio"/> if (!wahrheit) {} <input type="radio"/> if (wahrheit!) {}	1P
2.	<pre>int b = 0, c = 7; if (c &lt; 7)   if (c &gt;= 7) b = 1;   else      b = 2; else   if (c &gt; 7) b = 3;   else      b = 4; </pre> Was steht am Ende in der Variablen b?	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4	1P

3. Gegeben ist der folgende Quelltextausschnitt:

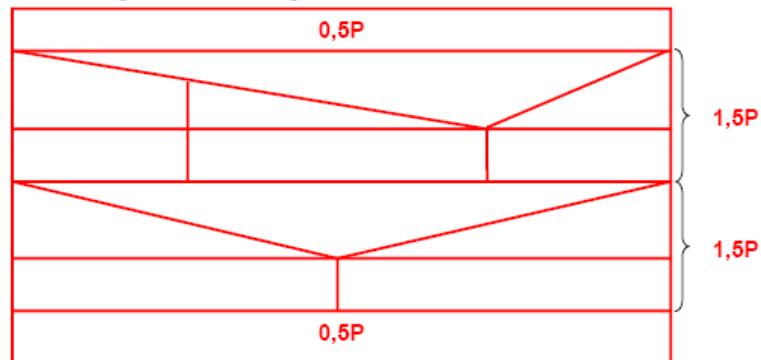
```
int a = 5;           // oder 7 oder 9
switch (a) {
  case 5:
    a = 3 * a;
    break;
  case 9:
    a = a / 2;
    break;
  default:
    a = a % 3;      // modulo
    break;
} // switch
if (a >= 15)
  a--;
else
  a++;
System.out.println("a: " + a);
```

\*5,5 und 6 (0,5P)

a	Ausgabe
5	14 (1P)
7	2 (1,5P)
9	5 (1,5P)*

a) Welcher Wert wird ausgegeben, bei den folgenden ursprünglichen Werten für die Variable a (3 Tabelleneinträge)? 4P

b) Zeichnen Sie das dazugehörige Struktogramm!  
Hinweis: Einträge in das Struktogramm sind nicht erforderlich! 4P



Klasse:

Datum:

Blatt Nr.: 4 / 4

### 3. Steuerstruktur: Iteration

1.	Gegeben ist die folgende Quelltextzeile: <code>for (int i = 1; i &lt; 7; i++) {}</code> Diese <b>Schleife</b> wird im <b>Struktogramm</b> mit Hilfe welchen Strukturelements nach DIN 66261 dargestellt?	<input checked="" type="radio"/> kopfgesteuerte Schleife <input type="radio"/> fußgesteuerte Schleife <input type="radio"/> Endlosschleife <input type="radio"/> keiner der genannten Schleifen	1P
2.	Welche Antwort(en) enthält/enthalten eine Endlosschleife?	<input checked="" type="radio"/> <code>for ( ; ; ) {}</code> <input checked="" type="radio"/> <code>do { } while (!false);</code> <input type="radio"/> <code>while (!true) { }</code> <input checked="" type="radio"/> <code>int y = 7;</code> <code>while (y == 7) { }</code>	1P

3. Gegeben ist der folgende Quelltextausschnitt:

```
a = 3 * 8;
b = a / 4;
while (b > 1) {
    a = a - 2;
    b = b - 1;
}
```

Welcher Wert steht am Ende des Algorithmus in der Variable a?

a  
14

2P

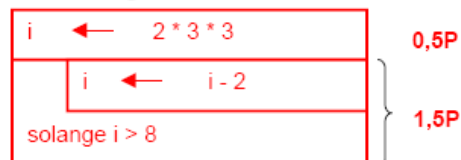
4. Gegeben ist der folgende Quelltextausschnitt:

```
i = (2 * 3) * 3;
do {
    i = i - 2;
} while (i > 8);
```

Wie oft wird der Schleifenkörper durchlaufen?  
Zeichnen Sie das **Struktogramm**!

Anzahl  
5

2P  
2P



5. Ergänzen Sie den folgenden Quelltext, so dass als Ausgabe: 10,8,6,4, erscheint! 2P

```

0,5P      0,5P      0,5P      0,5P
for (int z = 10; z >= 4; z = z - 2)
    System.out.print(z + ',');
```

**4. Arrays und Strings**

1.	Gegeben ist die folgende Variablendeklaration: String s = "Test"; String t = new String("Test"); Welche Bedingungen liefern true zurück?	<input checked="" type="radio"/> if (s == "Test") {} <input type="radio"/> if (t == s) {} <input checked="" type="radio"/> if (s.equals("Test")) {} <input checked="" type="radio"/> if (t.equals(s)) {}	<b>1P</b>
2.	int[] x = new int[25]; Welche Aussage(n) trifft/treffen nach der Ausführung der Programmzeile zu?	<input type="radio"/> x.length hat den Wert 24 <input type="radio"/> x[0] hat den Wert null <input type="radio"/> x[25] hat den Wert 0 <input type="radio"/> x[24] ist nicht definiert <input checked="" type="radio"/> keine Aussage trifft zu	<b>1P</b>

3. Gegeben ist der folgende Quelltextausschnitt: **2P**

```
int[] exampleArray = {10, 20, 30, 40};
System.out.println(exampleArray[2*2]*3);
```

Wie lautet die Bildschirmausgabe?

**(ArrayIndexOutOfBoundsException-) Exception**

4. Initialisieren Sie das folgende Feld mit Hilfe einer Zählschleife mit den Werten 15, 35 und 55! **3P**

```
short[] zahlenfeld = new short[3];
```

```
for (int i = 0; i <= 2; i++)          1P
    zahlenfeld[i] = 15 + i*20;        1P
```

5. Gegeben ist der folgende Quelltext: **3P**

```
String s = new String("HALLO");
s.toLowerCase();
System.out.println(s);
s = "kleine";
s += " Zwerge";
System.out.println(s.toUpperCase());
System.out.println(5*3 + 7 + " 5*3 Tore " + 3*5);
```

Wie lautet die Bildschirmausgabe?

```
HALLO          1P      hallo          0,5P
KLEINE ZWERGE 1P      kleineZwerge 0,5P
22 5*3 Tore 15 1P
```

### 5. Methoden

1.	Beim <b>Aufruf einer Methode</b> findet man <b>nie</b> ...	<input type="radio"/> einen aktuellen Parameter <input checked="" type="radio"/> einen formalen Parameter <input type="radio"/> einen Rückgabewert	<b>1P</b>
2.	Welche Aussage(n) trifft/treffen für die folgende <b>Quelltextzeile</b> zu: <code>if (isGeldAlle()) {}</code>	<input checked="" type="radio"/> Die Methode hat einen Boolean-Rückgabewert <input type="radio"/> Die Methode hat keinen Rückgabewert <input type="radio"/> Der Methode fehlt ein Parameter	<b>1P</b>
3.	Gegeben ist der folgende Quelltextauszug: <code>String s; s = liefereEtwas(4.7);</code> Welche Aussage(n) ist/sind richtig?	<input checked="" type="radio"/> Die Methode kann statisch sein <input checked="" type="radio"/> Die Methode hat einen Rückgabewert <input type="radio"/> s muss vom Datentyp <code>double</code> sein <input type="radio"/> Vor dem Wert 4.7 fehlt der Datentyp <code>double</code>	<b>1P</b>

4. Gegeben sind die folgenden Quelltextzeilen des Anwendungsprogramms, die in der main-Methode stehen: **3P**
- ```
int[] zahlenfeld = new int[10];
initialisiereFeld(zahlenfeld);
kleinsteZahl = ermittleKleinsteZahl(zahlenfeld);
```
- Wie lautet der **Methodenkopf** der statischen Methode „ermittleKleinsteZahl(...)“, welcher im Anwendungsprogramm „über“ der main-Methode steht?
- Hinweis:** Nur **eine Quelltextzeile** ist gefragt!
- 0,5P   0,5P   0,5P   0,5P   0,5P   0,5P**
- ```
public static int ermittleKleinsteZahl(int[] feld)
```

5. In der Klasse „Konto“ soll eine Methode „zahleEin“ implementiert werden, die als formalen Parameter einen Double-Geldbetrag hat und zurückliefert, ob das Konto nach der Einzahlung gedeckt ist (Attribut `kontostand > 0` bedeutet: Rückgabewert ist „true“). **4P**
- Geben Sie den vollständigen Quelltext der Methode „zahleEin“ (Methodenkopf und Implementierung) an!**
- 0,5P   0,5P   0,5P   0,5P**
- ```
public boolean zahleEin(double geldbetrag) {
    kontostand += geldbetrag;
    if (kontostand > 0)
        return true;
    return false;
}
```
- 0,5P   0,5P   1P   0,5P**

6. Einführung in die OOP / 7. Klasse und Objekt

|    |                                                                          |                                                                                                                                                                                                                                                                                                                                                                      |
|----|--------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Welche Aussage(n) über <b>Klassen</b> sind falsch?                       | <input checked="" type="checkbox"/> Eine Klasse ist in einem bestimmten Zustand<br><input checked="" type="checkbox"/> Eine Klasse ist die Summe der Objekte<br><input checked="" type="checkbox"/> Eine Klasse hat immer Attribute <b>1P</b>                                                                                                                        |
| 2. | Welche Aussage(n) über <b>Objekte</b> ist/sind richtig?                  | <input type="checkbox"/> Von jeder Klasse gibt es mindestens ein Objekt<br><input checked="" type="checkbox"/> Objekte können mit anderen Objekten Botschaften austauschen<br><input checked="" type="checkbox"/> Objekte befinden sich in einem Zustand<br><input type="checkbox"/> Objekte haben keinen Zugriff auf sich selbst <b>1P</b>                          |
| 3. | Welche Aussage(n) zu <b>Attributen</b> ist/sind richtig?                 | <input checked="" type="checkbox"/> Attribute sind vergleichbar mit Variablen<br><input type="checkbox"/> Ein Objekt deklariert Attribute<br><input type="checkbox"/> In einer Klasse haben die Attribute Werte<br><input type="checkbox"/> Attribute haben keinen Datentyp<br><input checked="" type="checkbox"/> Attribute werden in der Regel gekapselt <b>1P</b> |
| 4. | <b>Methoden</b> werden im UML-Objektdiagramm ...                         | <input type="checkbox"/> angegeben<br><input checked="" type="checkbox"/> nicht angegeben<br><input type="checkbox"/> je nach Bedarf angegeben <b>1P</b>                                                                                                                                                                                                             |
| 5. | Wozu braucht man die <b>Verwaltungsmethoden</b> („Setter“ und „Getter“)? | <input type="checkbox"/> Weil diese immer deklariert werden müssen<br><input type="checkbox"/> Weil die Attribute in der Regel öffentlich sind<br><input checked="" type="checkbox"/> Weil die Attribute in der Regel privat sind <b>1P</b>                                                                                                                          |
| 6. | Wozu dienen <b>Konstruktoren</b> ?                                       | <input type="checkbox"/> Um eine Vererbungshierarchie zu ermöglichen<br><input type="checkbox"/> Um eine Klasse zu initialisieren<br><input checked="" type="checkbox"/> Um ein Objekt zu initialisieren<br><input type="checkbox"/> Um ein Klasse deklarieren zu können <b>1P</b>                                                                                   |

7. Die Klasse „Fahrrad“ hat die Attribute „groesse“ (Datentyp int) und „hersteller“ (Datentyp String). **2P**

Wie lautet der **Methodenkopf** des **vollparametrisierten Konstruktors**?

**Hinweis:** Nur **eine Quelltextzeile** ist gefragt!

**1P**

**1P**

```
public Fahrrad(int groesse, String hersteller)
```

8. Die Klasse „Reisebuero“ hat u. a. das Attribut „anzahlKunden“ vom Datentyp „int“. **3P**  
Wie lautet der vollständige Quelltext der „**Setter**“-**Verwaltungsmethode**?

**0,5P**

**0,5P**

**0,5P**

**0,5P**

```
public void setAnzahlKunden(int anzahlKunden) {
    this.anzahlKunden = anzahlKunden; 1P
}
```

9. Die Klasse „Mensch“ hat das Attribut „vorname“ vom Datentyp „String“. 2P  
Wie lautet der vollständige Quelltext der „**Getter**“-Verwaltungsmethode?

```

    0,5P      0,5P
    public String getVorname() {
        return this.vorname;      1P
    }
  
```






10. Die Klasse „Surfbrett“ hat das Attribut „laenge“ (Datentyp double). 4,5P  
Zeichnen Sie das vollständige **UML-ODD-Klassendiagramm** inklusive dem parameterlosen und dem parametrisierten Konstruktor und allen Verwaltungsoperationen („Setter“ und „Getter“)!

|                           |             |
|---------------------------|-------------|
| <b>Surfbrett</b>          | <b>0,5P</b> |
| - laenge: double          | 1P          |
| + Surfbrett()             | 0,5P        |
| + Surfbrett(double)       | 0,5P        |
| + setLaenge(double): void | 1P          |
| + getLaenge(): double     | 1P          |

11. Jemand sagt: „Die Katze dort hinten ist 3 Jahre alt und schwarz.“ 2,5P  
Zeichnen Sie das **UML-Objektdiagramm** dieser Katze!

|                        |      |
|------------------------|------|
| <u>dieKatze: Katze</u> | 1,5P |
| alter = 3              | 0,5P |
| farbe = schwarz        | 0,5P |

### 8. Vererbung

|    |                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1. | Eine Oberklasse hat <b>2 Attribute</b> .<br>Wie viele Attributswerte hat ein Objekt der Unterklasse?                                                                                                | <input type="radio"/> maximal 2<br><input type="radio"/> genau 2<br><input checked="" type="radio"/> mindestens 2<br><input type="radio"/> kann man so nicht sagen                                                                                                                                                                                                                                                                                        | 1P |
| 2. | Welche Aussage(n) ist/sind richtig?<br>Eine <b>abstrakte Klasse</b> ...                                                                                                                             | <input type="radio"/> enthält immer eine abstrakte Methode<br><input checked="" type="radio"/> kann nicht instanziiert werden<br><input type="radio"/> hat keine Attribute                                                                                                                                                                                                                                                                                | 1P |
| 3. | Welcher „ <b>Vererbungs</b> pfeil“ ist im UML-Klassendiagramm anstelle der gestrichelten Linie vorzusehen?<br><br> | <input type="radio"/> <br><input type="radio"/> <br><input checked="" type="radio"/> <br><input type="radio"/>  | 1P |

4. Gegeben ist ein Quelltextausschnitt der folgenden Oberklasse:

7P

```
public abstract class Oberklasse {
    private int attribut1;
    // Im folgenden sind noch der parametrisierte Konstruktor und
    // die Set- und Get-Methode für das Attribut deklariert!!!
    public abstract void setzeAttribut1AufNull();
}
```

**Schreiben** Sie einen Teil des **Quelltextes** der Klasse „**Unterklasse**“: Sehen Sie ein gekapseltes **attribut2** (Datentyp: String), einen **vollparametrisierten Konstruktor** und die Implementierung der **abstrakten Methode** der Oberklasse vor!  
„Setter“/„Getter“ und den parameterlosen Konstruktor können Sie dagegen weglassen!

```

0,5P                1P
public class Unterklasse extends Oberklasse {

    private String attribut2;                1P

    public Unterklasse(int attribut1, String attribut2) {    1P
        super(attribut1);                                1P
        this.attribut2 = attribut2;                    0,5P
    }

    public void setzeAttribut1AufNull() {                1P
        setAttribut1(0);                                1P
    }

}

```

9. Assoziation (inkl. Klassenattribute und Klassenmethoden)

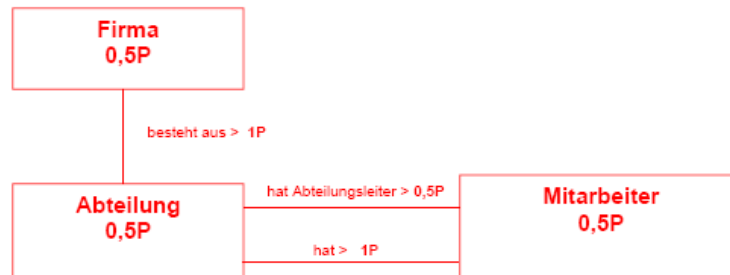
|    |                                                                                       |                                                                                                                                                                                                                                                                                                  |
|----|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Klassenattribute ...                                                                  | <input checked="" type="checkbox"/> sind immer Teil einer Klasse<br><input type="checkbox"/> sind den einzelnen Objekten zugeordnet<br><input checked="" type="checkbox"/> besitzen einen Wert <b>1P</b>                                                                                         |
| 2. | Welche Aussage(n) über <b>Klassenmethoden</b> (statische Methoden) trifft/treffen zu? | <input type="checkbox"/> Sie besitzen immer ein Klassenattribut<br><input type="checkbox"/> Sie haben nie einen Rückgabewert<br><input checked="" type="checkbox"/> Aufruf erfolgt in der Regel mit:<br>KlasseName.methodenname ()<br><input type="checkbox"/> Keine Aussage trifft zu <b>1P</b> |

3. Gegeben ist das folgende Szenario: Eine Firma besteht aus mehreren Abteilungen. Jede Abteilung hat genau einen Mitarbeiter als Abteilungsleiter und viele andere Mitarbeiter.

a) Zeichnen Sie das **UML-Klassendiagramm!**

4P

**Hinweis:** Attribute und Methoden sind **nicht** erforderlich



b) Ergänzen Sie den folgenden **Quelltext**, so dass die **Assoziationen** zwischen einer Abteilung und den Mitarbeitern **implementiert** sind! **Hinweis:** Durch den Konstruktor erhalten die privaten Attribute die entsprechenden Referenzen.

4P

```

public class Mitarbeiter { ... } // Klasse ist gegeben

public class Abteilung {
    → ___private Mitarbeiter abteilungsleiter; ___
    → ___private Mitarbeiter[] mitarbeiterListe; ___
    public Abteilung(Mitarbeiter abteilungsleiter,
                    Mitarbeiter[] mitarbeiterListe) {
    → ___this.abteilungsleiter = abteilungsleiter; ___
    → ___this.mitarbeiterListe = mitarbeiterListe; ___
    } // Konstruktor
} // Abteilung
  
```

4\* 1P



**10. Objektorientierte Programmierung** (dynamisch):

Gegeben ist der folgende Quelltext (zur Übersichtlichkeit wurden die Klassen Mensch und Fahrrad in das Anwendungsprogramm integriert):

```
public class DynamischeOOP {
```

```
    public static class Mensch {
```

```
        private String name;
        private Mensch ehpartner;
        private int anzahlFahrraeder;
        private Fahrrad[] fahrraeder;
```

```
        public Mensch() {
            this("Janine");
        }
```

```
        public Mensch(String name) {
            this.name = name;
            this.anzahlFahrraeder = 0;
            fahrraeder = new Fahrrad[100];
        }
```

```
        public void setName(String name) {
            this.name = name;
        }
```

```
        public String getName() {
            return this.name;
        }
```

```
        public int getAnzahlFahrraeder() {
            return this.anzahlFahrraeder;
        }
```

```
        public void heirate(Mensch einMensch) {
            ehpartner = einMensch;
        }
```

```
        public void kaufe(Fahrrad einFahrrad) {
            this.fahrraeder[anzahlFahrraeder] = einFahrrad;
            this.anzahlFahrraeder++;
        }
```

```
        public Mensch getEhpartner() {
            return this.ehpartner;
        }
```

```
        public Fahrrad getFahrrad(int i) {
            return fahrraeder[i];
        }
```

```
    } // Klasse Mensch
```

|                                                          | Ihr Code |
|----------------------------------------------------------|----------|
| <b>Erster</b> Buchstabe des Vornamens der <b>Mutter</b>  |          |
| <b>Zweiter</b> Buchstabe des Vornamens der <b>Mutter</b> |          |
| <b>Erster</b> Buchstabe des Vornamens des <b>Vaters</b>  |          |
| <b>Zweiter</b> Buchstabe des Vornamens des <b>Vaters</b> |          |

Klasse:

Datum:

Blatt Nr.: 12 / 12

```
public static class Fahrrad {  
  
    private String farbe;  
  
    public Fahrrad() {  
        this.farbe = ("unbekannt");  
    }  
  
    public Fahrrad(String farbe) {  
        this.farbe = farbe;  
    }  
  
    public void setFarbe(String farbe) {  
        this.farbe = farbe;  
    }  
  
    public String getFarbe() {  
        return this.farbe;  
    }  
  
} // Klasse Fahrrad  
  
public static void main(String[] args) {  
  
    Mensch ersterMensch = new Mensch("Klaus");  
    Mensch zweiterMensch = new Mensch();  
    Mensch dritterMensch = new Mensch();  
    Mensch vierterMensch = new Mensch();  
  
    zweiterMensch.setName("Harald");  
    dritterMensch.setName("Susi");  
  
    Fahrrad erstesFahrrad = new Fahrrad("rot");  
    Fahrrad zweitesFahrrad = new Fahrrad();  
    Fahrrad drittesFahrrad = new Fahrrad("blau");  
  
    ersterMensch.kaufe(zweitesFahrrad);  
    dritterMensch.kaufe(erstesFahrrad);  
    dritterMensch.kaufe(drittesFahrrad);  
  
    zweiterMensch.heirate(dritterMensch);  
    dritterMensch.heirate(zweiterMensch);  
  
    zweitesFahrrad.setFarbe("gruen");  
    drittesFahrrad.setFarbe("gelb");  
  
    System.out.println(ersterMensch.getName());  
    System.out.println(zweiterMensch.getEhepartner().getName());  
    int anzahl = dritterMensch.getAnzahlFahrraeder();  
    for (int i = 0; i < anzahl; i++)  
        System.out.println(i+1 + ". Fahrrad: " + dritterMensch.getFahrrad(i).getFarbe());  
    System.out.println(vierterMensch.getName());  
  
} // main  
  
} // Klasse DynamischeOOP
```

1. **Wie viele Objekte** werden zur Laufzeit durch die main-Methode erzeugt? **2P**

7

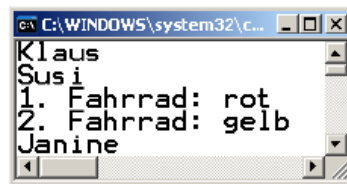
2. **Wie viele Methodenaufrufe auf Objekten** erfolgen zur Laufzeit in der main-Methode? **3P**

**Hinweis:**

- Die Häufigkeit des Durchlaufes des Schleifenkörpers der Zählschleife ist vom Wert der Variablen `anzahl1` abhängig (0..100) und somit auch die Anzahl der Methodenaufrufe durch die Zählschleife!
- Bitte nicht die Konstruktoraufrufe mitzählen!

18 (3P)    16,17,19,20 (2P)    14,15,21,22 (1P)

3. Wie lautet genau die **BildschirmAusgabe**? **5P**



```
C:\WINDOWS\system32\c...
Klaus
Susi
1. Fahrrad: rot
2. Fahrrad: gelb
Janine
```

5 \* 1P

**11. Objektorientierte Modellierung (OOM):**

Gegeben ist das folgende Szenario:

**10P**

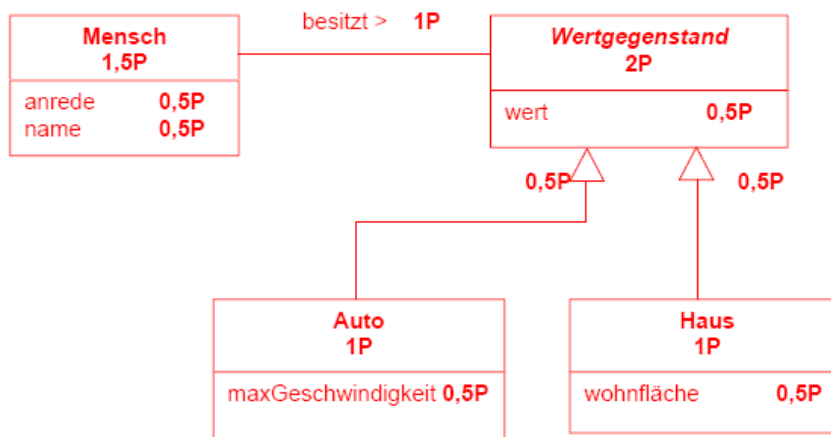
Menschen können Wertgegenstände besitzen, z.B.:

Herr Müller besitzt ein konkretes Auto mit einer maximalen Geschwindigkeit und einem gewissen Wert.

Frau Meyer besitzt ein konkretes Haus mit einer konkreten Quadratmeterzahl an Wohnfläche und einem gewissen Wert.

**Modellieren** Sie mit Hilfe eines **UML-Klassendiagramms** dieses Szenario!

**Hinweis:** Es genügt die **OOA-Ansicht**, also ohne setter, getter, Konstruktoren und Datentypen. Es müssen also nur Klassennamen und Attributnamen angeführt werden, aber keinerlei Methoden!



## N) Ergebnisse des Vergleichstests (Hauptstudie 2007/2008)

OG71 (OOP-First): Gesamtergebnis

| OG71 | CODE                | 1          | 2          | 3          | 4          | 5          | 67          | 8          | 9          | 10         | 11         | Summe       | S: 1-8      |
|------|---------------------|------------|------------|------------|------------|------------|-------------|------------|------------|------------|------------|-------------|-------------|
| 1    | ANDA                |            |            |            |            |            |             |            |            |            |            |             |             |
| 2    | ANFR                | 6,0        | 7,5        | 5,5        | 4,5        | 5,5        | 10,5        | 8,5        | 8,0        | 7,0        | 8,5        | 71,5        | 48,0        |
| 3    | ANPE                | 9,0        | 7,0        | 8,5        | 6,0        | 7,5        | 17,5        | 8,5        | 4,0        | 8,0        | 7,0        | 83,0        | 64,0        |
| 4    | ARST                | 7,0        | 5,0        | 3,5        | 4,5        | 6,5        | 16,0        | 10,0       | 9,0        | 8,5        | 3,0        | 73,0        | 52,5        |
| 5    | CLUL                | 6,5        | 6,0        | 6,0        | 6,0        | 6,0        | 11,5        | 2,5        | 7,5        | 6,5        | 5,0        | 63,5        | 44,5        |
| 6    | COVO                | 7,0        | 10,0       | 5,5        | 8,5        | 8,0        | 15,5        | 9,0        | 7,0        | 10,0       | 0,0        | 80,5        | 63,5        |
| 7    | DAHA                | 6,5        | 6,0        | 3,5        | 4,0        | 6,0        | 16,0        | 9,5        | 7,0        | 7,5        | 9,0        | 75,0        | 51,5        |
| 8    | EWAD                | 5,0        | 5,0        | 5,0        | 6,0        | 7,5        | 15,5        | 9,5        | 3,5        | 6,0        | 9,0        | 72,0        | 53,5        |
| 9    | JAKR                | 7,0        | 8,0        | 7,0        | 7,5        | 7,0        | 15,5        | 8,5        | 3,5        | 5,5        | 7,0        | 76,5        | 60,5        |
| 10   | JETH                | 8,5        | 5,5        | 5,0        | 7,0        | 6,5        | 17,0        | 7,0        | 9,5        | 9,0        | 9,0        | 84,0        | 56,5        |
| 11   | LIFR                | 4,5        | 3,5        | 6,5        | 7,0        | 5,5        | 12,5        | 8,0        | 7,0        | 8,0        | 8,0        | 70,5        | 47,5        |
| 12   | LINO                | 6,0        | 3,0        | 3,0        | 3,0        | 4,0        | 16,0        | 2,0        | 2,5        | 5,0        | 2,0        | 46,5        | 37,0        |
| 13   | MARE                | 7,5        | 4,5        | 4,0        | 2,0        | 3,0        | 13,0        | 9,0        | 3,5        | 6,0        | 9,5        | 62,0        | 43,0        |
| 14   | NAGO                |            |            |            |            |            |             |            |            |            |            |             |             |
| 15   | NUYU                | 9,0        | 7,5        | 7,5        | 8,5        | 10,0       | 18,0        | 3,0        | 8,5        | 9,0        | 6,5        | 87,5        | 63,5        |
| 16   | PERA                | 4,5        | 7,5        | 4,5        | 7,5        | 5,5        | 12,0        | 7,5        | 2,0        | 6,0        | 5,0        | 62,0        | 49,0        |
| 17   | PETH                |            |            |            |            |            |             |            |            |            |            |             |             |
| 18   | REDE                | 7,0        | 4,0        | 2,0        | 2,0        | 3,0        | 12,5        | 8,5        | 6,5        | 6,5        | 3,5        | 55,5        | 39,0        |
| 19   | SIJO                | 9,0        | 8,5        | 9,5        | 5,5        | 8,5        | 17,0        | 8,5        | 7,0        | 8,0        | 9,5        | 91,0        | 66,5        |
| 20   | SIPE                | 3,5        | 2,0        | 6,0        | 3,5        | 1,5        | 11,0        | 4,5        | 4,5        | 4,0        | 6,5        | 47,0        | 32,0        |
| 21   | SIVO                | 8,0        | 8,5        | 7,5        | 8,5        | 6,0        | 13,5        | 10,0       | 9,0        | 8,0        | 9,5        | 88,5        | 62,0        |
| 22   | SUNO                | 7,0        | 10,0       | 9,5        | 7,5        | 9,0        | 16,5        | 9,5        | 9,0        | 7,0        | 9,0        | 94,0        | 69,0        |
|      |                     |            |            |            |            |            |             |            |            |            |            |             |             |
|      | <b>Durchschnitt</b> | <b>6,8</b> | <b>6,3</b> | <b>5,8</b> | <b>5,7</b> | <b>6,1</b> | <b>14,6</b> | <b>7,6</b> | <b>6,2</b> | <b>7,1</b> | <b>6,7</b> | <b>72,8</b> | <b>52,8</b> |
|      |                     |            |            |            |            |            |             |            |            |            |            |             |             |
|      | <b>Prozentual</b>   | <b>68</b>  | <b>63</b>  | <b>58</b>  | <b>57</b>  | <b>61</b>  | <b>73</b>   | <b>76</b>  | <b>62</b>  | <b>71</b>  | <b>67</b>  | <b>66</b>   | <b>66</b>   |

Die Summe 72,8 und die Teilsumme (Aufgaben 1 bis 8) 52,8 wurde von Excel aus den einzelnen Teilergebnissen mit vielen Nachkommastellen berechnet. Summiert man die gerundeten Zahlen kommt man auf 72,9 bzw. 52,9.

OG71 (OOP-First): Teilergebnisse inkl. Itemschwierigkeiten

| OG71<br>max. Punkte | 1.1<br>1,0  | 1.2<br>1,0  | 1.3<br>1,0  | 1.4<br>1,0  | 1.5<br>3,0  | 1.6<br>3,0  | Summe<br>10,0 | 2.1<br>1    | 2.2<br>1    | 2.3a<br>4   | 2.3b<br>4   | Summe<br>10,0 |
|---------------------|-------------|-------------|-------------|-------------|-------------|-------------|---------------|-------------|-------------|-------------|-------------|---------------|
| ANFR                | 1           | 1           | 0           | 0,5         | 1,5         | 2           | 6,0           | 1,0         | 1,0         | 2,5         | 3,0         | 7,5           |
| ANPE                | 1           | 1           | 1           | 1           | 3           | 2           | 9,0           | 1,0         | 1,0         | 3,0         | 2,0         | 7,0           |
| ARST                | 1           | 1           | 0           | 0           | 2           | 3           | 7,0           | 1,0         | 0,0         | 2,5         | 1,5         | 5,0           |
| CLUL                | 1           | 1           | 1           | 0           | 1,5         | 2           | 6,5           | 0,0         | 0,0         | 2,5         | 3,5         | 6,0           |
| COVO                | 1           | 1           | 0           | 0           | 3           | 2           | 7,0           | 1,0         | 1,0         | 4,0         | 4,0         | 10,0          |
| DAHA                | 0,5         | 0           | 0           | 1           | 2           | 3           | 6,5           | 0,0         | 1,0         | 1,0         | 4,0         | 6,0           |
| EWAD                | 0,5         | 1           | 0           | 1           | 0,5         | 2           | 5,0           | 1,0         | 0,0         | 1,5         | 2,5         | 5,0           |
| JAKR                | 1           | 1           | 1           | 0           | 2           | 2           | 7,0           | 1,0         | 1,0         | 2,5         | 3,5         | 8,0           |
| JETH                | 1           | 1           | 1           | 0           | 2,5         | 3           | 8,5           | 1,0         | 0,0         | 1,5         | 3,0         | 5,5           |
| LIFR                | 0           | 0,5         | 0           | 0           | 2           | 2           | 4,5           | 1,0         | 0,0         | 1,5         | 1,0         | 3,5           |
| LINO                | 0,5         | 1           | 0           | 1           | 1           | 2,5         | 6,0           | 0,5         | 0,0         | 1,5         | 1,0         | 3,0           |
| MARE                | 0,5         | 1           | 1           | 1           | 1           | 3           | 7,5           | 1,0         | 0,0         | 1,5         | 2,0         | 4,5           |
| NUYU                | 1           | 1           | 1           | 1           | 3           | 2           | 9,0           | 1,0         | 1,0         | 1,5         | 4,0         | 7,5           |
| PERA                | 0           | 1           | 0           | 0           | 2,5         | 1           | 4,5           | 1,0         | 1,0         | 1,5         | 4,0         | 7,5           |
| REDE                | 1           | 1           | 0           | 1           | 2           | 2           | 7,0           | 1,0         | 1,0         | 0,0         | 2,0         | 4,0           |
| SIJO                | 1           | 1           | 1           | 1           | 3           | 2           | 9,0           | 1,0         | 1,0         | 2,5         | 4,0         | 8,5           |
| SIPE                | 0,5         | 1           | 0           | 1           | 1           | 0           | 3,5           | 1,0         | 1,0         | 0,0         | 0,0         | 2,0           |
| SIVO                | 1           | 1           | 0           | 1           | 3           | 2           | 8,0           | 1,0         | 1,0         | 2,5         | 4,0         | 8,5           |
| SUNO                | 1           | 1           | 0           | 0           | 3           | 2           | 7,0           | 1,0         | 1,0         | 4,0         | 4,0         | 10,0          |
| <b>Durchschnitt</b> | <b>0,8</b>  | <b>0,9</b>  | <b>0,4</b>  | <b>0,6</b>  | <b>2,1</b>  | <b>2,1</b>  | <b>6,8</b>    | <b>0,9</b>  | <b>0,6</b>  | <b>2,0</b>  | <b>2,8</b>  | <b>6,3</b>    |
| <b>Prozentual</b>   | <b>76</b>   | <b>92</b>   | <b>37</b>   | <b>55</b>   | <b>69</b>   | <b>69</b>   | <b>68</b>     | <b>87</b>   | <b>63</b>   | <b>49</b>   | <b>70</b>   | <b>63</b>     |
| <b>PI</b>           | <b>0,76</b> | <b>0,92</b> | <b>0,37</b> | <b>0,55</b> | <b>0,69</b> | <b>0,69</b> |               | <b>0,87</b> | <b>0,63</b> | <b>0,49</b> | <b>0,70</b> |               |

| 3.1<br>1    | 3.2<br>1    | 3.3<br>2    | 3.4a<br>2   | 3.4b<br>2   | 3.5<br>2    | Summe<br>10,0 | 4.1<br>1    | 4.2<br>1    | 4.3<br>2    | 4.4<br>3    | 4.5<br>3    | Summe<br>10,0 | 5.1<br>1    | 5.2<br>1    | 5.3<br>1    | 5.4<br>3    | 5.5<br>4    | Summe<br>10,0 |
|-------------|-------------|-------------|-------------|-------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|---------------|
| 0,0         | 0,0         | 0,0         | 2,0         | 2,0         | 1,5         | 5,5           | 1,0         | 0,0         | 0,0         | 1,0         | 2,5         | 4,5           | 0,0         | 0,0         | 0,0         | 2,0         | 3,5         | 5,5           |
| 1,0         | 0,0         | 2,0         | 2,0         | 2,0         | 1,5         | 8,5           | 0,5         | 1,0         | 0,0         | 2,0         | 2,5         | 6,0           | 0,0         | 0,0         | 0,5         | 3,0         | 4,0         | 7,5           |
| 0,0         | 0,0         | 0,0         | 0,0         | 2,0         | 1,5         | 3,5           | 1,0         | 0,0         | 0,0         | 1,0         | 2,5         | 4,5           | 0,0         | 1,0         | 0,0         | 2,0         | 3,5         | 6,5           |
| 1,0         | 0,0         | 0,0         | 2,0         | 2,0         | 1,0         | 6,0           | 1,0         | 0,0         | 2,0         | 1,0         | 2,0         | 6,0           | 1,0         | 0,0         | 1,0         | 1,5         | 2,5         | 6,0           |
| 1,0         | 0,0         | 0,0         | 2,0         | 2,0         | 0,5         | 5,5           | 1,0         | 1,0         | 2,0         | 2,5         | 2,0         | 8,5           | 0,0         | 1,0         | 1,0         | 3,0         | 3,0         | 8,0           |
| 0,0         | 0,5         | 0,0         | 0,0         | 2,0         | 1,0         | 3,5           | 0,5         | 0,0         | 0,0         | 1,5         | 2,0         | 4,0           | 0,0         | 0,0         | 1,0         | 2,5         | 2,5         | 6,0           |
| 1,0         | 0,5         | 0,0         | 2,0         | 1,0         | 0,5         | 5,0           | 1,0         | 1,0         | 2,0         | 0,5         | 1,5         | 6,0           | 1,0         | 1,0         | 0,0         | 2,0         | 3,5         | 7,5           |
| 0,0         | 0,0         | 2,0         | 2,0         | 2,0         | 1,0         | 7,0           | 1,0         | 1,0         | 2,0         | 1,0         | 2,5         | 7,5           | 1,0         | 0,0         | 0,0         | 2,5         | 3,5         | 7,0           |
| 1,0         | 0,5         | 0,0         | 0,0         | 2,0         | 1,5         | 5,0           | 0,5         | 1,0         | 2,0         | 1,0         | 2,5         | 7,0           | 0,0         | 0,0         | 0,5         | 2,5         | 3,5         | 6,5           |
| 0,0         | 1,0         | 2,0         | 0,0         | 2,0         | 1,5         | 6,5           | 1,0         | 1,0         | 2,0         | 1,0         | 2,0         | 7,0           | 0,0         | 0,0         | 0,0         | 2,5         | 3,0         | 5,5           |
| 0,0         | 0,0         | 0,0         | 2,0         | 1,0         | 0,0         | 3,0           | 0,5         | 0,0         | 0,0         | 0,0         | 2,5         | 3,0           | 0,0         | 1,0         | 0,0         | 1,0         | 2,0         | 4,0           |
| 1,0         | 0,0         | 0,0         | 0,0         | 2,0         | 1,0         | 4,0           | 0,5         | 0,0         | 0,0         | 0,0         | 1,5         | 2,0           | 0,0         | 0,0         | 0,0         | 0,0         | 3,0         | 3,0           |
| 1,0         | 0,5         | 2,0         | 0,0         | 2,0         | 2,0         | 7,5           | 1,0         | 1,0         | 1,0         | 3,0         | 2,5         | 8,5           | 1,0         | 1,0         | 1,0         | 3,0         | 4,0         | 10,0          |
| 0,0         | 0,0         | 0,0         | 2,0         | 2,0         | 0,5         | 4,5           | 1,0         | 0,0         | 2,0         | 2,5         | 2,0         | 7,5           | 0,0         | 1,0         | 0,5         | 2,0         | 2,0         | 5,5           |
| 1,0         | 0,0         | 0,0         | 0,0         | 0,5         | 0,5         | 2,0           | 0,0         | 0,0         | 0,0         | 0,0         | 2,0         | 2,0           | 1,0         | 1,0         | 0,0         | 0,0         | 1,0         | 3,0           |
| 1,0         | 0,5         | 2,0         | 2,0         | 2,0         | 2,0         | 9,5           | 0,5         | 0,0         | 0,0         | 2,5         | 2,5         | 5,5           | 0,0         | 1,0         | 1,0         | 2,5         | 4,0         | 8,5           |
| 1,0         | 1,0         | 2,0         | 2,0         | 0,0         | 0,0         | 6,0           | 1,0         | 1,0         | 0,0         | 0,0         | 1,5         | 3,5           | 0,0         | 0,0         | 0,5         | 0,0         | 1,0         | 1,5           |
| 1,0         | 1,0         | 2,0         | 2,0         | 0,0         | 1,5         | 7,5           | 1,0         | 1,0         | 2,0         | 2,0         | 2,5         | 8,5           | 0,0         | 1,0         | 0,5         | 1,5         | 3,0         | 6,0           |
| 1,0         | 0,5         | 2,0         | 2,0         | 2,0         | 2,0         | 9,5           | 0,5         | 0,0         | 2,0         | 3,0         | 2,0         | 7,5           | 0,0         | 1,0         | 1,0         | 3,0         | 4,0         | 9,0           |
| <b>0,6</b>  | <b>0,3</b>  | <b>0,8</b>  | <b>1,3</b>  | <b>1,6</b>  | <b>1,1</b>  | <b>5,8</b>    | <b>0,8</b>  | <b>0,5</b>  | <b>1,0</b>  | <b>1,3</b>  | <b>2,2</b>  | <b>5,7</b>    | <b>0,3</b>  | <b>0,5</b>  | <b>0,4</b>  | <b>1,9</b>  | <b>3,0</b>  | <b>6,1</b>    |
| <b>63</b>   | <b>32</b>   | <b>42</b>   | <b>63</b>   | <b>80</b>   | <b>55</b>   | <b>58</b>     | <b>76</b>   | <b>47</b>   | <b>50</b>   | <b>45</b>   | <b>72</b>   | <b>57</b>     | <b>26</b>   | <b>53</b>   | <b>45</b>   | <b>64</b>   | <b>74</b>   | <b>61</b>     |
| <b>0,63</b> | <b>0,32</b> | <b>0,42</b> | <b>0,63</b> | <b>0,80</b> | <b>0,55</b> |               | <b>0,76</b> | <b>0,47</b> | <b>0,50</b> | <b>0,45</b> | <b>0,72</b> |               | <b>0,26</b> | <b>0,53</b> | <b>0,45</b> | <b>0,64</b> | <b>0,74</b> |               |

| 67.1       | 67.2       | 67.3       | 67.4       | 67.5       | 67.6       | 67.7       | 67.8       | 67.9       | 67.10      | 67.11      | Summe       |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-------------|
| 1          | 1          | 1          | 1          | 1          | 1          | 2          | 3          | 2          | 4,5        | 2,5        | 20,0        |
| 0,0        | 0,0        | 0,0        | 1,0        | 1,0        | 0,0        | 2,0        | 1,5        | 1,0        | 3,5        | 0,5        | 10,5        |
| 0,5        | 1,0        | 0,0        | 1,0        | 1,0        | 1,0        | 2,0        | 3,0        | 2,0        | 4,5        | 1,5        | 17,5        |
| 0,5        | 1,0        | 0,0        | 1,0        | 1,0        | 1,0        | 2,0        | 2,0        | 2,0        | 4,0        | 1,5        | 16,0        |
| 0,5        | 1,0        | 0,0        | 1,0        | 0,0        | 0,0        | 1,5        | 1,5        | 2,0        | 2,5        | 1,5        | 11,5        |
| 1,0        | 1,0        | 1,0        | 0,0        | 1,0        | 1,0        | 2,0        | 3,0        | 2,0        | 3,0        | 0,5        | 15,5        |
| 0,5        | 0,5        | 0,5        | 1,0        | 1,0        | 0,5        | 2,0        | 3,0        | 2,0        | 4,5        | 0,5        | 16,0        |
| 0,5        | 0,5        | 1,0        | 1,0        | 1,0        | 0,0        | 2,0        | 2,0        | 2,0        | 4,5        | 1,0        | 15,5        |
| 1,0        | 0,5        | 0,0        | 1,0        | 0,0        | 1,0        | 2,0        | 2,5        | 2,0        | 4,0        | 1,5        | 15,5        |
| 1,0        | 1,0        | 0,0        | 1,0        | 1,0        | 0,5        | 2,0        | 3,0        | 2,0        | 4,0        | 1,5        | 17,0        |
| 0,5        | 0,5        | 0,0        | 0,0        | 1,0        | 0,0        | 2,0        | 3,0        | 2,0        | 3,5        | 0,0        | 12,5        |
| 1,0        | 0,5        | 0,0        | 1,0        | 1,0        | 0,5        | 2,0        | 2,0        | 2,0        | 4,0        | 2,0        | 16,0        |
| 0,0        | 0,5        | 0,0        | 0,0        | 1,0        | 0,0        | 2,0        | 3,0        | 2,0        | 3,5        | 1,0        | 13,0        |
| 1,0        | 0,5        | 0,5        | 0,0        | 1,0        | 1,0        | 2,0        | 3,0        | 2,0        | 4,5        | 2,5        | 18,0        |
| 0,0        | 0,5        | 0,0        | 1,0        | 1,0        | 0,0        | 1,5        | 2,5        | 1,0        | 4,5        | 0,0        | 12,0        |
| 0,0        | 0,0        | 0,5        | 0,0        | 1,0        | 1,0        | 2,0        | 2,5        | 2,0        | 3,0        | 0,5        | 12,5        |
| 0,5        | 0,5        | 1,0        | 1,0        | 1,0        | 1,0        | 1,5        | 3,0        | 2,0        | 4,0        | 1,5        | 17,0        |
| 1,0        | 0,0        | 0,0        | 0,0        | 1,0        | 1,0        | 1,0        | 1,5        | 1,5        | 2,0        | 2,0        | 11,0        |
| 0,5        | 0,0        | 0,0        | 0,0        | 1,0        | 0,5        | 2,0        | 2,5        | 2,0        | 4,0        | 1,0        | 13,5        |
| 1,0        | 0,5        | 1,0        | 1,0        | 1,0        | 1,0        | 1,5        | 3,0        | 2,0        | 4,0        | 0,5        | 16,5        |
| <b>0,6</b> | <b>0,5</b> | <b>0,3</b> | <b>0,6</b> | <b>0,9</b> | <b>0,6</b> | <b>1,8</b> | <b>2,5</b> | <b>1,9</b> | <b>3,8</b> | <b>1,1</b> | <b>14,6</b> |
| 58         | 53         | 29         | 63         | 89         | 58         | 92         | 83         | 93         | 84         | 44         | 73          |

| 8.1        | 8.2        | 8.3        | 8.4        | Summe      |
|------------|------------|------------|------------|------------|
| 1          | 1          | 1          | 7          | 10,0       |
| 1,0        | 1,0        | 1,0        | 5,5        | 8,5        |
| 1,0        | 1,0        | 1,0        | 5,5        | 8,5        |
| 1,0        | 1,0        | 1,0        | 7,0        | 10,0       |
| 1,0        | 0,0        | 1,0        | 0,5        | 2,5        |
| 1,0        | 1,0        | 1,0        | 6,0        | 9,0        |
| 1,0        | 1,0        | 1,0        | 6,5        | 9,5        |
| 1,0        | 1,0        | 1,0        | 6,5        | 9,5        |
| 1,0        | 0,5        | 1,0        | 6,0        | 8,5        |
| 0,0        | 0,5        | 1,0        | 5,5        | 7,0        |
| 1,0        | 0,0        | 1,0        | 6,0        | 8,0        |
| 1,0        | 0,0        | 1,0        | 0,0        | 2,0        |
| 1,0        | 1,0        | 1,0        | 6,0        | 9,0        |
| 1,0        | 1,0        | 1,0        | 0,0        | 3,0        |
| 1,0        | 0,5        | 0,0        | 6,0        | 7,5        |
| 1,0        | 1,0        | 1,0        | 5,5        | 8,5        |
| 1,0        | 1,0        | 1,0        | 5,5        | 8,5        |
| 1,0        | 1,0        | 1,0        | 1,5        | 4,5        |
| 1,0        | 1,0        | 1,0        | 7,0        | 10,0       |
| 1,0        | 1,0        | 1,0        | 6,5        | 9,5        |
| <b>0,9</b> | <b>0,8</b> | <b>0,9</b> | <b>4,9</b> | <b>7,6</b> |
| 95         | 76         | 95         | 70         | 76         |

| 9.1        | 9.2        | 9.3a       | 9.3b       | Summe      |
|------------|------------|------------|------------|------------|
| 1          | 1          | 4          | 4          | 10,0       |
| 1,0        | 1,0        | 3,0        | 3,0        | 8,0        |
| 0,5        | 1,0        | 1,5        | 1,0        | 4,0        |
| 1,0        | 1,0        | 3,0        | 4,0        | 9,0        |
| 0,5        | 1,0        | 3,0        | 3,0        | 7,5        |
| 0,5        | 1,0        | 4,0        | 1,5        | 7,0        |
| 0,5        | 1,0        | 3,0        | 2,5        | 7,0        |
| 0,5        | 0,0        | 3,0        | 0,0        | 3,5        |
| 0,5        | 1,0        | 2,0        | 0,0        | 3,5        |
| 0,5        | 1,0        | 4,0        | 4,0        | 9,5        |
| 0,5        | 1,0        | 2,5        | 3,0        | 7,0        |
| 0,0        | 1,0        | 1,5        | 0,0        | 2,5        |
| 0,5        | 0,5        | 1,0        | 1,5        | 3,5        |
| 0,5        | 1,0        | 4,0        | 3,0        | 8,5        |
| 0,5        | 0,0        | 1,5        | 0,0        | 2,0        |
| 0,5        | 0,5        | 2,5        | 3,0        | 6,5        |
| 0,5        | 1,0        | 3,5        | 2,0        | 7,0        |
| 1,0        | 1,0        | 2,5        | 0,0        | 4,5        |
| 1,0        | 1,0        | 4,0        | 3,0        | 9,0        |
| 0,5        | 1,0        | 3,5        | 4,0        | 9,0        |
| <b>0,6</b> | <b>0,8</b> | <b>2,8</b> | <b>2,0</b> | <b>6,2</b> |
| 58         | 84         | 70         | 51         | 62         |

0,58 0,53 0,29 0,63 0,89 0,58 0,92 0,83 0,93 0,84 0,44

0,95 0,76 0,95 0,70

0,58 0,84 0,70 0,51

| 10.1       | 10.2       | 10.3       | Summe      |
|------------|------------|------------|------------|
| 2          | 3          | 5          | 10,0       |
| 2,0        | 2,0        | 3,0        | 7,0        |
| 2,0        | 2,0        | 4,0        | 8,0        |
| 2,0        | 3,0        | 3,5        | 8,5        |
| 2,0        | 2,0        | 2,5        | 6,5        |
| 2,0        | 3,0        | 5,0        | 10,0       |
| 2,0        | 3,0        | 2,5        | 7,5        |
| 2,0        | 2,0        | 2,0        | 6,0        |
| 2,0        | 0,0        | 3,5        | 5,5        |
| 2,0        | 3,0        | 4,0        | 9,0        |
| 2,0        | 2,0        | 4,0        | 8,0        |
| 2,0        | 3,0        | 0,0        | 5,0        |
| 2,0        | 2,0        | 2,0        | 6,0        |
| 2,0        | 2,0        | 5,0        | 9,0        |
| 2,0        | 1,0        | 3,0        | 6,0        |
| 2,0        | 3,0        | 1,5        | 6,5        |
| 2,0        | 2,0        | 4,0        | 8,0        |
| 2,0        | 1,0        | 1,0        | 4,0        |
| 2,0        | 2,0        | 4,0        | 8,0        |
| 2,0        | 0,0        | 5,0        | 7,0        |
| <b>2,0</b> | <b>2,0</b> | <b>3,1</b> | <b>7,1</b> |
| 100        | 67         | 63         | 71         |

| 11         | Summe      |
|------------|------------|
| 10         | 10,0       |
| 8,5        | 8,5        |
| 7,0        | 7,0        |
| 3,0        | 3,0        |
| 5,0        | 5,0        |
| 0,0        | 0,0        |
| 9,0        | 9,0        |
| 9,0        | 9,0        |
| 7,0        | 7,0        |
| 9,0        | 9,0        |
| 8,0        | 8,0        |
| 2,0        | 2,0        |
| 9,5        | 9,5        |
| 6,5        | 6,5        |
| 5,0        | 5,0        |
| 3,5        | 3,5        |
| 9,5        | 9,5        |
| 6,5        | 6,5        |
| 9,5        | 9,5        |
| 9,0        | 9,0        |
| <b>6,7</b> | <b>6,7</b> |
| 67         | 67         |

| Summe       | Prozentual  |
|-------------|-------------|
| 110         | 100         |
| 71,5        | 65,0        |
| 83,0        | 75,5        |
| 73,0        | 66,4        |
| 63,5        | 57,7        |
| 80,5        | 73,2        |
| 75,0        | 68,2        |
| 72,0        | 65,5        |
| 76,5        | 69,5        |
| 84,0        | 76,4        |
| 70,5        | 64,1        |
| 46,5        | 42,3        |
| 62,0        | 56,4        |
| 87,5        | 79,5        |
| 62,0        | 56,4        |
| 55,5        | 50,5        |
| 91,0        | 82,7        |
| 47,0        | 42,7        |
| 88,5        | 80,5        |
| 94,0        | 85,5        |
| <b>72,8</b> | <b>66,2</b> |
| 66          | 66          |

| OG71         |
|--------------|
| max. Punkte  |
| ANFR         |
| ANPE         |
| ARST         |
| CLUL         |
| COVO         |
| DAHA         |
| EWAD         |
| JAKR         |
| JETH         |
| LIFR         |
| LINO         |
| MARE         |
| NUYU         |
| PERA         |
| REDE         |
| SIJO         |
| SIPE         |
| SIVO         |
| SUNO         |
| Durchschnitt |
| Prozentual   |

1,00 0,67 0,63

0,67

Durchschnitt

0,65

PI

OG72 (OOP-Later): Gesamtergebnis

| OG72 | Code                | 1          | 2          | 3          | 4          | 5          | 67          | 8          | 9          | 10         | 11         | Summe       | S: 1-8      |
|------|---------------------|------------|------------|------------|------------|------------|-------------|------------|------------|------------|------------|-------------|-------------|
| 1    | ANHA                | 5,0        | 5,5        | 4,0        | 2,0        | 3,0        | 8,0         | 5,0        | 3,5        | 7,0        | 7,5        | 50,5        | 32,5        |
| 2    | ANST                | 7,0        | 8,5        | 9,0        | 4,0        | 1,5        | 11,0        | 6,5        | 4,5        | 9,0        | 6,5        | 67,5        | 47,5        |
| 3    | BAJÖ                | 7,0        | 7,0        | 4,5        | 1,5        | 2,5        | 16,5        | 7,5        | 3,5        | 8,0        | 2,0        | 60,0        | 46,5        |
| 4    | BERE                | 8,0        | 9,0        | 7,5        | 8,0        | 5,0        | 13,5        | 6,5        | 1,0        | 7,0        | 4,0        | 69,5        | 57,5        |
| 5    | BETH                | 3,0        | 7,0        | 4,5        | 1,5        | 3,0        | 18,5        | 6,0        | 4,5        | 4,5        | 5,5        | 58,0        | 43,5        |
| 6    | BIFR                | 8,5        | 8,5        | 8,0        | 7,5        | 6,0        | 14,5        | 6,5        | 3,5        | 10,0       | 9,5        | 82,5        | 59,5        |
| 7    | BIHO                | 9,0        | 8,5        | 9,5        | 5,0        | 7,0        | 18,5        | 8,5        | 5,0        | 7,5        | 3,0        | 81,5        | 66,0        |
| 8    | BITH                | 3,0        | 6,0        | 2,5        | 2,0        | 1,0        | 10,0        | 4,5        | 4,0        | 4,0        | 3,5        | 40,5        | 29,0        |
| 9    | BIUW                | 7,5        | 6,5        | 2,5        | 4,5        | 5,5        | 16,0        | 7,5        | 7,0        | 9,0        | 8,0        | 74,0        | 50,0        |
| 10   | CARU                | 9,5        | 8,0        | 6,5        | 4,5        | 8,0        | 15,5        | 8,0        | 5,5        | 7,0        | 4,0        | 76,5        | 60,0        |
| 11   | CHGE                |            |            |            |            |            |             |            |            |            |            |             | 0,0         |
| 12   | CHMI                | 4,0        | 2,0        | 4,0        | 0,5        | 1,0        | 10,5        | 5,5        | 5,0        | 4,5        | 5,5        | 42,5        | 27,5        |
| 13   | DAUW                | 8,0        | 6,5        | 3,5        | 1,5        | 7,0        | 17,5        | 8,5        | 5,0        | 7,5        | 8,0        | 73,0        | 52,5        |
| 14   | INWO                | 10,0       | 8,5        | 7,5        | 8,0        | 8,5        | 17,5        | 9,5        | 8,0        | 9,0        | 6,5        | 93,0        | 69,5        |
| 15   | JUKL                | 5,0        | 5,0        | 2,5        | 0,0        | 3,5        | 8,0         | 1,5        | 4,0        | 4,0        | 3,5        | 37,0        | 25,5        |
| 16   | LASA                | 10,0       | 7,5        | 9,5        | 7,5        | 10,0       | 18,5        | 8,5        | 7,0        | 8,5        | 9,0        | 96,0        | 71,5        |
| 17   | MAAC                | 6,0        | 6,0        | 6,5        | 0,0        | 3,0        | 16,0        | 8,0        | 4,5        | 8,0        | 7,0        | 65,0        | 45,5        |
| 18   | REYV                |            |            |            |            |            |             |            |            |            |            |             |             |
| 19   | SADI                | 8,0        | 8,5        | 9,0        | 8,5        | 9,0        | 19,5        | 8,5        | 9,5        | 10,0       | 9,5        | 100,0       | 71,0        |
| 20   | SILO                | 9,0        | 9,0        | 5,0        | 2,5        | 6,0        | 15,5        | 6,5        | 4,5        | 3,5        | 4,0        | 65,5        | 53,5        |
| 21   | TAWA                | 5,0        | 7,5        | 7,5        | 2,0        | 0,5        | 18,0        | 7,5        | 3,5        | 5,5        | 5,5        | 62,5        | 48,0        |
| 22   | VEVA                |            |            |            |            |            |             |            |            |            |            |             |             |
|      | <b>Durchschnitt</b> | <b>7,0</b> | <b>7,1</b> | <b>6,0</b> | <b>3,7</b> | <b>4,8</b> | <b>14,9</b> | <b>6,9</b> | <b>4,9</b> | <b>7,0</b> | <b>5,9</b> | <b>68,2</b> | <b>50,3</b> |
|      | <b>Prozentual</b>   | <b>70</b>  | <b>71</b>  | <b>60</b>  | <b>37</b>  | <b>48</b>  | <b>74</b>   | <b>69</b>  | <b>49</b>  | <b>70</b>  | <b>59</b>  | <b>62</b>   | <b>63</b>   |

Die Teilsumme (Aufgaben 1 bis 8) 50,3 wurde von Excel aus den einzelnen Teilergebnissen mit vielen Nachkommastellen berechnet. Summiert man die gerundeten Zahlen kommt man auf 50,4.



OG72 (OOP-Later): Teilergebnisse inkl. Itemschwierigkeiten

| OG72         | 1.1                           | 1.2        | 1.3        | 1.4        | 1.5        | 1.6        | Summe               | 2.1        | 2.2        | 2.3a       | 2.3b       | Summe      |
|--------------|-------------------------------|------------|------------|------------|------------|------------|---------------------|------------|------------|------------|------------|------------|
| max. Punkte  | 1,0                           | 1,0        | 1,0        | 1,0        | 3,0        | 3,0        | 10,0                | 1          | 1          | 4          | 4          | 10,0       |
| ANHA         | 1                             | 0          | 0          | 1          | 0          | 3          | 5,0                 | 0,0        | 1,0        | 1,0        | 3,5        | 5,5        |
| ANST         | 1                             | 0,5        | 0          | 1          | 1,5        | 3          | 7,0                 | 1,0        | 1,0        | 2,5        | 4,0        | 8,5        |
| BAJÖ         | 1                             | 0          | 0          | 1          | 2          | 3          | 7,0                 | 0,0        | 0,0        | 3,0        | 4,0        | 7,0        |
| BERE         | 1                             | 1          | 1          | 0          | 3          | 2          | 8,0                 | 1,0        | 1,0        | 3,0        | 4,0        | 9,0        |
| BETH         | 1                             | 0          | 0          | 0,5        | 1,5        | 0,0        | 3,0                 | 1,0        | 0,0        | 3,0        | 3,0        | 7,0        |
| BIFR         | 1                             | 0,5        | 1          | 1          | 2          | 3          | 8,5                 | 1,0        | 1,0        | 2,5        | 4,0        | 8,5        |
| BIHO         | 1                             | 1          | 1          | 1          | 2          | 3          | 9,0                 | 0,5        | 1,0        | 3,0        | 4,0        | 8,5        |
| BITH         | 1                             | 0,5        | 0          | 1,0        | 0          | 0,5        | 3,0                 | 1,0        | 1,0        | 1,5        | 2,5        | 6,0        |
| BIUW         | 0,5                           | 0,5        | 1          | 1          | 1,5        | 3          | 7,5                 | 1,0        | 0,0        | 3,0        | 2,5        | 6,5        |
| CARU         | 0,5                           | 1          | 1          | 1          | 3          | 3          | 9,5                 | 1,0        | 0,0        | 3,0        | 4,0        | 8,0        |
| CHMI         | 1                             | 0,5        | 0          | 1          | 1,5        | 0          | 4,0                 | 0,0        | 0,0        | 0,0        | 2,0        | 2,0        |
| DAUW         | 1                             | 0          | 1          | 1          | 2          | 3          | 8,0                 | 0,0        | 0,0        | 3,0        | 3,5        | 6,5        |
| INWO         | 1                             | 1          | 1          | 1          | 3          | 3          | 10,0                | 1,0        | 1,0        | 2,5        | 4,0        | 8,5        |
| JUKL         | 0,5                           | 0          | 0          | 0          | 1,5        | 3          | 5,0                 | 1,0        | 1,0        | 0,0        | 3,0        | 5,0        |
| LASA         | 1                             | 1          | 1          | 1          | 3          | 3          | 10,0                | 1,0        | 1,0        | 1,5        | 4,0        | 7,5        |
| MAAC         | 0,5                           | 0,5        | 0          | 0          | 2          | 3          | 6,0                 | 1,0        | 1,0        | 0,0        | 4,0        | 6,0        |
| SADI         | 1                             | 0          | 1          | 1          | 2          | 3          | 8,0                 | 1,0        | 1,0        | 2,5        | 4,0        | 8,5        |
| SILO         | 1                             | 1          | 1          | 1          | 2          | 3          | 9,0                 | 0,0        | 1,0        | 4,0        | 4,0        | 9,0        |
| TAWA         | 1                             | 1          | 0          | 0,5        | 0,5        | 2          | 5,0                 | 0,0        | 1,0        | 3,0        | 3,5        | 7,5        |
| Durchschnitt | <b>0,9</b>                    | <b>0,5</b> | <b>0,5</b> | <b>0,8</b> | <b>1,8</b> | <b>2,4</b> | <b>7,0</b>          | <b>0,7</b> | <b>0,7</b> | <b>2,2</b> | <b>3,6</b> | <b>7,1</b> |
| Prozentual   | 89                            | 53         | 53         | 79         | 60         | 82         | 70                  | 66         | 68         | 55         | 89         | 71         |
| PI           | 0,89 0,53 0,53 0,79 0,60 0,82 |            |            |            |            |            | 0,66 0,68 0,55 0,89 |            |            |            |            |            |

| 3.1                           | 3.2        | 3.3        | 3.4a       | 3.4b       | 3.5        | Summe                    | 4.1        | 4.2        | 4.3        | 4.4        | 4.5        | Summe                    | 5.1        | 5.2        | 5.3        | 5.4        | 5.5        | Summe      |
|-------------------------------|------------|------------|------------|------------|------------|--------------------------|------------|------------|------------|------------|------------|--------------------------|------------|------------|------------|------------|------------|------------|
| 1                             | 1          | 2          | 2          | 2          | 2          | 10,0                     | 1          | 1          | 2          | 3          | 3          | 10,0                     | 1          | 1          | 1          | 3          | 4          | 10,0       |
| 0,0                           | 0,0        | 0,0        | 0,0        | 2,0        | 2,0        | 4,0                      | 0,0        | 0,0        | 0,0        | 0,0        | 2,0        | 2,0                      | 1,0        | 1,0        | 0,5        | 0,0        | 0,5        | 3,0        |
| 1,0                           | 0,0        | 2,0        | 2,0        | 2,0        | 2,0        | 9,0                      | 0,0        | 0,0        | 0,0        | 1,5        | 2,5        | 4,0                      | 0,0        | 0,0        | 0,0        | 1,5        | 0,0        | 1,5        |
| 1,0                           | 0,0        | 0,0        | 0,0        | 2,0        | 1,5        | 4,5                      | 0,0        | 0,0        | 0,0        | 0,0        | 1,5        | 1,5                      | 0,0        | 0,0        | 0,5        | 1,5        | 0,5        | 2,5        |
| 1,0                           | 0,0        | 2,0        | 2,0        | 2,0        | 0,5        | 7,5                      | 0,0        | 1,0        | 1,5        | 3,0        | 2,5        | 8,0                      | 1,0        | 0,0        | 1,0        | 0,0        | 3,0        | 5,0        |
| 0,0                           | 0,0        | 0,0        | 2,0        | 2,0        | 0,5        | 4,5                      | 0,0        | 0,0        | 0,0        | 0,0        | 1,5        | 1,5                      | 0,0        | 1,0        | 0,0        | 2,0        | 0,0        | 3,0        |
| 1,0                           | 0,5        | 2,0        | 2,0        | 2,0        | 0,5        | 8,0                      | 0,0        | 0,0        | 2,0        | 3,0        | 2,5        | 7,5                      | 1,0        | 1,0        | 0,5        | 0,0        | 3,5        | 6,0        |
| 1,0                           | 1,0        | 2,0        | 2,0        | 2,0        | 1,5        | 9,5                      | 0,0        | 0,0        | 0,0        | 2,5        | 2,5        | 5,0                      | 0,0        | 1,0        | 0,5        | 1,5        | 4,0        | 7,0        |
| 0,0                           | 0,0        | 0,0        | 2,0        | 0,0        | 0,5        | 2,5                      | 0,5        | 0,0        | 0,0        | 0,5        | 1,0        | 2,0                      | 0,0        | 0,0        | 0,0        | 0,0        | 1,0        | 1,0        |
| 0,0                           | 0,0        | 0,0        | 0,0        | 2,0        | 0,5        | 2,5                      | 0,5        | 0,0        | 0,0        | 1,5        | 2,5        | 4,5                      | 1,0        | 0,0        | 0,0        | 2,5        | 2,0        | 5,5        |
| 1,0                           | 1,0        | 0,0        | 2,0        | 2,0        | 0,5        | 6,5                      | 0,0        | 0,0        | 0,0        | 1,5        | 3,0        | 4,5                      | 0,0        | 1,0        | 0,5        | 2,5        | 4,0        | 8,0        |
| 1,0                           | 0,5        | 0,0        | 2,0        | 0,5        | 0,0        | 4,0                      | 0,0        | 0,0        | 0,0        | 0,0        | 0,5        | 0,5                      | 0,0        | 1,0        | 0,0        | 0,0        | 0,0        | 1,0        |
| 0,0                           | 0,0        | 0,0        | 0,0        | 2,0        | 1,5        | 3,5                      | 0,0        | 0,0        | 0,0        | 0,0        | 1,5        | 1,5                      | 1,0        | 1,0        | 0,0        | 2,0        | 3,0        | 7,0        |
| 1,0                           | 0,5        | 2,0        | 0,0        | 2,0        | 2,0        | 7,5                      | 0,5        | 0,0        | 2,0        | 3,0        | 2,5        | 8,0                      | 1,0        | 1,0        | 0,5        | 2,5        | 3,5        | 8,5        |
| 0,0                           | 0,0        | 0,0        | 0,0        | 2,0        | 0,5        | 2,5                      | 0,0        | 0,0        | 0,0        | 0,0        | 0,0        | 0,0                      | 1,0        | 1,0        | 0,5        | 0,5        | 0,5        | 3,5        |
| 1,0                           | 1,0        | 2,0        | 2,0        | 2,0        | 1,5        | 9,5                      | 0,5        | 0,0        | 2,0        | 3,0        | 2,0        | 7,5                      | 1,0        | 1,0        | 1,0        | 3,0        | 4,0        | 10,0       |
| 1,0                           | 0,0        | 2,0        | 0,0        | 2,0        | 1,5        | 6,5                      | 0,0        | 0,0        | 0,0        | 0,0        | 0,0        | 0,0                      | 0,0        | 0,0        | 0,0        | 2,0        | 1,0        | 3,0        |
| 1,0                           | 0,5        | 2,0        | 2,0        | 2,0        | 1,5        | 9,0                      | 0,5        | 0,0        | 2,0        | 3,0        | 3,0        | 8,5                      | 1,0        | 1,0        | 0,5        | 2,5        | 4,0        | 9,0        |
| 0,0                           | 0,0        | 0,0        | 2,0        | 2,0        | 1,0        | 5,0                      | 0,0        | 1,0        | 0,0        | 0,0        | 1,5        | 2,5                      | 0,0        | 1,0        | 1,0        | 2,0        | 2,0        | 6,0        |
| 0,0                           | 0,0        | 2,0        | 2,0        | 2,0        | 1,5        | 7,5                      | 0,0        | 0,0        | 0,0        | 0,0        | 2,0        | 2,0                      | 0,0        | 0,0        | 0,5        | 0,0        | 0,0        | 0,5        |
| <b>0,6</b>                    | <b>0,3</b> | <b>0,9</b> | <b>1,3</b> | <b>1,8</b> | <b>1,1</b> | <b>6,0</b>               | <b>0,1</b> | <b>0,1</b> | <b>0,5</b> | <b>1,2</b> | <b>1,8</b> | <b>3,7</b>               | <b>0,5</b> | <b>0,6</b> | <b>0,4</b> | <b>1,4</b> | <b>1,9</b> | <b>4,8</b> |
| 58                            | 26         | 47         | 63         | 91         | 55         | 60                       | 13         | 11         | 25         | 39         | 61         | 37                       | 47         | 63         | 39         | 46         | 48         | 48         |
| 0,58 0,26 0,47 0,63 0,91 0,55 |            |            |            |            |            | 0,13 0,11 0,25 0,39 0,61 |            |            |            |            |            | 0,47 0,63 0,39 0,46 0,48 |            |            |            |            |            |            |

| 67.1       | 67.2       | 67.3       | 67.4       | 67.5       | 67.6       | 67.7       | 67.8       | 67.9       | 67.10      | 67.11      | Summe       |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-------------|
| 1          | 1          | 1          | 1          | 1          | 1          | 2          | 3          | 2          | 4,5        | 2,5        | 20,0        |
| 0,5        | 1,0        | 0,0        | 1,0        | 0,0        | 0,0        | 1,0        | 0,5        | 0,5        | 1,5        | 2,0        | 8,0         |
| 0,5        | 0,5        | 0,5        | 0,0        | 1,0        | 0,5        | 1,5        | 3,0        | 0,0        | 1,5        | 2,0        | 11,0        |
| 0,0        | 1,0        | 0,5        | 1,0        | 1,0        | 1,0        | 2,0        | 2,0        | 2,0        | 3,5        | 2,5        | 16,5        |
| 1,0        | 0,0        | 0,5        | 0,0        | 1,0        | 1,0        | 0,0        | 2,5        | 1,5        | 4,5        | 1,5        | 13,5        |
| 0,5        | 1,0        | 1,0        | 1,0        | 1,0        | 1,0        | 2,0        | 3,0        | 2,0        | 4,0        | 2,0        | 18,5        |
| 0,5        | 1,0        | 0,0        | 0,0        | 1,0        | 1,0        | 2,0        | 3,0        | 2,0        | 2,5        | 1,5        | 14,5        |
| 0,5        | 1,0        | 1,0        | 1,0        | 1,0        | 1,0        | 2,0        | 3,0        | 2,0        | 4,0        | 2,0        | 18,5        |
| 1,0        | 0,0        | 0,0        | 1,0        | 1,0        | 1,0        | 1,0        | 0,5        | 1,0        | 2,0        | 1,5        | 10,0        |
| 0,5        | 0,5        | 0,0        | 1,0        | 1,0        | 1,0        | 2,0        | 2,0        | 1,0        | 4,5        | 2,5        | 16,0        |
| 0,0        | 0,5        | 1,0        | 0,0        | 1,0        | 1,0        | 2,0        | 3,0        | 2,0        | 4,5        | 0,5        | 15,5        |
| 0,0        | 0,0        | 0,5        | 0,0        | 0,0        | 0,0        | 2,0        | 0,5        | 1,0        | 4,0        | 2,5        | 10,5        |
| 0,0        | 1,0        | 0,5        | 1,0        | 0,0        | 1,0        | 2,0        | 3,0        | 2,0        | 4,5        | 2,5        | 17,5        |
| 0,0        | 0,5        | 1,0        | 1,0        | 1,0        | 1,0        | 2,0        | 2,5        | 2,0        | 4,5        | 2,0        | 17,5        |
| 0,0        | 0,5        | 0,0        | 1,0        | 1,0        | 0,5        | 0,0        | 0,0        | 0,0        | 2,5        | 2,5        | 8,0         |
| 1,0        | 0,5        | 0,5        | 1,0        | 1,0        | 1,0        | 2,0        | 3,0        | 2,0        | 4,5        | 2,0        | 18,5        |
| 0,0        | 0,5        | 1,0        | 0,0        | 1,0        | 0,0        | 2,0        | 3,0        | 2,0        | 4,0        | 2,5        | 16,0        |
| 1,0        | 0,5        | 1,0        | 1,0        | 1,0        | 1,0        | 2,0        | 3,0        | 2,0        | 4,5        | 2,5        | 19,5        |
| 1,0        | 1,0        | 0,5        | 0,0        | 1,0        | 0,5        | 1,5        | 3,0        | 2,0        | 3,5        | 1,5        | 15,5        |
| 0,5        | 0,5        | 1,0        | 1,0        | 0,0        | 1,0        | 2,0        | 3,0        | 2,0        | 4,5        | 2,5        | 18,0        |
| <b>0,4</b> | <b>0,6</b> | <b>0,6</b> | <b>0,6</b> | <b>0,8</b> | <b>0,8</b> | <b>1,6</b> | <b>2,3</b> | <b>1,5</b> | <b>3,6</b> | <b>2,0</b> | <b>14,9</b> |
| 45         | 61         | 55         | 63         | 79         | 76         | 82         | 76         | 76         | 81         | 81         | 74          |

| 8.1        | 8.2        | 8.3        | 8.4        | Summe      |
|------------|------------|------------|------------|------------|
| 1          | 1          | 1          | 7          | 10,0       |
| 1,0        | 1,0        | 1,0        | 2,0        | 5,0        |
| 1,0        | 1,0        | 0,0        | 4,5        | 6,5        |
| 1,0        | 0,0        | 1,0        | 5,5        | 7,5        |
| 1,0        | 0,5        | 1,0        | 4,0        | 6,5        |
| 1,0        | 1,0        | 1,0        | 3,0        | 6,0        |
| 1,0        | 1,0        | 1,0        | 3,5        | 6,5        |
| 1,0        | 0,0        | 1,0        | 6,5        | 8,5        |
| 0,0        | 0,5        | 1,0        | 3,0        | 4,5        |
| 1,0        | 1,0        | 1,0        | 4,5        | 7,5        |
| 1,0        | 0,0        | 1,0        | 6,0        | 8,0        |
| 1,0        | 0,0        | 1,0        | 3,5        | 5,5        |
| 1,0        | 1,0        | 0,0        | 6,5        | 8,5        |
| 1,0        | 1,0        | 1,0        | 6,5        | 9,5        |
| 0,5        | 0,0        | 1,0        | 0,0        | 1,5        |
| 1,0        | 0,5        | 1,0        | 6,0        | 8,5        |
| 1,0        | 0,0        | 1,0        | 6,0        | 8,0        |
| 1,0        | 1,0        | 1,0        | 5,5        | 8,5        |
| 1,0        | 1,0        | 1,0        | 3,5        | 6,5        |
| 1,0        | 0,5        | 1,0        | 5,0        | 7,5        |
| <b>0,9</b> | <b>0,6</b> | <b>0,9</b> | <b>4,5</b> | <b>6,9</b> |
| 92         | 58         | 89         | 64         | 69         |

| 9.1        | 9.2        | 9.3a       | 9.3b       | Summe      |
|------------|------------|------------|------------|------------|
| 1          | 1          | 4          | 4          | 10,0       |
| 0,0        | 0,0        | 3,5        | 0,0        | 3,5        |
| 0,0        | 1,0        | 3,5        | 0,0        | 4,5        |
| 1,0        | 0,5        | 2,0        | 0,0        | 3,5        |
| 0,5        | 0,5        | 0,0        | 0,0        | 1,0        |
| 1,0        | 1,0        | 2,5        | 0,0        | 4,5        |
| 0,5        | 0,0        | 3,0        | 0,0        | 3,5        |
| 1,0        | 0,5        | 3,5        | 0,0        | 5,0        |
| 0,0        | 1,0        | 3,0        | 0,0        | 4,0        |
| 0,0        | 0,5        | 3,0        | 3,5        | 7,0        |
| 0,5        | 1,0        | 4,0        | 0,0        | 5,5        |
| 1,0        | 1,0        | 3,0        | 0,0        | 5,0        |
| 1,0        | 1,0        | 3,0        | 0,0        | 5,0        |
| 1,0        | 1,0        | 3,5        | 2,5        | 8,0        |
| 1,0        | 0,0        | 3,0        | 0,0        | 4,0        |
| 0,5        | 1,0        | 1,5        | 4,0        | 7,0        |
| 0,5        | 1,0        | 3,0        | 0,0        | 4,5        |
| 0,5        | 1,0        | 4,0        | 4,0        | 9,5        |
| 0,5        | 1,0        | 3,0        | 0,0        | 4,5        |
| 0,0        | 0,0        | 3,5        | 0,0        | 3,5        |
| <b>0,6</b> | <b>0,7</b> | <b>2,9</b> | <b>0,7</b> | <b>4,9</b> |
| 55         | 68         | 73         | 18         | 49         |

0,45 0,61 0,55 0,63 0,79 0,76 0,82 0,76 0,76 0,81 0,81 0,92 0,58 0,89 0,64 0,55 0,68 0,73 0,18

| 10.1       | 10.2       | 10.3       | Summe      |
|------------|------------|------------|------------|
| 2          | 3          | 5          | 10,0       |
| 2,0        | 2,0        | 3,0        | 7,0        |
| 2,0        | 3,0        | 4,0        | 9,0        |
| 2,0        | 2,0        | 4,0        | 8,0        |
| 2,0        | 2,0        | 3,0        | 7,0        |
| 2,0        | 0,0        | 2,5        | 4,5        |
| 2,0        | 3,0        | 5,0        | 10,0       |
| 2,0        | 3,0        | 2,5        | 7,5        |
| 2,0        | 0,0        | 2,0        | 4,0        |
| 2,0        | 3,0        | 4,0        | 9,0        |
| 2,0        | 1,0        | 4,0        | 7,0        |
| 2,0        | 0,0        | 2,5        | 4,5        |
| 2,0        | 3,0        | 2,5        | 7,5        |
| 2,0        | 3,0        | 4,0        | 9,0        |
| 2,0        | 0,0        | 2,0        | 4,0        |
| 2,0        | 3,0        | 3,5        | 8,5        |
| 2,0        | 1,0        | 5,0        | 8,0        |
| 2,0        | 3,0        | 5,0        | 10,0       |
| 2,0        | 0,0        | 1,5        | 3,5        |
| 2,0        | 2,0        | 1,5        | 5,5        |
| <b>2,0</b> | <b>1,8</b> | <b>3,2</b> | <b>7,0</b> |
| 100        | 60         | 65         | 70         |

| 11         | Summe      |
|------------|------------|
| 10         | 10,0       |
| 7,5        | 7,5        |
| 6,5        | 6,5        |
| 2,0        | 2,0        |
| 4,0        | 4,0        |
| 5,5        | 5,5        |
| 9,5        | 9,5        |
| 3,0        | 3,0        |
| 3,5        | 3,5        |
| 8,0        | 8,0        |
| 4,0        | 4,0        |
| 5,5        | 5,5        |
| 8,0        | 8,0        |
| 6,5        | 6,5        |
| 3,5        | 3,5        |
| 9,0        | 9,0        |
| 7,0        | 7,0        |
| 9,5        | 9,5        |
| 4,0        | 4,0        |
| 5,5        | 5,5        |
| <b>5,9</b> | <b>5,9</b> |
| 59         | 59         |


| Summe       | Prozentual  |
|-------------|-------------|
| 110         | 100         |
| 50,5        | 45,9        |
| 67,5        | 61,4        |
| 60,0        | 54,5        |
| 69,5        | 63,2        |
| 58,0        | 52,7        |
| 82,5        | 75,0        |
| 81,5        | 74,1        |
| 40,5        | 36,8        |
| 74,0        | 67,3        |
| 76,5        | 69,5        |
| 42,5        | 38,6        |
| 73,0        | 66,4        |
| 93,0        | 84,5        |
| 37,0        | 33,6        |
| 96,0        | 87,3        |
| 65,0        | 59,1        |
| 100,0       | 90,9        |
| 65,5        | 59,5        |
| 62,5        | 56,8        |
| <b>68,2</b> | <b>62,0</b> |
| 62          | 62          |

| OG72                | max. Punkte |
|---------------------|-------------|
| ANHA                |             |
| ANST                |             |
| BAJÖ                |             |
| BERE                |             |
| BETH                |             |
| BIFR                |             |
| BIHO                |             |
| BITH                |             |
| BIUW                |             |
| CARU                |             |
| CHMI                |             |
| DAUW                |             |
| INWO                |             |
| JUKL                |             |
| LASA                |             |
| MAAC                |             |
| SADI                |             |
| SILO                |             |
| TAWA                |             |
| <b>Durchschnitt</b> |             |
| <b>Prozentual</b>   |             |

1,00 0,60 0,65 0,59 Durchschnitt 0,61 Pi

## O) Nachhaltigkeitstest (Follow-Up-Test) inkl. Löser (Hauptstudie 2007/2008)

Der Nachhaltigkeitstest 2008 (ohne die roten Lösungen) wurde auch als Grundlage des Eingangstests (Pre-Test) verwendet (siehe Anhang Q und R).

|                                                                                   |                                                   |                                           |
|-----------------------------------------------------------------------------------|---------------------------------------------------|-------------------------------------------|
|  | <b>Nachhaltigkeitstest<br/>2008 (inkl. Löser)</b> | <b>Informatik</b><br>Beginn der Kursphase |
| Ehemalige Klasse:                                                                 | Datum:                                            | Blatt Nr.: 1 / 1                          |

|                                                   |          |
|---------------------------------------------------|----------|
|                                                   | Ihr Code |
| <b>Erster Buchstabe des Vornamens der Mutter</b>  |          |
| <b>Zweiter Buchstabe des Vornamens der Mutter</b> |          |
| <b>Erster Buchstabe des Vornamens des Vaters</b>  |          |
| <b>Zweiter Buchstabe des Vornamens des Vaters</b> |          |

In der folgenden Lernerfolgskontrolle werden wieder Fragen zu 9 Themengebieten gestellt.

**ACHTUNG: Die Fragen sehen zwar gleich bzw. ähnlich aus wie die der ersten Lernerfolgskontrolle, sie sind aber größtenteils verändert worden, so dass auch die Lösungen jetzt anders lauten!**

Zu jedem dieser Themengebiete gibt es in der Regel ein Blatt mit einer Mischung aus Ankreuzaufgaben und Fragen, die einer eigenen Lösung bedürfen. In jedem Themengebiet können 10 Punkte erreicht werden.

**Hinweis: Bei den Multiple-Choice-Fragen sind evt. beliebig viele Antworten (aber mindestens eine) richtig!** Fehlerhafte Kreuze führen zu Punktabzug.

Nach diesen Fragen erhalten Sie im Anschluss wieder Zusatzfragen zu dem zehnten und elften Themengebiet. Die dort erzielten Punkte werden wie gewohnt zu den anderen Punkten hinzuaddiert!

| Nr.   | Themengebiet                                 | Erreichbare Punkte | Erreichte Punkte |
|-------|----------------------------------------------|--------------------|------------------|
| 1     | Datentypen und Steuerstruktur Sequenz        | 10                 |                  |
| 2     | Steuerstruktur: Selektion                    | 10                 |                  |
| 3     | Steuerstruktur: Iteration                    | 10                 |                  |
| 4     | Arrays und Strings                           | 10                 |                  |
| 5     | Methoden                                     | 10                 |                  |
| 6 / 7 | Einführung in die OOP / Klasse und Objekt    | 20                 |                  |
| 8     | Vererbung                                    | 10                 |                  |
| 9     | Assoziation                                  | 10                 |                  |
| 10    | Zusatz: dynamische OOP                       | 10                 |                  |
| 11    | Zusatz: Objektorientierte Modellierung (OOM) | 10                 |                  |
| Summe |                                              | 110                |                  |

Vielen Dank für Ihre Mitarbeit!

Ehler 2008
[ehler@oszimt.de](mailto:ehler@oszimt.de)
www.oszimt.de

Ehemalige Klasse:

Datum:

Blatt Nr.: 2 / 2

**1. Datentypen und Steuerstruktur Sequenz**

|    |                                                                                                                           |                                                                                                                                                                                                                                                                                                                                   |           |
|----|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 1. | Durch den <b>Datentyp</b> einer Variablen wird festgelegt, welche(n) ...                                                  | <input checked="" type="checkbox"/> Werte die Variable annehmen kann<br><input type="checkbox"/> Namen die Variable hat<br><input type="checkbox"/> Adresse die Variable im Speicher hat<br><input type="checkbox"/> Wert die Variable hat<br><input checked="" type="checkbox"/> Anzahl von Bits die Variable im Speicher belegt | <b>1P</b> |
| 2. | Gegeben sind die beiden Quelltextzeilen:<br><pre>final double PI; PI = 3.1415;</pre> Welche Aussage(n) trifft/treffen zu? | <input type="checkbox"/> Compilerfehler in Zeile 1<br><input type="checkbox"/> Compilerfehler in Zeile 2<br><input checked="" type="checkbox"/> PI ist eine Konstante<br><input type="checkbox"/> PI muss klein geschrieben werden                                                                                                | <b>1P</b> |
| 3. | Welche Werte kann eine Variable vom <b>Datentyp short</b> annehmen?                                                       | <input type="checkbox"/> 0 ... 65.535<br><input checked="" type="checkbox"/> -32.768 bis 32.767<br><input type="checkbox"/> -32.767 bis 32.768<br><input type="checkbox"/> keine Aussage trifft zu                                                                                                                                | <b>1P</b> |
| 4. | Gegeben sind die beiden Quelltextzeilen:<br><pre>byte a = 7; int b = a;</pre> Welche Aussage(n) trifft/treffen zu?        | <input type="checkbox"/> Es gibt einen Compilerfehler<br><input checked="" type="checkbox"/> Es gibt keinen Compilerfehler<br><input type="checkbox"/> b belegt 16 Bit<br><input checked="" type="checkbox"/> b belegt 32 Bit                                                                                                     | <b>1P</b> |

5. Gegeben ist die folgende lückenhafte **Variablendeklaration**: **3P**

```
char    a = 'a';

float   b = 7.0f;           3 * 1P

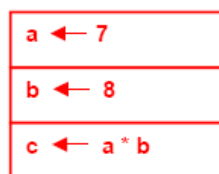
boolean c = (5 == 7);
```

**Ergänzen** Sie den **Quelltext** in den 3 Lücken!

6. Gegeben sind die drei Quelltextzeilen:

```
byte a = 7;
int b = 8;
double c;
c = a * b;
```

**Zeichnen** Sie das dazugehörige **Struktogramm inkl. Einträge!** **3P**



**3 \* 1P**  
**ohne Einträge: 1P**

2. Steuerstruktur: Selektion

|    |                                                                                                                                                          |                                                                                                                                                                                                                     |    |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1. | Gegeben ist der folgende Vergleich:<br>if (!isRichtig) {}<br>Dies kann man auch wie folgt schreiben:                                                     | <input type="radio"/> if (isRichtig = true) {}<br><input type="radio"/> if (isRichtig = false) {}<br><input type="radio"/> if (isRichtig == true) {}<br><input checked="" type="radio"/> if (isRichtig == false) {} | 1P |
| 2. | <pre>int b = 0, c = 7; if (c &lt; 7)   if (c &gt;= 7) b = 1;   else      b = 2; else   if (c &gt; 7) b = 3; </pre> Was steht am Ende in der Variablen b? | <input checked="" type="radio"/> 0<br><input type="radio"/> 1<br><input type="radio"/> 2<br><input type="radio"/> 3<br><input type="radio"/> 4                                                                      | 1P |

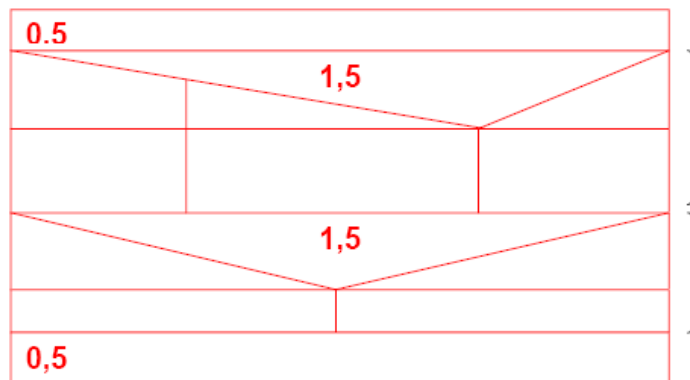
3. Gegeben ist der folgende Quelltextausschnitt:

```
int a = 4; // oder 6 oder 8
switch (a) {
  case 4:
    a = 3 * a;
    break;
  case 6:
    a = a / 5;
    break;
  default:
    a = a % 3; // modulo
    break;
} // switch
if (a < 10)
  a--;
else
  a++;
System.out.println("a: " + a);
```

| a | Ausgabe                |
|---|------------------------|
| 4 | 13 (1P)                |
| 6 | 0 (1,5),<br>0,2 (0,5P) |
| 8 | 1 (1,5P)               |

a) Welcher Wert wird ausgegeben, bei den folgenden ursprünglichen Werten für die Variable a (3 Tabelleneinträge)? **4P**

b) Zeichnen Sie das dazugehörige Struktogramm! **4P**  
Hinweis: Einträge in das Struktogramm sind **nicht** erforderlich!



|                   |        |                  |
|-------------------|--------|------------------|
| Ehemalige Klasse: | Datum: | Blatt Nr.: 4 / 4 |
|-------------------|--------|------------------|

### 3. Steuerstruktur: Iteration

|    |                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                       |    |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1. | Gegeben ist die folgende Quelltextzeile:<br><code>for (int i = 1; i &lt; 7; i++) { }</code><br>Diese <b>Schleife</b> wird im <b>Struktogramm</b> mit Hilfe welchen Strukturelements nach DIN 66261 dargestellt? | <input checked="" type="radio"/> kopfgesteuerte Schleife<br><input type="radio"/> fußgesteuerte Schleife<br><input type="radio"/> Endlosschleife<br><input type="radio"/> keiner der genannten Schleifen                                              | 1P |
| 2. | Welche Antwort(en) enthält/enthalten <b>keine</b> Endlosschleife?                                                                                                                                               | <input type="radio"/> <code>for ( ; ; ) { }</code><br><input checked="" type="radio"/> <code>do { } while (false);</code><br><input checked="" type="radio"/> <code>while (!true) { }</code><br><input type="radio"/> <code>while (9 == 9) { }</code> | 1P |

3. Gegeben ist der folgende Quelltextausschnitt:

```
int a = 4 * 8;
int b = a / 2;
while (b > 2) {
    a = a - 3;
    b = b - 2;
}
```

Welcher Wert steht am Ende des Algorithmus in der Variable **a**?

a  
11

2P

4. Gegeben ist der folgende Quelltextausschnitt:

```
int i = (4 * 3) * 3;
do {
    i++;
    i = i - 4;
} while (i > 6);
```

9 oder 11: 0,5P  
Anzahl

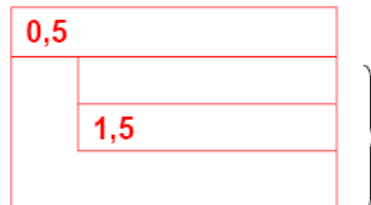
10

2P  
2P

Wie oft wird der Schleifenkörper durchlaufen?

Zeichnen Sie das **Struktogramm**!

Hinweis: Einträge in das Struktogramm sind **nicht** erforderlich!



5. Ergänzen Sie den folgenden **Quelltext**, so dass als Ausgabe: „24,12,6,“ erscheint! 2P

```
for (int z = 24; z >= 6; z = z/2)
    System.out.print(z + ",");
```

#### 4. Arrays und Strings

|    |                                                                                                                                                   |                                                                                                                                                                                                                                              |    |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1. | Gegeben ist die folgende Variablendeklaration:<br>String s = "Test";<br>String t = new String("Test");<br>Welche Bedingungen liefern true zurück? | <input checked="" type="checkbox"/> if (s == "Test") {}<br><input type="checkbox"/> if (t == s) {}<br><input checked="" type="checkbox"/> if (s.equals("Test")) {}<br><input checked="" type="checkbox"/> if (t.equals(s)) {}                | 1P |
| 2. | int[] x = new int[25];<br>Welche Aussage(n) trifft/treffen nach der Ausführung der Programmzeile zu?                                              | <input checked="" type="checkbox"/> x.length hat den Wert 25<br><input type="checkbox"/> x[0] hat den Wert null<br><input checked="" type="checkbox"/> x[24] hat den Wert 0<br><input checked="" type="checkbox"/> x[25] ist nicht definiert | 1P |

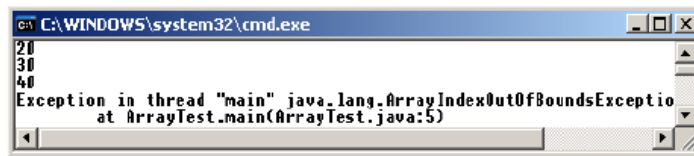
Ein Kreuz richtig: 0P, zwei Kreuze richtig: 0,5P

3. Gegeben ist der folgende Quelltextausschnitt: 3P

```
int[] exampleArray = {10, 20, 30, 40};
for (int i = 1; i <= exampleArray.length; i++)
    System.out.println(exampleArray[i]);
```

Wie lautet die **Bildschirmausgabe**?

10 bis 40: 1P



4. Initialisieren Sie das folgende Feld **mit Hilfe einer Zählschleife** mit den Werten 'a', 'b' und 'c!' 3P

```
char[] zeichenfeld = new char[3];

zeichen = 'a';
for (int i = 0; i < 3; i++)
    zeichenfeld[i] = zeichen++;
```

5. Gegeben ist der folgende Quelltext: 2P

```
String s1 = new String("berlin");
s1.toUpperCase();
String s2 = null;
System.out.println(s1 + s2 + s1.length());
```

Wie lautet die Bildschirmausgabe?



Ehemalige Klasse:

Datum:

Blatt Nr.: 6 / 6

### 5. Methoden

|    |                                                                                                                                              |                                                                                                                                                                                                                                                                |           |
|----|----------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 1. | Beim <b>Aufruf einer Methode</b> findet man immer ...                                                                                        | <input checked="" type="checkbox"/> einen Methodennamen<br><input type="checkbox"/> einen aktuellen Parameter<br><input type="checkbox"/> einen formalen Parameter<br><input type="checkbox"/> einen Rückgabewert                                              | <b>1P</b> |
| 2. | Welche Aussage(n) trifft/treffen für die <b>Methode berechne()</b> zu?<br><code>if (berechne() == 4.7) {}</code>                             | <input type="checkbox"/> Sie hat einen boolean-Rückgabewert<br><input checked="" type="checkbox"/> Sie hat einen double-Rückgabewert<br><input type="checkbox"/> Sie hat keinen Rückgabewert                                                                   | <b>1P</b> |
| 3. | Gegeben ist die folgende <b>Methode tueEtwas(...)</b> :<br><code>int a;<br/>a = tueEtwas(4.7);</code><br>Welche Aussage(n) ist/sind richtig? | <input type="checkbox"/> Die Methode hat keinen Rückgabewert<br><input type="checkbox"/> a muss vom Datentyp double sein<br><input type="checkbox"/> Vor dem Wert 4.7 fehlt der Datentyp double<br><input checked="" type="checkbox"/> keine Aussage trifft zu | <b>1P</b> |

4. Gegeben sind die folgenden Quelltextzeilen des Anwendungsprogramms, die in der main-Methode stehen: **3P**
- ```
char[] zeichenfeld = new char[10];
initialisiereFeld(zeichenfeld);
anzahl = ermittleHaeufigkeit(zeichenfeld, 'c');
```
- Wie lautet der **Methodenkopf** der statischen Methode „ermittleHaeufigkeit(...)“, welcher im Anwendungsprogramm „über“ der main-Methode steht?  
**Hinweis: Nur eine Quelltextzeile ist gefragt!**
- 0,5P
0,5P
0,5P
1P
0,5P
- ```
public static int ermittleHaeufigkeit(char[] feld, char c)
```

5. In der Klasse „Formelsammlung“ soll eine Methode „berechneFlaeche“ **4P** implementiert werden, die als formale Parameter eine laenge (double) und breite (double) hat und die berechnete flaeche zurückliefert. Ist einer der beiden Parameter negativ, wird 0 zurück gegeben.
- Geben Sie den vollständigen Quelltext der Methode „berechneFlaeche“ (Methodenkopf und Implementierung) an!**
- 0,5P
0,5P
0,5P
- ```
public double berechneFlaeche(double laenge, double breite) {
    if (laenge > 0 && breite > 0)
        return laenge * breite;
    return 0;
}
```
- 0,5P
1,0P
0,5P



Ehemalige Klasse:

Datum:

Blatt Nr.: 7 / 7

6. Einführung in die OOP / 7. Klasse und Objekt

1.	Welche Aussage(n) über <b>Klassen</b> sind richtig?	<input type="radio"/> Eine Klasse ist in einem bestimmten Zustand <input type="radio"/> Eine Klasse ist die Summe der Objekte <input type="radio"/> Eine Klasse hat immer Attribute <input checked="" type="radio"/> Keine Aussage ist richtig <b>1P</b>
2.	Welche Aussage(n) über <b>Objekte</b> ist/sind richtig?	<input type="radio"/> Von jeder Klasse gibt es mindestens ein Objekt <input checked="" type="radio"/> Objekte können mit anderen Objekten Botschaften austauschen <input checked="" type="radio"/> Objekte befinden sich in einem Zustand <input checked="" type="radio"/> Objekte haben einen Zugriff auf sich selbst <b>1P</b>
3.	Welche Aussage(n) zu <b>Attributen</b> ist/sind richtig?	<input checked="" type="radio"/> Attribute sind vergleichbar mit Variablen <input type="radio"/> Ein Objekt deklariert Attribute <input checked="" type="radio"/> Attribute haben einen Datentyp <input checked="" type="radio"/> Attribute sind in der Regel privat <b>1P</b>
4.	<b>Methoden</b> werden im UML-Objektdiagramm ...	<input type="radio"/> angegeben <input checked="" type="radio"/> nicht angegeben <input type="radio"/> je nach Bedarf angegeben <b>1P</b>
5.	Wozu braucht man die <b>Verwaltungsmethoden</b> („Setter“ und „Getter“)?	<input type="radio"/> Weil diese immer deklariert werden müssen <input type="radio"/> Weil die Attribute in der Regel öffentlich sind <input checked="" type="radio"/> Weil die Attribute in der Regel privat sind <b>1P</b>
6.	Wozu dienen <b>Konstruktoren</b> ?	<input type="radio"/> Um eine Vererbungshierarchie zu ermöglichen <input type="radio"/> Um eine Klasse zu initialisieren <input checked="" type="radio"/> Um ein Objekt zu initialisieren <input type="radio"/> Um ein Klasse deklarieren zu können <b>1P</b>

7. Die Klasse „Auto“ hat die Attribute „leistung“ (Datentyp int) und „hersteller“ (Datentyp String). **2P**

Wie lautet der **Methodenkopf** des **vollparametrisierten Konstruktors**?

**Hinweis:** Nur **eine Quelltextzeile** ist gefragt!

**0,5P 0,5P 0,5P 0,5P**

```
public Auto(int leistung, String hersteller)
```

8. Die Klasse „Bruecke“ hat u. a. das Attribut „breite“ vom Datentyp „float“. **3P**

Wie lautet der vollständige Quelltext der „**Setter**“-**Verwaltungsmethode**?

**0,5P 0,5P 1P**

```
public void setBreite(float breite) {
    this.breite = breite;    1P
}
```

Ehemalige Klasse:

Datum:

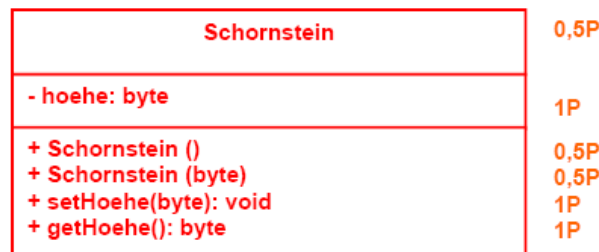
Blatt Nr.: 8 / 8

9. Die Klasse „Code“ hat u. a. das Attribut „character“ vom Datentyp „char“. 2P  
Wie lautet der vollständige Quelltext der „Getter“-Verwaltungsmethode?

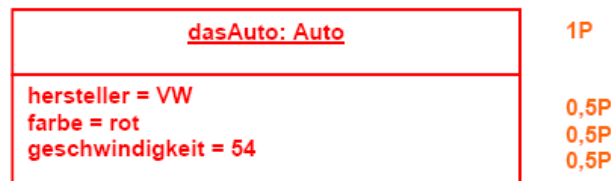
```

0,5P  0,5P
public char getCharacter() {
    return this.character;
}      0,5P  0,5P
    
```

10. Die Klasse „Schornstein“ hat das Attribut „hoehe“ (Datentyp byte). 4,5P  
Zeichnen Sie das vollständige **UML-OOD-Klassendiagramm** inklusive dem parameterlosen und dem parametrisierten Konstruktor und allen Verwaltungsoperationen („Setter“ und „Getter“)!



11. Ein konkretes Auto von VW ist rot und fährt gerade 54km/h. 2,5P  
Zeichnen Sie das **UML-Objektdiagramm** dieses Autos!



Ehemalige Klasse:

Datum:

Blatt Nr.: 9 / 9

### 8. Vererbung

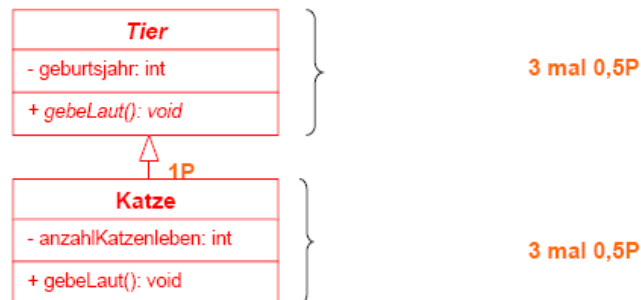
1.	Eine <b>Unterklasse</b> hat <b>3 Attribute</b> . Wie viele Attributwerte hat ein Objekt der <b>Oberklasse</b> ?	<input type="radio"/> maximal 3 <input type="radio"/> genau 3 <input type="radio"/> mindestens 3 <input checked="" type="radio"/> unbekannt, da Oberklasse von Unterklasse unabhängig ist <b>1P</b>
2.	Welche Aussage(n) ist/sind richtig? Eine <b>abstrakte Klasse</b> ...	<input type="radio"/> kann keine Oberklasse sein <input type="radio"/> kann keine Attribute haben <input type="radio"/> kann keine Methoden haben <input checked="" type="radio"/> keine Antwort richtig <b>1P</b>

3. Gegeben ist der folgende Quelltextausschnitt:

```
public abstract class Tier {
    private int geburtsjahr;
    public abstract void gebeLaut();
} // Tier

public class Katze extends Tier {
    private int anzahlKatzenleben;
    public void gebeLaut() {
        System.out.println("Miau");
    }
} // Katze
```

a) Zeichnen Sie das **UML-Klassendiagramm** (nur mit den gegebenen Attributen und Methoden, also ohne setter/getter und Konstruktoren) **4P**



b) Schreiben Sie den **Quelltext** des **vollparametrisierten Konstruktor** der Klasse **Katze**! Hinweis: Die Oberklasse **Tier** hat einen parametrisierten Konstruktor. **4P**

```
public Katze(int geburtsjahr, int anzahlKatzenleben) {
    super(geburtsjahr);
    this.anzahlKatzenleben = anzahlKatzenleben;
} // Konstruktor
```

Ehemalige Klasse:

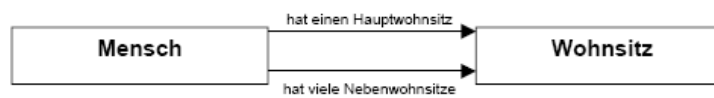
Datum:

Blatt Nr.: 10 / 10

9. Assoziation (inkl. Klassenattribute und Klassenmethoden)

1.	<b>Klassenattribute</b> werden im <b>UML-Klassendiagramm</b> ...	<input type="radio"/> kursiv geschrieben <input checked="" type="radio"/> unterstrichen <input type="radio"/> mit dem Wort <code>static</code> versehen	<b>1P</b>
2.	Welche Aussage(n) über <b>Klassenmethoden</b> (statische Methoden) trifft/treffen zu?	<input type="radio"/> Sie besitzen immer ein Klassenattribut <input type="radio"/> Sie haben nie einen Rückgabewert <input type="radio"/> Aufruf erfolgt in der Regel mit: <code>objektname.methodenname()</code> <input checked="" type="radio"/> Keine Aussage trifft zu	<b>1P</b>

3. Gegeben ist das folgende Szenario:



**Ergänzen** Sie den folgenden **Quelltext!**

**3P**

```

public class Mensch {
    //Attribute
    private String name;

    // Realisierung der Assoziationen
    1,5P
    private Wohnsitz hauptwohnsitz;
    1,5P
    private Wohnsitz[] nebenwohnsitze = new Wohnsitz[10];
    3P auch ohne private und new möglich ...
} // Klasse Mensch
  
```

4. Die Klasse Baum soll ihre Objekte kennen (max. 10).

**5P**

**Ergänzen** Sie den folgenden **Quelltext!**

```

public class Baum {
    //Attribute
    private String art;

    // Klassenattribute
    private static anzahlBaeume = 0;

    private static Baum[] baeume= new Baum[10]; 2P

    public Baum() {
        art = "Unbekannt";

        baeume[anzahlBaeume] = this; 2P
        anzahlBaeume++; 1P
    } // Konstruktor
} // Klasse Baum
  
```

**10. Objektorientierte Programmierung** (dynamisch):

Gegeben ist der folgende Quelltext (zur Übersichtlichkeit wurden die Klassen `Mensch`, `Sattel` und `Pferd` in das Anwendungsprogramm integriert):

```
public class DynamischeOOP {

    public static class Mensch {

        private String name;

        public Mensch() {
            this("Anna");
        }

        public Mensch(String name) {
            this.name = name;
        }

        public void setName(String name) {
            this.name = name;
        }

        public String getName() {
            return this.name;
        }

    } // Klasse Mensch

    public static class Sattel {

        String farbe;

        public Sattel() {
            this("gelb");
        }

        public Sattel(String farbe) {
            this.farbe = farbe;
        }

        public void setFarbe(String farbe) {
            this.farbe = farbe;
        }

        public String getFarbe() {
            return this.farbe;
        }

    } // Klasse Sattel
```

Ehemalige Klasse:

Datum:

Blatt Nr.: 12 / 12

```
public static class Pferd {  
  
    private static int anzahlPferde = 0;  
    private static Pferd[] pferde = new Pferd[10];  
    private String name;  
    private Mensch besitzer;  
    private Mensch reiter;  
    private Sattel sattel;  
  
    public Pferd() {  
        this("Rico");  
    }  
  
    public Pferd(String name) {  
        this.name = name;  
        pferde[anzahlPferde] = this;  
        anzahlPferde++;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return this.name;  
    }  
  
    public static int getAnzahlPferde() {  
        return anzahlPferde;  
    }  
  
    public void setReiter(Mensch reiter) {  
        this.reiter = reiter;  
    }  
  
    public Mensch getReiter() {  
        return this.reiter;  
    }  
  
    public void setBesitzer(Mensch besitzer) {  
        this.besitzer = besitzer;  
    }  
  
    public Mensch getBesitzer() {  
        return this.besitzer;  
    }  
  
    public Sattel getSattel() {  
        return this.sattel;  
    }  
  
    public static Pferd getPferde(int i) {  
        return pferde[i];  
    }  
  
    public void verkaufe(Mensch neuerBesitzer) {  
        this.besitzer = neuerBesitzer;  
    }  
}
```

Ehemalige Klasse:

Datum:

Blatt Nr.: 13 / 13

```
public void sattele(Sattel sattel) {
    this.sattel = sattel;
}

} // Klasse Pferd

public static void main(String[] args) {

    Mensch ersterMensch = new Mensch();
    Mensch zweiterMensch = new Mensch();
    Mensch dritterMensch = new Mensch("Berta");

    ersterMensch.setName("Irene");
    dritterMensch.setName("Susi");

    Pferd erstesPferd = new Pferd("Smeralda");
    Pferd zweitesPferd = new Pferd();
    Pferd drittesPferd = zweitesPferd;

    Sattel ersterSattel = new Sattel();
    Sattel zweiterSattel = new Sattel("gruen");
    Sattel dritterSattel = new Sattel("blau");

    erstesPferd.setBesitzer(dritterMensch);
    zweitesPferd.setBesitzer(ersterMensch);
    drittesPferd.setBesitzer(zweiterMensch);

    erstesPferd.sattele(zweiterSattel);
    zweitesPferd.sattele(dritterSattel);
    drittesPferd.sattele(ersterSattel);

    erstesPferd.setReiter(ersterMensch);
    zweitesPferd.setReiter(zweiterMensch);
    drittesPferd.setReiter(dritterMensch);

    erstesPferd.verkaufe(ersterMensch);
    zweiterSattel.setFarbe("rot");
    erstesPferd.setName("Leo");

    for (int i = 0; i < Pferd.getAnzahlPferde(); i++) {
        System.out.println("Pferd: " + Pferd.getPferde(i).getName());
        System.out.println("Besitzer: " + Pferd.getPferde(i).getBesitzer().getName());
        System.out.println("Reiter: " + Pferd.getPferde(i).getReiter().getName());
        System.out.println("Sattel: " + Pferd.getPferde(i).getSattel().getFarbe() + "\n");
    } // for

} // main

} // Klasse DynamischeOOP
```

Ehemalige Klasse:

Datum:

Blatt Nr.: 14 / 14

1. **Wie viele Pferde** werden zur Laufzeit durch die main-Methode erzeugt? **2P**

**2 Pferde 2P (3 Pferde 1P)**

2. Wie lautet genau die **Bildschirmausgabe**? **8P**



```
C:\WINDOWS\system32\cmd.exe
Pferd: Leo
Besitzer: Irene
Reiter: Irene
Sattel: rot

Pferd: Rico
Besitzer: Anna
Reiter: Susi
Sattel: gelb

G:\Dissertation\Nachhaltigkeitstest>Pause
```

**8 \* 1P**

3. Ausgabe für 3. Pferd: -2P



Ehemalige Klasse:

Datum:

Blatt Nr.: 15 / 15

### 11. Objektorientierte Modellierung (OOM):

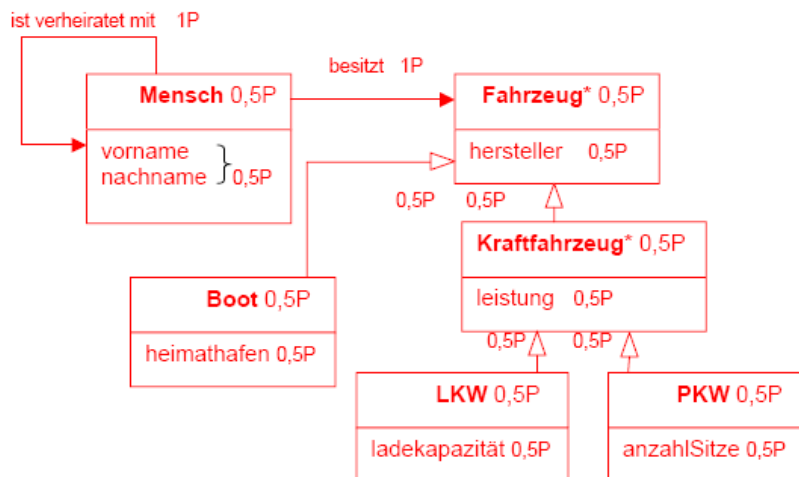
Gegeben ist das folgende Szenario:

10P

Menschen können Fahrzeuge besitzen und Ehepartner haben, z.B.:  
 Maria Müller besitzt ein konkretes Boot (Hersteller: C&C), welches den Heimathafen in Berlin hat. Peter Meyer besitzt einen konkreten PKW (Hersteller: VW) mit 4 Sitzen und 95PS.  
 Rainer Müller besitzt einen LKW (Hersteller: Volvo) mit 850PS und einer Ladekapazität von 40 Tonnen. Maria Müller ist mit Rainer Müller verheiratet.

**Modellieren** Sie mit Hilfe eines **UML-Klassendiagramms** (inkl. Vererbungen und Assoziationen) dieses Szenario, indem Sie die konkreten Daten versuchen zu generalisieren!

**Hinweis:** Es genügt die **OOA-Ansicht**, also ohne setter, getter, Konstruktoren und Datentypen. Es müssen also nur **Klassennamen** und **Attributnamen** angeführt werden, aber keinerlei Methoden und Datentypen!



\*auch als abstrakte Klasse richtig

## P) Ergebnisse des Nachhaltigkeitstests (Hauptstudie 2007/2008)

OG71 (OOP-First): Gesamtergebnis

OG71	CODE	1	2	3	4	5	67	8	9	10	11	Summe	S: 1-8
1	ANDA												
2	ANFR	7,0	8,0	5,5	2,0	4,0	11,5	4,5	1,0	7,0	1,0	51,5	42,5
3	ANPE	8,5	9,0	6,5	6,5	7,5	16,0	8,5	2,5	5,0	8,0	78,0	62,5
4	ARST	6,0	7,5	9,0	4,5	5,0	17,0	8,5	1,0	4,0	8,0	70,5	57,5
5	CLUL	4,5	6,5	3,0	0,5	5,5	7,5	2,0	1,0	4,0	1,0	35,5	29,5
6	COVO	8,0	9,0	6,5	5,0	9,0	15,5	5,0	5,5	5,0	7,0	75,5	58,0
7	DAHA	7,5	9,0	5,5	7,0	8,5	15,0	9,0	3,5	5,0	8,0	78,0	61,5
8	EWAD	4,5	6,0	3,0	3,0	3,0	15,0	7,5	1,0	4,0	9,0	56,0	42,0
9	JAKR	8,5	7,5	5,0	3,5	3,5	14,5	7,0	1,0	7,0	7,5	65,0	49,5
10	JETH	7,0	6,5	7,0	5,5	7,0	17,5	8,0	4,5	3,0	4,0	70,0	58,5
11	LIFR	4,0	4,5	5,0	2,0	5,0	13,0	6,0	0,0	8,0	7,0	54,5	39,5
12	LINO	5,0	2,0	7,5	3,0	0,5	1,0	1,0	0,0	5,0	0,0	25,0	20,0
13	MARE	4,0	4,5	4,0	0,5	2,0	13,0	6,5	0,0	3,0	2,0	39,5	34,5
14	NAGO												
15	NUYU	8,5	8,5	7,5	2,5	9,5	20,0	8,0	9,0	5,0	7,5	86,0	64,5
16	PERA	6,0	4,5	1,0	4,0	4,0	11,0	2,0	1,0	3,0	2,5	39,0	32,5
17	PETH												
18	REDE	7,0	3,5	3,5	0,0	3,5	10,0	0,5	0,0	7,0	2,0	37,0	28,0
19	SIJO	7,5	9,0	9,5	7,5	7,0	17,5	7,5	8,5	7,0	8,5	89,5	65,5
20	SIPE	2,5	0,0	2,0	1,0	1,5	8,5	1,5	0,0	5,0	4,5	26,5	17,0
21	SIVO	8,5	7,5	7,0	4,5	6,5	14,0	8,5	1,0	7,0	9,5	74,0	56,5
22	SUNO	6,5	10,0	9,5	6,0	8,0	17,5	8,5	7,0	7,0	4,0	84,0	66,0
	<b>Durchschnitt</b>	<b>6,4</b>	<b>6,5</b>	<b>5,7</b>	<b>3,6</b>	<b>5,3</b>	<b>13,4</b>	<b>5,8</b>	<b>2,5</b>	<b>5,3</b>	<b>5,3</b>	<b>59,7</b>	<b>46,6</b>
	<b>Prozentual</b>	<b>64</b>	<b>65</b>	<b>57</b>	<b>36</b>	<b>53</b>	<b>67</b>	<b>58</b>	<b>25</b>	<b>53</b>	<b>53</b>	<b>54</b>	<b>58</b>

Die Summe 59,7 bzw. die Teilsumme (Aufgaben 1 bis 8) 46,6 wurde von Excel aus den einzelnen Teilergebnissen mit vielen Nachkommastellen berechnet. Summiert man die gerundeten Zahlen kommt man auf 59,8 bzw. 46,7.

OG71 (OOP-First): Teilergebnisse inkl. Itemschwierigkeiten

OG71	1.1	1.2	1.3	1.4	1.5	1.6	Summe	2.1	2.2	2.3a	2.3b	Summe
max. Punkte	1,0	1,0	1,0	1,0	3,0	3,0	10,0	1	1	4	4	10,0
ANFR	1	1	0	0	2	3	7,0	0,0	1,0	4,0	3,0	8,0
ANPE	1	1	1	0	3	2,5	8,5	1,0	1,0	4,0	3,0	9,0
ARST	0,5	1	0	0	2	2,5	6,0	1,0	0,0	2,5	4,0	7,5
CLUL	1	0	0	0	1,5	2	4,5	0,0	1,0	2,5	3,0	6,5
COVO	1	1	0	0,5	3	2,5	8,0	1,0	1,0	4,0	3,0	9,0
DAHA	1	0,5	1	0,5	3	1,5	7,5	1,0	0,0	4,0	4,0	9,0
EWAD	1	0	0	0,5	1	2	4,5	1,0	0,0	1,0	4,0	6,0
JAKR	1	1	1	0,5	2	3	8,5	1,0	1,0	2,5	3,0	7,5
JETH	1	1	0	0	3	2	7,0	1,0	1,0	1,5	3,0	6,5
LIFR	1	0,5	0	0	1,5	1	4,0	0,5	0,0	2,5	1,5	4,5
LINO	1	1	0	0	0	3	5,0	0,0	1,0	1,0	0,0	2,0
MARE	0	0,5	0	0	1	2,5	4,0	1,0	0,0	1,5	2,0	4,5
NUYU	1	0,5	1	0,5	3	2,5	8,5	1,0	1,0	2,5	4,0	8,5
PERA	1	1	0	0	2	2	6,0	1,0	0,0	1,0	2,5	4,5
REDE	1	1	1	0	2	2	7,0	1,0	1,0	0,0	1,5	3,5
SIJO	1	0,5	1	0	3	2	7,5	1,0	1,0	3,0	4,0	9,0
SIPE	0,5	1	0	0	1	0	2,5	0,0	0,0	0,0	0,0	0,0
SIVO	1	1	0	0,5	3	3	8,5	0,0	1,0	2,5	4,0	7,5
SUNO	1	0,5	0	0	3	2	6,5	1,0	1,0	4,0	4,0	10,0
Durchschnitt	0,9	0,7	0,3	0,2	2,1	2,2	6,4	0,7	0,6	2,3	2,8	6,5
Prozentual	89	74	32	16	70	72	64	71	63	58	70	65
Pi	0,89	0,74	0,32	0,16	0,70	0,72		0,71	0,63	0,58	0,70	

3.1	3.2	3.3	3.4a	3.4b	3.5	Summe	4.1	4.2	4.3	4.4	4.5	Summe	5.1	5.2	5.3	5.4	5.5	Summe
1	1	2	2	2	2	10,0	1	1	3	3	2	10,0	1	1	1	3	4	10,0
0,0	0,0	0,0	2,0	2,0	1,5	5,5	1,0	0,0	1,0	0,0	0,0	2,0	0,5	0,0	0,0	1,0	2,5	4,0
1,0	0,5	0,0	2,0	1,5	1,5	6,5	0,0	0,5	2,5	2,5	1,0	6,5	0,5	1,0	0,0	3,0	3,0	7,5
1,0	0,5	2,0	2,0	2,0	1,5	9,0	1,0	0,0	1,0	1,5	1,0	4,5	0,5	0,0	1,0	1,5	2,0	5,0
1,0	0,0	0,0	0,5	1,0	0,5	3,0	0,5	0,0	0,0	0,0	0,0	0,5	1,0	1,0	0,0	1,0	2,5	5,5
1,0	0,5	2,0	0,0	2,0	1,0	6,5	1,0	0,0	3,0	0,0	1,0	5,0	1,0	1,0	1,0	3,0	3,0	9,0
0,0	0,0	0,0	2,0	2,0	1,5	5,5	0,5	0,5	2,0	3,0	1,0	7,0	0,5	1,0	1,0	2,0	4,0	8,5
1,0	0,5	0,0	0,0	1,5	0,0	3,0	0,5	0,0	1,0	0,0	1,5	3,0	0,0	1,0	0,0	1,0	1,0	3,0
0,0	0,0	0,0	2,0	1,5	1,5	5,0	0,5	0,0	1,0	1,0	1,0	3,5	0,0	0,0	0,0	1,0	2,5	3,5
1,0	0,0	0,0	2,0	2,0	2,0	7,0	0,5	0,0	2,0	1,5	1,5	5,5	1,0	1,0	0,0	1,5	3,5	7,0
0,0	0,5	2,0	0,0	1,0	1,5	5,0	0,0	0,0	1,0	0,0	1,0	2,0	0,0	1,0	0,0	1,0	3,0	5,0
1,0	0,5	2,0	2,0	2,0	0,0	7,5	0,0	0,0	2,0	0,0	1,0	3,0	0,5	0,0	0,0	0,0	0,0	0,5
0,5	0,5	0,0	0,5	2,0	0,5	4,0	0,0	0,0	0,0	0,0	0,5	0,5	0,0	1,0	0,0	0,0	1,0	2,0
1,0	0,5	0,0	2,0	2,0	2,0	7,5	0,5	0,0	1,0	0,0	1,0	2,5	0,5	1,0	1,0	3,0	4,0	9,5
0,0	0,0	0,0	0,0	0,0	1,0	1,0	0,5	0,0	0,0	2,5	1,0	4,0	0,0	0,0	0,0	1,5	2,5	4,0
0,0	0,5	0,0	0,5	1,5	1,0	3,5	0,0	0,0	0,0	0,0	0,0	0,0	1,0	1,0	0,0	0,0	1,5	3,5
1,0	0,5	2,0	2,0	2,0	2,0	9,5	0,5	0,0	3,0	3,0	1,0	7,5	1,0	1,0	0,0	1,0	4,0	7,0
1,0	0,0	0,0	0,0	1,0	0,0	2,0	0,5	0,0	0,0	0,0	0,5	1,0	0,5	0,0	1,0	0,0	0,0	1,5
1,0	0,0	2,0	2,0	1,5	0,5	7,0	1,0	0,5	2,0	0,0	1,0	4,5	0,5	1,0	0,0	1,5	3,5	6,5
1,0	0,5	2,0	2,0	2,0	2,0	9,5	0,5	0,0	3,0	1,5	1,0	6,0	0,0	1,0	0,0	3,0	4,0	8,0
0,7	0,3	0,7	1,2	1,6	1,1	5,7	0,5	0,1	1,3	0,9	0,8	3,6	0,5	0,7	0,3	1,4	2,5	5,3
66	29	37	62	80	57	57	47	8	45	29	42	36	47	68	26	46	63	53
0,66	0,29	0,37	0,62	0,80	0,57		0,47	0,08	0,45	0,29	0,42		0,47	0,68	0,26	0,46	0,63	

67.1	67.2	67.3	67.4	67.5	67.6	67.7	67.8	67.9	67.10	67.11	Summe
1	1	1	1	1	1	2	3	2	4,5	2,5	20,0
1,0	0,0	0,5	0,0	1,0	0,5	1,5	2,0	2,0	2,5	0,5	11,5
0,0	0,5	0,5	1,0	1,0	1,0	2,0	3,0	1,5	4,0	1,5	16,0
1,0	0,5	0,5	1,0	1,0	1,0	2,0	3,0	2,0	3,5	1,5	17,0
0,0	0,5	0,0	0,0	0,0	1,0	1,5	1,0	1,5	0,0	2,0	7,5
1,0	1,0	1,0	0,0	1,0	1,0	2,0	3,0	2,0	3,5	0,0	15,5
0,0	0,5	1,0	1,0	1,0	0,5	2,0	3,0	2,0	3,0	1,0	15,0
1,0	0,0	0,5	1,0	1,0	0,0	2,0	3,0	2,0	3,5	1,0	15,0
1,0	0,0	0,5	0,0	1,0	1,0	1,5	2,0	2,0	4,0	1,5	14,5
1,0	1,0	0,5	1,0	1,0	0,0	2,0	3,0	2,0	4,5	1,5	17,5
0,0	0,5	1,0	1,0	1,0	0,0	2,0	1,5	2,0	3,0	1,0	13,0
1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0
0,0	0,0	0,5	0,0	1,0	1,0	2,0	2,5	1,5	4,0	0,5	13,0
1,0	1,0	1,0	1,0	1,0	1,0	2,0	3,0	2,0	4,5	2,5	20,0
1,0	0,5	1,0	1,0	1,0	0,0	1,5	1,0	0,5	3,5	0,0	11,0
0,0	0,0	0,5	0,0	1,0	1,0	2,0	1,5	1,0	3,0	0,0	10,0
1,0	0,5	1,0	1,0	1,0	1,0	2,0	3,0	2,0	3,5	1,5	17,5
1,0	0,0	0,0	0,0	1,0	1,0	1,5	0,5	1,0	1,0	1,5	8,5
1,0	0,5	0,5	0,0	1,0	0,5	1,5	2,0	2,0	3,5	1,5	14,0
1,0	0,5	0,5	1,0	1,0	1,0	2,0	3,0	2,0	4,0	1,5	17,5
0,7	0,4	0,6	0,5	0,9	0,7	1,7	2,2	1,6	3,1	1,1	13,4
68	39	58	53	89	66	87	72	82	68	43	67

8.1	8.2	8.3a	8.3b	Summe
1	1	4	4	10,0
0,0	1,0	3,0	0,5	4,5
1,0	1,0	3,5	3,0	8,5
0,0	1,0	4,0	3,5	8,5
0,0	0,0	1,0	1,0	2,0
0,0	0,0	1,0	4,0	5,0
0,0	1,0	4,0	4,0	9,0
1,0	1,0	3,5	2,0	7,5
0,0	1,0	3,5	2,5	7,0
0,0	1,0	3,0	4,0	8,0
1,0	1,0	0,0	4,0	6,0
0,0	1,0	0,0	0,0	1,0
0,0	0,0	3,5	3,0	6,5
0,0	1,0	3,0	4,0	8,0
0,0	1,0	1,0	0,0	2,0
0,0	0,0	0,5	0,0	0,5
1,0	1,0	2,5	3,0	7,5
0,0	0,0	1,0	0,5	1,5
0,0	1,0	4,0	3,5	8,5
0,0	1,0	3,5	4,0	8,5
0,2	0,7	2,4	2,4	5,8
21	74	60	61	58

9.1	9.2	9.3a	9.3b	Summe
1	1	3	5	10,0
0,0	1,0	0,0	0,0	1,0
0,0	0,0	1,5	1,0	2,5
0,0	1,0	0,0	0,0	1,0
0,0	0,0	1,0	0,0	1,0
0,0	1,0	1,5	3,0	5,5
1,0	1,0	1,5	0,0	3,5
1,0	0,0	0,0	0,0	1,0
1,0	0,0	0,0	0,0	1,0
1,0	1,0	2,5	0,0	4,5
0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0
1,0	1,0	2,5	4,5	9,0
1,0	0,0	0,0	0,0	1,0
0,0	0,0	0,0	0,0	0,0
1,0	1,0	2,5	4,0	8,5
0,0	0,0	0,0	0,0	0,0
1,0	0,0	0,0	0,0	1,0
1,0	1,0	1,0	4,0	7,0
0,5	0,4	0,7	0,9	2,5
47	42	25	17	25

0,68	0,39	0,58	0,53	0,89	0,66	0,87	0,72	0,82	0,68	0,43	0,21	0,74	0,60	0,61	0,47	0,42	0,25	0,17
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

10.1	10.2	Summe
2	8	10,0
1,0	6,0	7,0
1,0	4,0	5,0
2,0	2,0	4,0
2,0	2,0	4,0
1,0	4,0	5,0
2,0	3,0	5,0
2,0	2,0	4,0
1,0	6,0	7,0
0,0	3,0	3,0
2,0	6,0	8,0
2,0	3,0	5,0
1,0	2,0	3,0
1,0	4,0	5,0
1,0	2,0	3,0
2,0	5,0	7,0
1,0	6,0	7,0
2,0	3,0	5,0
1,0	6,0	7,0
2,0	5,0	7,0
1,4	3,9	5,3
71	49	53

11	Summe
10	10,0
1,0	1,0
8,0	8,0
8,0	8,0
1,0	1,0
7,0	7,0
8,0	8,0
9,0	9,0
7,5	7,5
4,0	4,0
7,0	7,0
0,0	0,0
2,0	2,0
7,5	7,5
2,5	2,5
2,0	2,0
8,5	8,5
4,5	4,5
9,5	9,5
4,0	4,0
5,3	5,3
53	53

Summe	Prozentual
110	100
51,5	46,8
78,0	70,9
70,5	64,1
35,5	32,3
75,5	68,6
78,0	70,9
56,0	50,9
65,0	59,1
70,0	63,6
54,5	49,5
25,0	22,7
39,5	35,9
86,0	78,2
39,0	35,5
37,0	33,6
89,5	81,4
26,5	24,1
74,0	67,3
84,0	76,4
59,7	54,3
54	54

OG71	max. Punkte
ANFR	
ANPE	
ARST	
CLUL	
COVO	
DAHA	
EWAD	
JAKR	
JETH	
LIFR	
LINO	
MARE	
NUYU	
PERA	
REDE	
SIJO	
SIPE	
SIVO	
SUNO	
Durchschnitt	
Prozentual	

0,71	0,49	0,53	Durchschnitt	0,54	Pi
------	------	------	--------------	------	----

OG71 (OOP-First): Aufgabe 4 noch einmal aufgeschlüsselt nach den Themen Arrays bzw. Strings

OG71	CODE	Arrays (max 7)	Strings (max 3)	Summe: Aufgabe 4 (max 10)
1	ANDA			
2	ANFR	1,0	1,0	2,0
3	ANPE	5,5	1,0	6,5
4	ARST	2,5	2,0	4,5
5	CLUL	0,0	0,5	0,5
6	COVO	3,0	2,0	5,0
7	DAHA	5,5	1,5	7,0
8	EWAD	1,0	2,0	3,0
9	JAKR	2,0	1,5	3,5
10	JETH	3,5	2,0	5,5
11	LIFR	1,0	1,0	2,0
12	LINO	2,0	1,0	3,0
13	MARE	0,0	0,5	0,5
14	NAGO			
15	NUYU	1,0	1,5	2,5
16	PERA	2,5	1,5	4,0
17	PETH			
18	REDE	0,0	0,0	0,0
19	SIJO	6,0	1,5	7,5
20	SIPE	0,0	1,0	1,0
21	SIVO	2,5	2,0	4,5
22	SUNO	4,5	1,5	6,0
<b>Durchschnitt</b>		<b>2,3</b>	<b>1,3</b>	<b>3,6</b>
<b>Prozentual</b>		<b>33</b>	<b>44</b>	<b>36</b>

OG72 (OOP-Later): Gesamtergebnis

OG72	Code	1	2	3	4	5	67	8	9	10	11	Summe	S: 1-8
1	ANHA	6,0	7,5	5,0	2,5	5,0	10,5	4,5	1,0	7,0	0,5	49,5	41,0
2	ANST	7,0	9,0	6,0	6,5	3,0	10,0	4,0	0,0	7,0	3,0	55,5	45,5
3	BAJÖ	7,0	4,0	4,5	3,0	2,0	15,0	6,0	0,0	7,0	8,0	56,5	41,5
4	BERE	7,0	6,0	6,5	6,5	2,5	9,5	0,5	1,0	7,0	2,5	49,0	38,5
5	BETH	3,5	6,0	5,0	1,0	2,0	9,5	8,0	0,0	5,0	9,0	49,0	35,0
6	BIFR	9,5	9,5	6,0	5,5	6,5	9,5	1,0	1,0	3,0	10,0	61,5	47,5
7	BIHO	9,0	10,0	5,5	5,5	6,0	16,5	10,0	1,5	6,0	8,5	78,5	62,5
8	BITH	7,5	10,0	4,0	4,0	5,5	15,0	10,0	1,5	4,0	8,5	70,0	56,0
9	BIUW	7,5	6,0	6,5	1,5	6,5	14,0	9,0	0,0	2,0	9,0	62,0	51,0
10	CARU	8,5	8,0	4,5	4,5	5,0	14,0	2,0	0,0	7,0	2,0	55,5	46,5
11	CHGE												
12	CHMI	5,0	2,5	3,0	3,5	1,0	9,5	4,0	0,0	1,0	4,0	33,5	28,5
13	DAUW	9,5	8,0	7,5	1,5	3,0	18,5	9,0	1,0	1,0	5,0	64,0	57,0
14	INWO	9,0	9,5	7,5	6,0	8,0	17,5	9,5	1,0	7,0	9,5	84,5	67,0
15	JUKL	5,5	3,5	2,0	0,0	2,0	6,5	2,0	0,0	4,0	6,0	31,5	21,5
16	LASA	10,0	10,0	7,5	8,0	10,0	15,5	8,0	10,0	5,0	8,5	92,5	69,0
17	MAAC	7,0	5,5	4,5	2,0	4,0	14,0	7,5	1,0	6,0	9,0	60,5	44,5
18	REYV												
19	SADI	7,5	9,5	5,0	7,0	10,0	18,0	7,5	3,0	7,0	9,5	84,0	64,5
20	SILO	8,0	9,5	6,0	4,0	4,5	10,5	4,5	8,0	7,0	5,0	67,0	47,0
21	TAWA	3,0	7,0	6,0	0,0	1,5	14,0	7,0	0,0	4,0	2,0	44,5	38,5
22	VEVA												
<b>Durchschnitt</b>		<b>7,2</b>	<b>7,4</b>	<b>5,4</b>	<b>3,8</b>	<b>4,6</b>	<b>13,0</b>	<b>6,0</b>	<b>1,6</b>	<b>5,1</b>	<b>6,3</b>	<b>60,5</b>	<b>47,5</b>
<b>Prozentual</b>		<b>72</b>	<b>74</b>	<b>54</b>	<b>38</b>	<b>46</b>	<b>65</b>	<b>60</b>	<b>16</b>	<b>51</b>	<b>63</b>	<b>55</b>	<b>59</b>

Die Summe 60,5 bzw. die Teilsumme (Aufgaben 1 bis 8) 47,5 wurde von Excel aus den einzelnen Teilergebnissen mit vielen Nachkommastellen berechnet. Summiert man die gerundeten Zahlen kommt man auf 60,4 bzw. 47,4.

OG72 (OOP-Later): Teilergebnisse inkl. Itemschwierigkeiten

OG72	1.1	1.2	1.3	1.4	1.5	1.6	Summe	2.1	2.2	2.3a	2.3b	Summe
max. Punkte	1,0	1,0	1,0	1,0	3,0	3,0	10,0	1	1	4	4	10,0
ANHA	1	0	1	0,5	1	2,5	6,0	1,0	1,0	2,0	3,5	7,5
ANST	1	0,5	1	0,5	1,5	2,5	7,0	1,0	1,0	4,0	3,0	9,0
BAJÖ	1	0,5	0	0,5	2	3	7,0	0,0	0,0	0,0	4,0	4,0
BERE	1	1	1	0	3	1	7,0	1,0	1,0	0,0	4,0	6,0
BETH	1	1	0	0,0	1,5	0,0	3,5	0,0	0,0	3,0	3,0	6,0
BIFR	1	1	1	0,5	3	3	9,5	1,0	1,0	4,0	3,5	9,5
BIHO	1	1	1	0,5	3	2,5	9,0	1,0	1,0	4,0	4,0	10,0
BITH	1	1	1	0,5	3	1	7,5	1,0	1,0	4,0	4,0	10,0
BIUW	1	1	1	0,5	1,5	2,5	7,5	0,0	0,0	2,5	3,5	6,0
CARU	1	1	0	0,5	3	3	8,5	1,0	0,0	3,0	4,0	8,0
CHMI	1	1	0	1	2	0	5,0	0,0	0,0	2,5	0,0	2,5
DAUW	1	1	1	0,5	3	3	9,5	1,0	0,0	3,0	4,0	8,0
INWO	1	0,5	1	0,5	3	3	9,0	1,0	1,0	4,0	3,5	9,5
JUKL	0,5	0,5	0	0	1,5	3	5,5	0,0	0,0	0,0	3,5	3,5
LASA	1	1	1	1	3	3	10,0	1,0	1,0	4,0	4,0	10,0
MAAC	0,5	0	1	0,5	2	3	7,0	1,0	0,0	0,5	4,0	5,5
SADI	1	1	0	0,5	2	3	7,5	1,0	1,0	4,0	3,5	9,5
SILO	1	0,5	1	0,5	2	3	8,0	1,0	1,0	4,0	3,5	9,5
TAWA	0,5	1	1	0,5	0	0	3,0	0,0	0,0	3,0	4,0	7,0
Durchschnitt	0,9	0,8	0,7	0,5	2,2	2,2	7,2	0,7	0,5	2,7	3,5	7,4
Prozentual	92	76	68	47	72	74	72	68	53	68	88	74

Pi	0,92	0,76	0,68	0,47	0,72	0,74	0,68	0,53	0,68	0,88
----	------	------	------	------	------	------	------	------	------	------

3.1	3.2	3.3	3.4a	3.4b	3.5	Summe	4.1	4.2	4.3	4.4	4.5	Summe	5.1	5.2	5.3	5.4	5.5	Summe
1	1	2	2	2	2	10,0	1	1	3	3	2	10,0	1	1	1	3	4	10,0
1,0	0,5	0,0	0,5	1,5	1,5	5,0	0,5	0,0	2,0	0,0	0,0	2,5	1,0	1,0	0,0	2,0	1,0	5,0
1,0	0,0	2,0	0,0	1,5	1,5	6,0	0,5	0,0	2,0	2,5	1,5	6,5	0,5	1,0	1,0	0,5	0,0	3,0
1,0	0,0	0,0	0,0	2,0	1,5	4,5	0,5	0,0	1,0	0,0	1,5	3,0	0,5	0,0	0,0	1,0	0,5	2,0
1,0	1,0	0,0	2,0	1,5	1,0	6,5	0,0	0,5	3,0	1,0	2,0	6,5	0,5	1,0	1,0	0,0	0,0	2,5
0,0	1,0	0,0	2,0	0,5	1,5	5,0	0,0	0,0	0,0	0,0	1,0	1,0	1,0	1,0	0,0	0,0	0,0	2,0
1,0	0,5	0,0	0,5	2,0	2,0	6,0	0,0	0,0	2,0	2,0	1,5	5,5	1,0	0,0	1,0	1,0	3,5	6,5
0,0	1,0	0,0	2,0	1,5	1,0	5,5	0,5	0,0	1,5	2,0	1,5	5,5	1,0	0,0	0,0	1,0	4,0	6,0
0,0	1,0	0,0	1,0	1,5	0,5	4,0	0,5	0,0	1,0	1,5	1,0	4,0	1,0	0,0	0,0	1,0	3,5	5,5
1,0	0,0	0,0	2,0	2,0	1,5	6,5	0,0	0,0	0,0	0,0	1,5	1,5	1,0	1,0	0,0	1,0	3,5	6,5
1,0	1,0	0,0	0,0	1,5	1,0	4,5	1,0	0,5	1,0	1,0	1,0	4,5	1,0	0,0	0,0	1,5	2,5	5,0
1,0	0,0	0,0	2,0	0,0	0,0	3,0	0,5	0,0	2,0	0,0	1,0	3,5	0,0	1,0	0,0	0,0	0,0	1,0
0,0	0,0	2,0	2,0	1,5	2,0	7,5	0,0	0,5	1,0	0,0	0,0	1,5	0,5	1,0	0,0	1,5	0,0	3,0
1,0	1,0	2,0	0,5	1,5	1,5	7,5	0,5	0,5	3,0	1,0	1,0	6,0	1,0	1,0	1,0	2,0	3,0	8,0
0,0	0,0	0,0	0,0	2,0	0,0	2,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	1,0	0,0	0,0	0,0	2,0
1,0	1,0	0,0	2,0	2,0	1,5	7,5	1,0	0,0	3,0	2,5	1,5	8,0	1,0	1,0	1,0	3,0	4,0	10,0
1,0	0,0	0,0	0,0	2,0	1,5	4,5	0,0	0,0	0,0	1,0	1,0	2,0	0,5	0,0	0,0	1,0	2,5	4,0
1,0	0,5	0,0	0,0	1,5	2,0	5,0	0,5	1,0	1,0	3,0	1,5	7,0	1,0	1,0	1,0	3,0	4,0	10,0
0,0	0,0	2,0	0,5	2,0	1,5	6,0	0,5	0,0	1,0	1,0	1,5	4,0	0,0	1,0	0,0	2,0	1,5	4,5
1,0	1,0	0,0	2,0	2,0	0,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,5	0,0	1,0	0,0	0,0	1,5
0,7	0,5	0,4	1,0	1,6	1,2	5,4	0,3	0,2	1,3	1,0	1,1	3,8	0,7	0,6	0,4	1,1	1,8	4,6
68	50	21	50	79	61	54	34	16	43	32	53	38	74	63	37	38	44	46

0,68	0,50	0,21	0,50	0,79	0,61	0,34	0,16	0,43	0,32	0,53	0,74	0,63	0,37	0,38	0,44
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

67.1	67.2	67.3	67.4	67.5	67.6	67.7	67.8	67.9	67.10	67.11	Summe
1	1	1	1	1	1	2	3	2	4,5	2,5	20,0
1,0	0,5	0,5	1,0	0,0	1,0	1,5	2,0	1,0	0,0	2,0	10,5
0,0	0,0	1,0	0,0	1,0	0,5	2,0	2,5	1,5	0,0	1,5	10,0
0,0	0,5	0,5	1,0	0,5	1,0	2,0	2,0	2,0	3,0	2,5	15,0
0,0	0,0	1,0	0,0	1,0	1,0	0,0	1,5	1,0	2,5	1,5	9,5
0,0	0,5	0,0	0,0	0,0	0,0	0,0	3,0	1,0	3,0	2,0	9,5
0,0	0,0	0,5	0,0	1,0	1,0	2,0	3,0	2,0	0,0	0,0	9,5
0,0	0,5	1,0	1,0	1,0	0,5	2,0	3,0	2,0	4,0	1,5	16,5
0,0	0,5	0,5	1,0	1,0	1,0	1,5	2,5	1,5	4,0	1,5	15,0
0,0	0,5	1,0	1,0	0,5	1,0	2,0	2,5	2,0	2,0	1,5	14,0
1,0	0,5	1,0	0,0	1,0	1,0	1,5	2,5	1,5	3,5	0,5	14,0
0,0	0,0	0,5	0,0	1,0	0,5	2,0	1,0	0,5	2,5	1,5	9,5
1,0	1,0	1,0	1,0	0,5	1,0	2,0	3,0	2,0	3,5	2,5	18,5
0,0	1,0	1,0	0,0	1,0	1,0	2,0	2,5	2,0	4,5	2,5	17,5
0,0	0,5	0,5	1,0	1,0	1,0	0,0	0,0	0,0	1,0	1,5	6,5
0,0	0,0	1,0	0,0	1,0	1,0	2,0	3,0	2,0	3,5	2,0	15,5
0,0	0,0	0,5	0,0	1,0	0,0	2,0	3,0	2,0	4,0	1,5	14,0
0,0	1,0	1,0	1,0	1,0	1,0	1,5	3,0	2,0	4,5	2,0	18,0
1,0	0,5	0,5	1,0	0,0	1,0	1,5	2,0	1,0	0,0	2,0	10,5
0,0	0,0	0,5	1,0	1,0	1,0	1,5	2,0	1,5	4,0	1,5	14,0
0,2	0,4	0,7	0,5	0,8	0,8	1,5	2,3	1,5	2,6	1,7	13,0
21	39	71	53	76	82	76	77	75	58	66	65

8.1	8.2	8.3a	8.3b	Summe
1	1	4	4	10,0
1,0	1,0	2,5	0,0	4,5
1,0	1,0	2,0	0,0	4,0
1,0	1,0	1,0	3,0	6,0
0,0	0,0	0,5	0,0	0,5
1,0	1,0	2,5	3,5	8,0
0,0	1,0	0,0	0,0	1,0
1,0	1,0	4,0	4,0	10,0
1,0	1,0	4,0	4,0	10,0
1,0	1,0	4,0	3,0	9,0
1,0	1,0	0,0	0,0	2,0
0,0	1,0	3,0	0,0	4,0
1,0	1,0	3,5	3,5	9,0
1,0	1,0	4,0	3,5	9,5
0,0	0,0	2,0	0,0	2,0
0,0	1,0	3,0	4,0	8,0
0,0	1,0	3,5	3,0	7,5
0,0	1,0	3,0	3,5	7,5
1,0	1,0	2,5	0,0	4,5
1,0	0,0	3,0	3,0	7,0
0,6	0,8	2,5	2,0	6,0
63	84	63	50	60

9.1	9.2	9.3a	9.3b	Summe
1	1	3	5	10,0
0,0	1,0	0,0	0,0	1,0
0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0
0,0	1,0	0,0	0,0	1,0
0,0	0,0	0,0	0,0	0,0
0,0	1,0	0,0	0,0	1,0
0,5	1,0	0,0	0,0	1,5
0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0
0,5	1,0	0,0	0,0	1,5
0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0
1,0	0,0	0,0	0,0	1,0
0,0	0,0	0,0	1,0	1,0
0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0
1,0	1,0	3,0	5,0	10,0
1,0	0,0	0,0	0,0	1,0
1,0	0,0	1,0	1,0	3,0
0,0	1,0	3,0	4,0	8,0
0,0	0,0	0,0	0,0	0,0
0,3	0,4	0,4	0,6	1,6
26	37	12	12	16

0,21 0,39 0,71 0,53 0,76 0,82 0,76 0,77 0,75 0,58 0,66 0,63 0,84 0,63 0,50 0,26 0,37 0,12 0,12

10.1	10.2	Summe
2	8	10,0
1,0	6,0	7,0
1,0	6,0	7,0
1,0	6,0	7,0
1,0	6,0	7,0
2,0	3,0	5,0
1,0	2,0	3,0
1,0	5,0	6,0
1,0	3,0	4,0
1,0	1,0	2,0
2,0	5,0	7,0
1,0	0,0	1,0
1,0	0,0	1,0
1,0	6,0	7,0
2,0	2,0	4,0
2,0	3,0	5,0
1,0	5,0	6,0
1,0	6,0	7,0
1,0	6,0	7,0
2,0	2,0	4,0
1,3	3,8	5,1
63	48	51

11	Summe
10	10,0
0,5	0,5
3,0	3,0
8,0	8,0
2,5	2,5
9,0	9,0
10,0	10,0
8,5	8,5
8,5	8,5
9,0	9,0
2,0	2,0
4,0	4,0
5,0	5,0
9,5	9,5
6,0	6,0
8,5	8,5
9,0	9,0
9,5	9,5
5,0	5,0
2,0	2,0
6,3	6,3
63	63

Summe	Prozentual
110	100
49,5	45,0
55,5	50,5
56,5	51,4
49,0	44,5
49,0	44,5
61,5	55,9
78,5	71,4
70,0	63,6
62,0	56,4
55,5	50,5
33,5	30,5
64,0	58,2
84,5	76,8
31,5	28,6
92,5	84,1
60,5	55,0
84,0	76,4
67,0	60,9
44,5	40,5
60,5	55,0
55	55

OG72
max. Punkte
ANHA
ANST
BAJÖ
BERE
BETH
BIFR
BIHO
BITH
BIUW
CARU
CHMI
DAUW
INWO
JUKL
LASA
MAAC
SADI
SILO
TAWA
Durchschnitt
Prozentual

0,63 0,48 0,63 Durchschnitt 0,56 Pi




OG72 (OOP-Later): Aufgabe 4 noch einmal aufgeschlüsselt nach den Themen Arrays bzw. Strings

OG72	Code	Arrays (max 7)	Strings (max 3)	Summe: Aufgabe 4 (max 10)
1	ANHA	2,0	0,5	2,5
2	ANST	4,5	2,0	6,5
3	BAJÖ	1,0	2,0	3,0
4	BERE	4,5	2,0	6,5
5	BETH	0,0	1,0	1,0
6	BIFR	4,0	1,5	5,5
7	BIHO	3,5	2,0	5,5
8	BITH	2,5	1,5	4,0
9	BIUW	0,0	1,5	1,5
10	CARU	2,5	2,0	4,5
11	CHGE			
12	CHMI	2,0	1,5	3,5
13	DAUW	1,5	0,0	1,5
14	INWO	4,5	1,5	6,0
15	JUKL	0,0	0,0	0,0
16	LASA	5,5	2,5	8,0
17	MAAC	1,0	1,0	2,0
18	REYV			
19	SADI	5,0	2,0	7,0
20	SILO	2,0	2,0	4,0
21	TAWA	0,0	0,0	0,0
22	VEVA			
<b>Durchschnitt</b>		<b>2,4</b>	<b>1,4</b>	<b>3,8</b>
<b>Prozentual</b>		<b>35</b>	<b>46</b>	<b>38</b>

## Q) Vortest (Pre-Test) 2008: erste Gruppe

Die Lösung findet sich beim Nachhaltigkeitstest (Anhang O).

	<b>Eingangsbefragung 2008</b>	Informatik Beginn der Einführungsphase
Anonyme Umfrage (Gruppe1)	Datum:	Blatt Nr.: 1 / 1

In der folgenden Eingangsbefragung werden Fragen zu 4 Themengebieten gestellt.

Zu jedem dieser Themengebiete gibt es in der Regel ein Blatt mit einer Mischung aus Ankreuzaufgaben und Fragen, die einer eigenen Lösung bedürfen. In jedem Themengebiet können 10 Punkte erreicht werden.

**Hinweis: Bei den Multiple-Choice-Fragen sind evt. beliebig viele Antworten (aber mindestens eine) richtig!** Fehlerhafte Kreuze führen zu Punktabzug.

Nr.	Themengebiet	Erreichbare Punkte	Erreichte Punkte
1	Datentypen und Steuerstruktur Sequenz	10	
3	Steuerstruktur: Iteration	10	
6 / 7	Einführung in die OOP / Klasse und Objekt	20	
Summe		40	

*Vielen Dank für Ihre Mitarbeit!*

Ehler 2008
[ehler@oszimt.de](mailto:ehler@oszimt.de)
[www.oszimt.de](http://www.oszimt.de)

**1. Datentypen und Steuerstruktur Sequenz**

1.	Durch den <b>Datentyp</b> einer Variablen wird festgelegt, welche(n) ...	<input type="radio"/> Werte die Variable annehmen kann <input type="radio"/> Namen die Variable hat <input type="radio"/> Adresse die Variable im Speicher hat <input type="radio"/> Wert die Variable hat <input type="radio"/> Anzahl von Bits die Variable im Speicher belegt	<b>1P</b>
2.	Gegeben sind die beiden Quelltextzeilen: <pre>final double PI; PI = 3.1415;</pre> Welche Aussage(n) trifft/treffen zu?	<input type="radio"/> Compilerfehler in Zeile 1 <input type="radio"/> Compilerfehler in Zeile 2 <input type="radio"/> PI ist eine Konstante <input type="radio"/> PI muss klein geschrieben werden	<b>1P</b>
3.	Welche Werte kann eine Variable vom <b>Datentyp short</b> annehmen?	<input type="radio"/> 0 ... 65.535 <input type="radio"/> -32.768 bis 32.767 <input type="radio"/> -32.767 bis 32.768 <input type="radio"/> keine Aussage trifft zu	<b>1P</b>
4.	Gegeben sind die beiden Quelltextzeilen: <pre>byte a = 7; int b = a;</pre> Welche Aussage(n) trifft/treffen zu?	<input type="radio"/> Es gibt einen Compilerfehler <input type="radio"/> Es gibt keinen Compilerfehler <input type="radio"/> b belegt 16 Bit <input type="radio"/> b belegt 32 Bit	<b>1P</b>

5. Gegeben ist die folgende lückenhafte **Variablendeklaration**: **3P**

```
char    a = _____;
_____ b = 7.0f;
_____ c = (5 == 7);
```

**Ergänzen** Sie den **Quelltext** in den 3 Lücken!

6. Gegeben sind die drei Quelltextzeilen:

```
byte a = 7;
int b = 8;
double c;
c = a * b;
```

**Zeichnen** Sie das dazugehörige **Struktogramm inkl. Einträge!** **3P**

**3. Steuerstruktur: Iteration**

1.	Gegeben ist die folgende Quelltextzeile: <code>for (int i = 1; i &lt; 7; i++) { }</code> Diese <b>Schleife</b> wird im <b>Struktogramm</b> mit Hilfe welchen Strukturelements nach DIN 66261 dargestellt?	<input type="radio"/> kopfgesteuerte Schleife <input type="radio"/> fußgesteuerte Schleife <input type="radio"/> Endlosschleife <input type="radio"/> keiner der genannten Schleifen <p align="right"><b>1P</b></p>
2.	Welche Antwort(en) enthält/enhalten <b>keine</b> Endlosschleife?	<input type="radio"/> <code>for ( ; ; ) { }</code> <input type="radio"/> <code>do { } while (false);</code> <input type="radio"/> <code>while (!true) { }</code> <input type="radio"/> <code>while (9 == 9) { }</code> <p align="right"><b>1P</b></p>

3. Gegeben ist der folgende Quelltextausschnitt:

```
int a = 4 * 8;
int b = a / 2;
while (b > 2) {
    a = a - 3;
    b = b - 2;
}
```

Welcher Wert steht am Ende des Algorithmus in der Variable **a**?

a

**2P**

4. Gegeben ist der folgende Quelltextausschnitt:

```
i = (4 * 3) * 3;
do {
    i++;
    i = i - 4;
} while (i > 6);
```

Wie oft wird der Schleifenkörper durchlaufen?  
Zeichnen Sie das **Struktogramm**!

**Hinweis:** Einträge in das Struktogramm sind **nicht** erforderlich!

Anzahl

**2P  
2P**

5. **Ergänzen** Sie den folgenden **Quelltext**, so dass als Ausgabe: „24,12,6,“ erscheint! **2P**

```
for ( _____ )
    System.out.print(z + ",");
```

**6. Einführung in die OOP / 7. Klasse und Objekt**


1.	Welche Aussage(n) über <b>Klassen</b> sind richtig?	<input type="radio"/> Eine Klasse ist in einem bestimmten Zustand <input type="radio"/> Eine Klasse ist die Summe der Objekte <input type="radio"/> Eine Klasse hat immer Attribute <input type="radio"/> Keine Aussage ist richtig <b>1P</b>
2.	Welche Aussage(n) über <b>Objekte</b> ist/sind richtig?	<input type="radio"/> Von jeder Klasse gibt es mindestens ein Objekt <input type="radio"/> Objekte können mit anderen Objekten Botschaften austauschen <input type="radio"/> Objekte befinden sich in einem Zustand <input type="radio"/> Objekte haben einen Zugriff auf sich selbst <b>1P</b>
3.	Welche Aussage(n) zu <b>Attributen</b> ist/sind richtig?	<input type="radio"/> Attribute sind vergleichbar mit Variablen <input type="radio"/> Ein Objekt deklariert Attribute <input type="radio"/> Attribute haben einen Datentyp <input type="radio"/> Attribute sind in der Regel privat <b>1P</b>
4.	<b>Methoden</b> werden im UML-Objektdiagramm ...	<input type="radio"/> angegeben <input type="radio"/> nicht angegeben <input type="radio"/> je nach Bedarf angegeben <b>1P</b>
5.	Wozu braucht man die <b>Verwaltungsmethoden</b> („Setter“ und „Getter“)?	<input type="radio"/> Weil diese immer deklariert werden müssen <input type="radio"/> Weil die Attribute in der Regel öffentlich sind <input type="radio"/> Weil die Attribute in der Regel privat sind <b>1P</b>
6.	Wozu dienen <b>Konstruktoren</b> ?	<input type="radio"/> Um eine Vererbungshierarchie zu ermöglichen <input type="radio"/> Um eine Klasse zu initialisieren <input type="radio"/> Um ein Objekt zu initialisieren <input type="radio"/> Um ein Klasse deklarieren zu können <b>1P</b>

7. Die Klasse „Auto“ hat die Attribute „leistung“ (Datentyp int) und „hersteller“ (Datentyp String).  
Wie lautet der **Methodenkopf** des **vollparametrisierten Konstruktors**?  
**Hinweis:** Nur **eine Quelltextzeile** ist gefragt! **2P**

8. Die Klasse „Bruecke“ hat u. a. das Attribut „breite“ vom Datentyp „float“.  
Wie lautet der vollständige Quelltext der „**Setter**“-**Verwaltungsmethode**? **3P**

## R) Vortest (Pre-Test) 2008: zweite Gruppe

Die Lösung findet sich beim Nachhaltigkeitstest (Anhang O).

	<b>Eingangsbefragung 2008</b>	Informatik Beginn der Einführungsphase
Anonyme Umfrage (Gruppe2)	Datum:	Blatt Nr.: 1 / 1

In der folgenden Eingangsbefragung werden Fragen zu 4 Themengebieten gestellt.

Zu jedem dieser Themengebiete gibt es in der Regel ein Blatt mit einer Mischung aus Ankreuzaufgaben und Fragen, die einer eigenen Lösung bedürfen. In jedem Themengebiet können 10 Punkte erreicht werden.

**Hinweis: Bei den Multiple-Choice-Fragen sind evt. beliebig viele Antworten (aber mindestens eine) richtig!** Fehlerhafte Kreuze führen zu Punktabzug.

Nr.	Themengebiet	Erreichbare Punkte	Erreichte Punkte
2	Steuerstruktur: Selektion	10	
4	Arrays und Strings	10	
5	Methoden	10	
8	Vererbung	10	
Summe		40	

*Vielen Dank für Ihre Mitarbeit!*

Ehler 2008
[ehleri@oszimt.de](mailto:ehleri@oszimt.de)
www.oszimt.de

**2. Steuerstruktur: Selektion**

1.	Gegeben ist der folgende Vergleich: <code>if (!isRichtig) {}</code> Dies kann man auch wie folgt schreiben:	<input type="radio"/> <code>if (isRichtig = true) {}</code> <input type="radio"/> <code>if (isRichtig = false) {}</code> <input type="radio"/> <code>if (isRichtig == true) {}</code> <input type="radio"/> <code>if (isRichtig == false) {}</code>
		<b>1P</b>
2.	<pre>int b = 0, c = 7; if (c &lt; 7)     if (c &gt;= 7) b = 1;     else      b = 2; else     if (c &gt; 7) b = 3;     else      b = 4; </pre> Was steht am Ende in der Variablen b?	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
		<b>1P</b>

3. Gegeben ist der folgende Quelltextauschnitt:

```
int a = 4; // oder 6 oder 8
switch (a) {
    case 4:
        a = 3 * a;
        break;
    case 6:
        a = a / 5;
        break;
    default:
        a = a % 3; // modulo
        break;
} // switch
if (a < 10)
    a--;
else
    a++;
System.out.println("a: " + a);
```

a	Ausgabe
4	
6	
8	

a) Welcher Wert wird ausgegeben, bei den folgenden ursprünglichen Werten für die Variable a (3 Tabelleneinträge)? **4P**

b) Zeichnen Sie das dazugehörige **Struktogramm!** **4P**  
**Hinweis:** Einträge in das Struktogramm sind **nicht** erforderlich!

**4. Arrays und Strings**

1.	Gegeben ist die folgende Variablendeklaration: <code>String s = "Test"; String t = new String("Test");</code> Welche Bedingungen liefern true zurück?	<input type="radio"/> <code>if (s == "Test") {}</code> <input type="radio"/> <code>if (t == s) {}</code> <input type="radio"/> <code>if (s.equals("Test")) {}</code> <input type="radio"/> <code>if (t.equals(s)) {}</code> <b>1P</b>
2.	int[] x = new int[25]; Welche Aussage(n) trifft/treffen nach der Ausführung der Programmzeile zu?	<input type="radio"/> x.length hat den Wert 25 <input type="radio"/> x[0] hat den Wert null <input type="radio"/> x[24] hat den Wert 0 <input type="radio"/> x[25] ist nicht definiert <b>1P</b>

3. Gegeben ist der folgende Quelltextausschnitt: **3P**

```
int[] exampleArray = {10, 20, 30, 40};
for (int i = 1; i <= exampleArray.length; i++)
    System.out.println(exampleArray[i]);
```

Wie lautet die **Bildschirmausgabe**?

4. Initialisieren Sie das folgende Feld **mit Hilfe einer Zählschleife** mit den Werten 'a', 'b' und 'c'! **3P**

```
char[] zeichenfeld = new char[3];
```

5. Gegeben ist der folgende Quelltext: **2P**

```
String s1 = new String("berlin");
s1.toUpperCase();
String s2 = null;
System.out.println(s1 + s2 + s1.length());
```

Wie lautet die Bildschirmausgabe?



**5. Methoden**

1.	Beim <b>Aufruf einer Methode</b> findet man immer ...	<input type="radio"/> einen Methodennamen <input type="radio"/> einen aktuellen Parameter <input type="radio"/> einen formalen Parameter <input type="radio"/> einen Rückgabewert <span style="float: right;"><b>1P</b></span>
2.	Welche Aussage(n) trifft/treffen für die <b>Methode berechne()</b> zu? <code>if (berechne() == 4.7) {}</code>	<input type="radio"/> Sie hat einen boolean-Rückgabewert <input type="radio"/> Sie hat einen double-Rückgabewert <input type="radio"/> Sie hat keinen Rückgabewert <span style="float: right;"><b>1P</b></span>
3.	Gegeben ist die folgende <b>Methode tueEtwas(...)</b> : <code>int a; a = tueEtwas(4.7);</code> Welche Aussage(n) ist/sind richtig?	<input type="radio"/> Die Methode hat keinen Rückgabewert <input type="radio"/> a muss vom Datentyp <code>double</code> sein <input type="radio"/> Vor dem Wert 4.7 fehlt der Datentyp <code>double</code> <input type="radio"/> keine Aussage trifft zu <span style="float: right;"><b>1P</b></span>

4. Gegeben sind die folgenden Quelltextzeilen des Anwendungsprogramms, die in der main-Methode stehen: **3P**

```
char[] zeichenfeld = new char[10];
initialisiereFeld(zeichenfeld);
anzahl = ermittleHaeufigkeit(zeichenfeld, 'c');
```

Wie lautet der **Methodenkopf** der statischen Methode „ermittleHaeufigkeit(...)“, welcher im Anwendungsprogramm „über“ der main-Methode steht?  
**Hinweis:** Nur **eine Quelltextzeile** ist gefragt!

5. In der Klasse „Formelsammlung“ soll eine Methode „berechneFlaeche“ **4P**  
implementiert werden, die als formale Parameter eine laenge (double) und breite (double) hat und die berechnete flaeche zurückliefert. Ist einer der beiden Parameter negativ, wird 0 zurück gegeben.  
**Geben Sie den vollständigen Quelltext der Methode „berechneFlaeche“ (Methodenkopf und Implementierung) an!**

**8. Vererbung**

1.	Eine <b>Unterklasse</b> hat <b>3 Attribute</b> . Wie viele Attributswerte hat ein Objekt der <b>Oberklasse</b> ?	<input type="radio"/> maximal 3 <input type="radio"/> genau 3 <input type="radio"/> mindestens 3 <input type="radio"/> unbekannt, da Oberklasse von Unterklasse unabhängig ist <b>1P</b>
2.	Welche Aussage(n) ist/sind richtig? Eine <b>abstrakte Klasse</b> ...	<input type="radio"/> kann keine Oberklasse sein <input type="radio"/> kann keine Attribute haben <input type="radio"/> kann keine Methoden haben <input type="radio"/> keine Antwort richtig <b>1P</b>

3. Gegeben ist der folgende Quelltextausschnitt:

```
public abstract class Tier {
    private int geburtsjahr;
    public abstract void gebeLaut();
} // Tier

public class Katze extends Tier {
    private int anzahlKatzenleben;
    public void gebeLaut() {
        System.out.println("Miau");
    }
} // Katze
```

a) Zeichnen Sie das **UML-Klassendiagramm** (nur mit den gegebenen Attributen und Methoden, also ohne setter/getter und Konstruktoren) **4P**

b) Schreiben Sie den **Quelltext** des **vollparametrisierten Konstruktors** der Klasse **Katze**! Hinweis: Die Oberklasse Tier hat einen parametrisierten Konstruktor. **4P**

## S) Ergebnisse des Vortests 2008


14 Schülerinnen und Schüler der Klasse OG81 erhielten den Test 1, 12 Schülerinnen und Schüler den Test 2.

OG81	1	2	3	4	5	67	8	Summe
	1	1,5	0,5			0,0		2,0
	2	1,5	0,0			3,0		4,5
	3	1,0	0,5			0,5		2,0
	4	1,5	1,5			1,0		4,0
	5	2,0	0,0			2,0		4,0
	6	1,5	0,0			1,0		2,5
	7	0,5	0,0			2,0		2,5
	8	2,0	0,0			0,5		2,5
	9	4,5	3,5			6,0		14,0
	10	2,0	4,0			2,0		8,0
	11	1,5	0,0			0,0		1,5
	12	2,0	1,0			4,0		7,0
	13	1,0	3,5			1,0		5,5
	14	2,5	0,0			0,5		3,0
	15		0,0		0,5	1,5	0,0	2,0
	16		0,0		0,0	0,0	2,0	2,0
	17		0,0		0,5	1,0	1,0	2,5
	18		0,0		0,0	2,0	1,0	3,0
	19		0,5		1,5	1,0	0,0	3,0
	20		2,0		0,0	0,0	0,0	2,0
	21		1,5		0,0	0,0	0,0	1,5
	22		0,0		0,0	0,0	0,0	0,0
	23		1,0		0,0	1,0	1,0	3,0
	24		1,0		0,0	1,0	0,0	2,0
	25		0,0		2,0	2,0	0,0	4,0
	26		5,0		4,0	4,5	0,0	13,5
<b>Durchschnitt</b>	<b>1,8</b>	<b>0,9</b>	<b>1,0</b>	<b>0,7</b>	<b>1,2</b>	<b>1,7</b>	<b>0,4</b>	<b>3,9</b>
<b>Prozentual</b>	<b>18</b>	<b>9</b>	<b>10</b>	<b>7</b>	<b>12</b>	<b>8</b>	<b>4</b>	<b>10</b>

9 Schülerinnen und Schüler der Klasse OG82 erhielten den Test 1, 15 Schülerinnen und Schüler den Test 2.

OG82		1	2	3	4	5	67	8	Summe
	1	0,5		1,0			1,5		3,0
	2	2,5		0,0			0,0		2,5
	3	1,0		0,5			2,0		3,5
	4	2,0		1,5			2,0		5,5
	5	2,0		2,0			1,5		5,5
	6	1,5		0,0			2,5		4,0
	7	1,5		0,5			1,0		3,0
	8	2,0		0,0			0,0		2,0
	9	2,0		1,0			3,0		6,0
	10		1,5		0,0	1,0		0,0	2,5
	11		1,5		1,0	0,0		1,0	3,5
	12		7,0		2,0	1,5		1,0	11,5
	13		2,0		1,0	2,0		0,0	5,0
	14		1,5		0,5	1,0		1,0	4,0
	15		0,0		0,0	1,0		1,0	2,0
	16		0,0		0,0	0,0		2,0	2,0
	17		0,0		0,0	2,0		2,0	4,0
	18		0,0		0,0	1,0		1,0	2,0
	19		1,0		0,0	1,0		2,0	4,0
	20		0,0		0,0	0,0		1,0	1,0
	21		0,0		0,0	1,0		0,0	1,0
	22		0,0		1,0	2,0		2,0	5,0
	23		3,5		1,0	3,0		1,0	8,5
	24		0,0		0,5	2,0		1,0	3,5
<b>Durchschnitt</b>		<b>1,7</b>	<b>1,2</b>	<b>0,7</b>	<b>0,5</b>	<b>1,2</b>	<b>1,5</b>	<b>1,1</b>	<b>3,9</b>
<b>Prozentual</b>		<b>17</b>	<b>12</b>	<b>7</b>	<b>5</b>	<b>12</b>	<b>8</b>	<b>11</b>	<b>10</b>

## T) Informations- und Arbeitsblätter zu den Themen 8 und 9

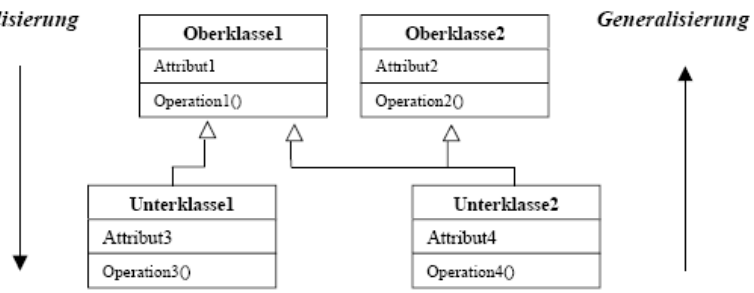
	<b>OOP mit Java</b> Skript	<b>Informatik</b>
Name:	Klasse:	Datum:
		Blatt Nr.: 19/19

### OOP4: Vererbung

Vererbung bedeutet, dass eine spezialisierte Klasse (**Unterklasse**, subclass, **abgeleitete Klasse**) über die Eigenschaften, das Verhalten und die Assoziationen einer oder mehrerer Klassen (**Oberklassen**, superclasses, **Basisklassen**) verfügen kann. Die Unterklasse enthält dabei in der Regel zusätzliche Informationen (Attribute, Operationen, Assoziationen).

*Spezialisierung*  *Generalisierung*



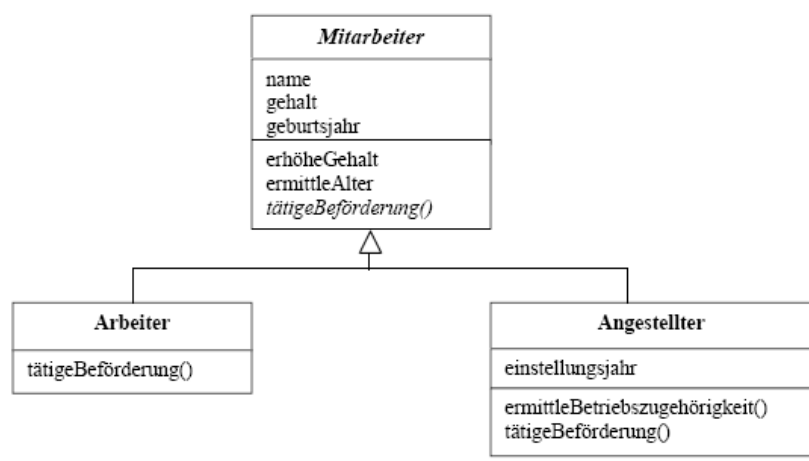
```

classDiagram
    class Oberklasse1 {
        Attribut1
        Operation1()
    }
    class Oberklasse2 {
        Attribut2
        Operation2()
    }
    class Unterklass1 {
        Attribut3
        Operation3()
    }
    class Unterklass2 {
        Attribut4
        Operation4()
    }
    Oberklasse1 <|-- Unterklass1
    Oberklasse1 <|-- Unterklass2
    Oberklasse2 <|-- Unterklass2
    
```

Die Unterklass1 kennt das Attribut1, das Attribut3, die Operation1 und die Operation3 (**Einfachvererbung**). Die Unterklass2 kennt das Attribut1, das Attribut2, das Attribut4, die Operation1, die Operation2 und die Operation4 (**Die sog. Mehrfachvererbung gibt es aber in Java nicht!**).

### Abstrakte Klasse

Wenn von einer Oberklasse **keine Objekte** erzeugt werden, dann ist sie abstrakt. Sie kann auch **abstrakte Operationen** enthalten. Diese müssen dann in den Unterklassen implementiert werden.



```

classDiagram
    class Mitarbeiter {
        name
        gehalt
        geburtsjahr
        erhöheGehalt
        ermittleAlter
        tätigeBeförderung()
    }
    class Arbeiter {
        tätigeBeförderung()
    }
    class Angestellter {
        einstellungsjahr
        ermittleBetriebszugehörigkeit()
        tätigeBeförderung()
    }
    Mitarbeiter <|-- Arbeiter
    Mitarbeiter <|-- Angestellter
    
```

[ehlert@oszimt.de](mailto:ehlert@oszimt.de) Java-OOP-Skript-Neu Oktober 2005

**Vererbung in Java**

Klassen sind vererbbar, d.h. eine Klasse kann von einer anderen Klasse deren Merkmale (Attribute und Methoden) übernehmen. Zum Beispiel könnte man die Mitarbeiter einer Firma in Angestellte, Arbeiter und Manager aufteilen, die trotzdem alle Mitarbeiter der Firma sind:

```
public class Angestellter extends Mitarbeiter {}
public class Arbeiter extends Mitarbeiter {}
public class Manager extends Mitarbeiter {}
```

Werden von der Klasse Mitarbeiter keine Objekte mehr erzeugt, ist sie eine sog. abstrakte Klasse:

```
public abstract class Mitarbeiter {}
```

**Hinzufügen von Attributen und Methoden**

Abgeleitete Klassen zeichnen sich oft durch hinzukommende Merkmale aus. Nehmen wir einmal an, dass bei Angestellten das Einstellungsjahr wichtig ist, um Dienstjubiläen bestimmen zu können:

```
public class Angestellter extends Mitarbeiter {
    private int einstellungsjahr;
    public int ermittleBetriebszugehoerigkeit(int aktuellesJahr) {
        return(aktuellesJahr - einstellungsjahr);
    } // ermittleBetriebszugehoerigkeit
} // abgeleitete Klasse Angestellter
```

Nun wird erkennbar, dass Vererbung viel Programmieraufwand erspart, da in der abgeleiteten Klasse nicht mehr die Attribute und Methoden der Basisklasse erneut definiert werden müssen.

**Überschreiben von Methoden**

Es ist auch möglich, Methoden höherer Klassen zu modifizieren bzw. zu überschreiben. Dabei werden einzelne Methoden der ursprünglichen Basisklasse in der abgeleiteten Klasse neu definiert:

<pre>// Basisklasse bzw. Oberklasse public abstract class Mitarbeiter {     private String name;     private int gehalt;     private int geburtsjahr;     public Mitarbeiter() { ... }     public Mitarbeiter(String,int,int) { ... }     public void setName(String) { ... }     public String getName() { ... }     public void setGehalt(int) { ... }     public int getGehalt() { ... }     public void erhoeheGehalt(int) { ... }     public void erniedrigeGehalt(int) { ... }     public void setGeburtsjahr(int) { ... }     public int getGeburtsjahr() { ... }     public int ermittleAlter(int) { ... }      public void taetigeBefoerderung() { ... } // Methode enthält Implementierung, kann //public abstract void taetigeBefoerderung(); // aber überschrieben werden; // Methode enthält keine Implementierung; } // Basisklasse Mitarbeiter // sie muss von Unterklassen überschrieben // werden!</pre>	<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <th align="center" style="padding: 5px;">Mitarbeiter</th> </tr> <tr> <td style="padding: 5px;">                     - name: String                      - gehalt: int                      - geburtsjahr: int                 </td> </tr> <tr> <td style="padding: 5px;">                     + erhoeheGehalt(int): void                      + erniedrigeGehalt(int): void                      + ermittleAlter(int): int                      + taetigeBefoerderung(): void bzw.                      + taetigeBefoerderung(): void                 </td> </tr> </table>	Mitarbeiter	- name: String - gehalt: int - geburtsjahr: int	+ erhoeheGehalt(int): void + erniedrigeGehalt(int): void + ermittleAlter(int): int + taetigeBefoerderung(): void bzw. + taetigeBefoerderung(): void
Mitarbeiter				
- name: String - gehalt: int - geburtsjahr: int				
+ erhoeheGehalt(int): void + erniedrigeGehalt(int): void + ermittleAlter(int): int + taetigeBefoerderung(): void bzw. + taetigeBefoerderung(): void				

Dabei ist die ursprüngliche Methode für eine Beförderung wie folgt definiert:

```
public void taetigeBefoerderung() {
    erhoeheGehalt(500);
} // taetigeBefoerderung
```

Wird bei einer abgeleiteten Klasse eine Methode nicht neu definiert, wird die Basismethode übernommen, sprich das Gehalt bei einer Beförderung um 500 EURO erhöht .

Nun kann es aber sein, dass bei einer Beförderung von Arbeitern oder Managern andere Beträge gelten. Dann muss die **Methode** neu definiert bzw. **überschrieben** werden:

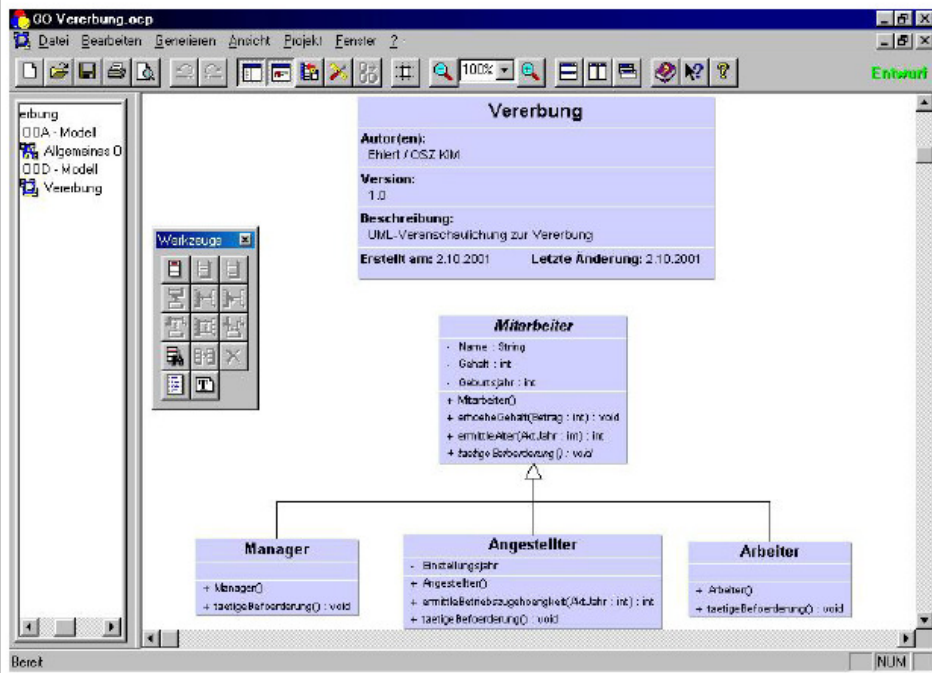
```
public class Arbeiter extends Mitarbeiter {
    public void taetigeBefoerderung() {
        erhoeheGehalt(400);
    } //taetigeBefoerderung
} // Arbeiter
```

```
public class Manager extends Mitarbeiter {
    public void taetigeBefoerderung() {
        erhoeheGehalt(1000);
    } // taetigeBefoerderung
} // Manager
```

Überschreiben alle abgeleiteten Klassen die ursprüngliche Methode, dann reicht es den Methodenkopf in der Basisklasse ohne jegliche Implementierung anzugeben. Die Methode ist dann abstrakt:

```
public abstract void taetigeBefoerderung();
```

In der UML-Darstellung werden abstrakte Klassen und Methoden kursiv geschrieben, Vererbung wird mit einem Pfeil gekennzeichnet (die Attribute sollten eigentlich klein geschrieben sein):



Name:

Klasse:

Datum:

Blatt Nr.: 22/22

### Konstruktor einer abgeleiteten Klasse

Wie wird aber eine abgeleitete Klasse initialisiert? Oft wird einfach mit dem Schlüsselwort „super“ auf den Konstruktor der Basisklasse zurückgegriffen:

```
public class Arbeiter extends Mitarbeiter {

    public Arbeiter(String name, int gehalt, int geburtsjahr) {
        super(name,gehalt,geburtsjahr);
    } // parametrisierter Konstruktor

    public void taetigeBefoerderung() {
        erhoeheGehalt(400);
    } // taetigeBefoerderung

} // Arbeiter
```

Bei der Klasse „Angestellter“ muss aber zusätzlich das Einstellungsjahr initialisiert werden:

```
public class Angestellter extends Mitarbeiter {

    // Hinzufuegen eines Attributs
    private int einstellungsjahr;

    // Eigener Konstruktor
    public Angestellter(String name,int gehalt,int geburtsjahr,int einstellungsjahr){
        super(name,gehalt,geburtsjahr);
        this.einstellungsjahr = einstellungsjahr;
    } // parametrisierter Konstruktor

    // Hinzufuegen einer Methode
    public int ermittleBetriebszugehoerigkeit(int aktuellesJahr) {
        return (aktuellesJahr - einstellungsjahr);
    } // ermittleBetriebszugehoerigkeit

    public void taetigeBefoerderung() {
        erhoeheGehalt( 500);
    } // taetigeBefoerderung

} // Angestellter
```

Es kann auch ein parameterloser Konstruktor definiert werden für den Fall, dass bei der Objekterzeugung keine Werte angegeben werden. Dann wird z.B. bei den Managern (unrealistisch!) angenommen, dass sie 1945 geboren wurden und 10.000 EURO im Monat verdienen.

```
public class Manager extends Mitarbeiter {

    public Manager() {
        super("Unbekannt",10000,1945);
    } // parameterloser Konstruktor

    public Manager(String name, int gehalt, int geburtsjahr) {
        super(name,gehalt,geburtsjahr);
    } // parametrisierter Konstruktor

    public void taetigeBefoerderung() {
        erhoeheGehalt(1000);
    } // taetigeBefoerderung

} // Manager
```



**Übung OOP4**

- a) Implementieren Sie die Klasse Mitarbeiter und die Methode „taetigeBefoerderung()“ als abstrakt (**Mitarbeiter.java**).
- b) Führen Sie die 3 neuen abgeleiteten Klassen Angestellter (**Angestellter.java**), Arbeiter (**Arbeiter.java**) und Manager (**Manager.java**) ein.
- c) Implementieren Sie die abgeleitete Klasse Auszubildender (**Auszubildender.java**). Die zusätzlichen Attribute sind „ausbildungsbeginn“ und „ausbildungsdauer“. Die zusätzlichen Methoden sind „ermittleAusbildungsjahr(int AktJahr)“, „setAusbildungsdauer(int Dauer)“, „getAusbildungsdauer()“, „setAusbildungsbeginn(int )“ und „getAusbildungsbeginn()“. Da ein Azubi nicht befördert werden kann, darf die ursprüngliche Methode „taetigeBefoerderung()“ keine Auswirkung auf das Gehalt haben. Jedes Ausbildungsjahr bewirkt aber 200 EURO mehr Ausbildungsbeihilfe.
- d) Schreiben Sie ein Hauptprogramm (**OOP4.java**), in welchem Sie **4 Objekte** erzeugen, denn die Firma hat 4 Mitarbeiter.  
Der **Arbeiter Herr Schulz** verdient im Jahr 2000 2600 EURO. Er ist Jahrgang 1962.  
Der **Angestellte Herr Meyer** verdient im selben Jahr 3000 EURO. Er ist 1958 geboren und seit 1981 in der Firma. Der **Manager heißt Herr Schneider**.  
Die **Auszubildende Frau Lange** beginnt gerade ihre dreijährige Ausbildung mit einer Ausbildungsbeihilfe von 600 EURO. Sie ist 16 Jahre alt.
- e) Realisieren Sie einen **Ausgabealgorithmus**, der für die Jahre 2000 bis einschließlich 2003 die Angabe des Alters und des Gehalts jeden Mitarbeiters auf dem Bildschirm ausgibt.
- f) Beachten Sie bei der **Einkommensentwicklung** bitte folgendes:  
Die **Auszubildende** wird nach der Ausbildung leider nicht übernommen.  
**Arbeiter** bekommen in der Regel 3% Lohnerhöhung, es sei denn, sie wurden im Jahr davor entweder 40 oder 60 Jahre alt. Dann werden sie befördert.  
**Angestellte** erhalten in der Regel 4% Gehaltserhöhung, es sei denn, sie hatten im Jahr zuvor ein Dienstjubiläum (alle 10 Jahre). Dann werden sie befördert.  
Der **Manager** wird jedes Jahr befördert.
- g) Führen Sie bitte zuerst einen **Schreibetischtest** aus:

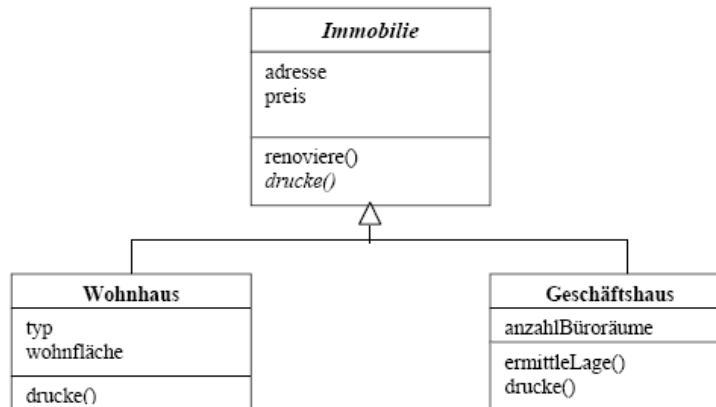
Mitarbeiter	Jahr	Alter	Gehalt	Beförderung im nächsten Jahr
Azubi Frau Lange	2000	16		nein
Arbeiter Herr Schulz	2000		2600	nein
Angestellter Herr Meyer	2000	42		nein
Manager Schneider	2000		10.000	ja
Azubi Frau Lange	2001		800	nein
Arbeiter Herr Schulz	2001			
Angestellter Herr Meyer	2001		3120	
Manager Schneider	2001			
Azubi Frau Lange	2002			
Arbeiter Herr Schulz	2002			
Angestellter Herr Meyer	2002			
Manager Schneider	2002			
Arbeiter Herr Schulz	2003			
Angestellter Herr Meyer	2003			
Manager Schneider	2003			

- h) Zeichnen Sie das UML-Klassendiagramm (s. Seite 21) und das UML-Objektdiagramm (für das Jahr 2000) mit Hilfe eines UML-Tools!

## OOO5: Polymorphismus

### Abstrakte Klasse

Wenn von einer Oberklasse **keine Objekte** erzeugt werden, dann ist sie abstrakt. Sie kann auch **abstrakte Operationen** enthalten. Diese müssen dann in den Unterklassen implementiert werden.



### Polymorphismus

Der Polymorphismus erlaubt es, **gleiche Botschaften** (wie z.B. `drucke()`) **an Objekte unterschiedlicher Klassen zu senden**. Die aufgerufenen Operationen können zu unterschiedlichen Ergebnissen führen, bei einem Wohnhaus z.B. den Grundriss drucken, bei einem Geschäftshaus hingegen die Fassadenansicht. Steht beim Kompilieren noch nicht fest, zu welchem Typ ein Objekt gehört (ist z.B. das Haus ein Wohnhaus oder ist es ein Geschäftshaus?) oder wechselt ein Objekt während des Programmlaufs seine Identität spricht man vom „**späten Binden**“ (late binding).

```

//import Immobilie;
//import Geschaeftshaus;
//import Wohnhaus;

public class ImmobilienTest {

    public static void main(String[] args) {

        Immobilie einHaus; // Immobilie ist abstrakte Klasse!

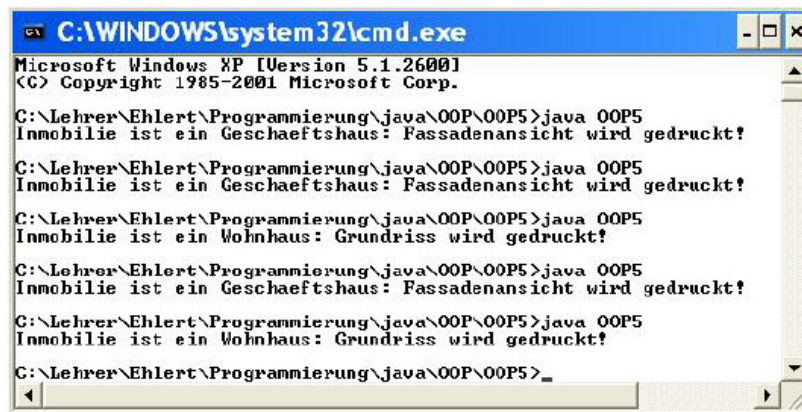
        if ((int) (Math.round(Math.random())) == 0) // Zufallszahl: 0 oder 1
            einHaus = new Geschaeftshaus ();
        else
            einHaus = new Wohnhaus ();

        einHaus.drucke(); // Fassadenansicht ODER Grundriss!

    } // main
} // ImmobilienTest
    
```

**Übung OOP5**

- a) Implementieren Sie die abstrakte Klasse **Immobilie** mit der Adresse als String- und dem Preis als int-Datentyp. Die Methode „drucke()“ ist abstrakt. Durch eine Renovierung wird der Preis der Immobilie um 20% gesteigert!
- b) Implementieren Sie die abgeleitete Klasse **Wohnhaus**. Der Typ soll als String- und die Wohnfläche als float-Datentyp deklariert werden. Die Methode „drucke()“ bewirkt die Bildschirmausgabe „Die Immobilie ist ein Wohnhaus: Grundriss wird gedruckt!“.
- c) Implementieren Sie die abgeleitete Klasse **Geschäftshaus**. Die Anzahl der Büroräume soll als int-Datentyp deklariert werden. Die Methode „drucke()“ bewirkt die Bildschirmausgabe „Die Immobilie ist ein Geschäftshaus: Fassadenansicht wird gedruckt!“.  
Die Methode „ermittleLage()“ liefert einen String zurück, der „Citylage“ enthält.
- d) Implementieren Sie die Klasse **Immobilientest** bzw. **OOP5** (s. Seite 24) und beweisen Sie das sog. „späte Binden“: Auf dem Bildschirm sollte nach Programmstart mal „Die Immobilie ist ein Wohnhaus: Grundriss wird gedruckt!“, mal „Die Immobilie ist ein Geschäftshaus: Fassadenansicht wird gedruckt!“ erscheinen.



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600.1]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP5>java OOP5
Immobilie ist ein Geschäftshaus: Fassadenansicht wird gedruckt!

C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP5>java OOP5
Immobilie ist ein Geschäftshaus: Fassadenansicht wird gedruckt!

C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP5>java OOP5
Immobilie ist ein Wohnhaus: Grundriss wird gedruckt!

C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP5>java OOP5
Immobilie ist ein Geschäftshaus: Fassadenansicht wird gedruckt!

C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP5>java OOP5
Immobilie ist ein Wohnhaus: Grundriss wird gedruckt!

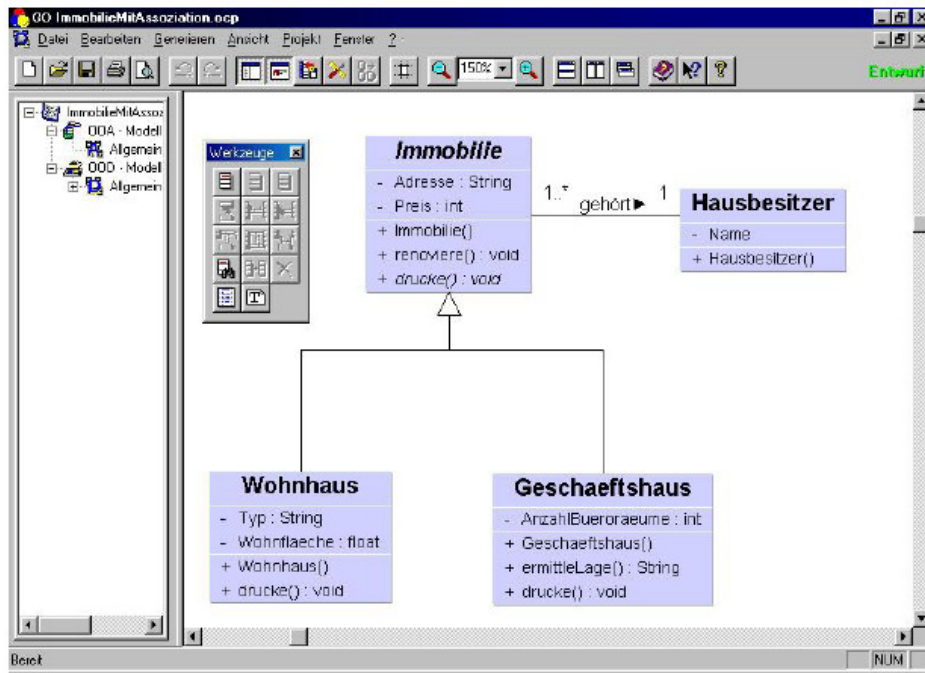
C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP5>_
  
```

**OOB6: Assoziationen**

Eine Assoziation modelliert **Verbindungen** zwischen Objekten einer oder mehrerer Klassen. Sie modelliert dabei stets **Beziehungen zwischen Objekten**, nicht zwischen Klassen. Es ist jedoch auch üblich, von einer Assoziation zwischen Klassen zu sprechen.

Im unteren Klassendiagramm erkennt man: Zu jeder Immobilie **gehört** genau **ein** Hausbesitzer, ein Hausbesitzer kann aber **mehrere** Immobilien haben. Dies entspricht einer 1-n-Kardinalität.

**UML-Klassendiagramm\***



\* Die Attribute sollten klein geschrieben werden

Bei der Realisierung der Assoziation zwischen Immoblie und Hausbesitzer gibt es 3 Möglichkeiten:

1. Die Hausbesitzer kennen ihre Immobilien, die Immobilien aber ihre Besitzer nicht!
2. Die Immobilien kennen ihre Besitzer, die Hausbesitzer aber ihre Immobilien nicht!  
(Dies würde man bei relationalen Datenbanksystemen für diese 1-n-Beziehung modellieren)
3. Beide Seiten verweisen aufeinander, kennen sich also!

Ein „n-Verweis“ wird in der OOB oft mit einer Liste realisiert, die die Referenzen auf die einzelnen Objekte enthält.

Bei einem „1-Verweis“ genügt eine Referenz auf das Objekt.

Im folgenden Beispiel wird die Klasse Immoblie nicht verändert, sondern nur eine Liste bzw. ein Feld in der Klasse Hausbesitzer integriert, somit also die erste Assoziations-Möglichkeit realisiert.

Name:

Klasse:

Datum:

Blatt Nr.: 27/32

**Umsetzung einer „n-Assoziation“ mit Java**

```
/**
 * file      Hausbesitzer.java
 * description Klasse für die OOP6
 * @author    Ehlert / OSZ IMT
 * @version   1.0    / Oktober 2005
 */

public class Hausbesitzer {

    private String name;

    // Realisierung der Assoziation zur Klasse Immobilie
    // Einschränkung: Ein Hausbesitzer kann max. 10 Immobilien besitzen
    private int anzahlHaeuser = 0;
    private Immobilie immobilienListe[] = new Immobilie[10];

    // parameterloser Konstruktor
    public Hausbesitzer() {
        name = "Unbekannt";
        anzahlHaeuser = 0;
    }

    // parametrisierter Konstruktor
    public Hausbesitzer(String name) {
        this.name = name;
        anzahlHaeuser = 0;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return this.name;
    }

    public int getAnzahlHaeuser() {
        return this.anzahlHaeuser;
    }

    public void setLinkHaus(Immobilie haus) {
        immobilienListe[anzahlHaeuser] = haus;
        anzahlHaeuser++;
    }

    public Immobilie getLinkHaus(int nummer) {
        return immobilienListe[nummer-1];
    }

    public void removeLinkHaus(Immobilie Haus) {
        // ...;
    }
}
```

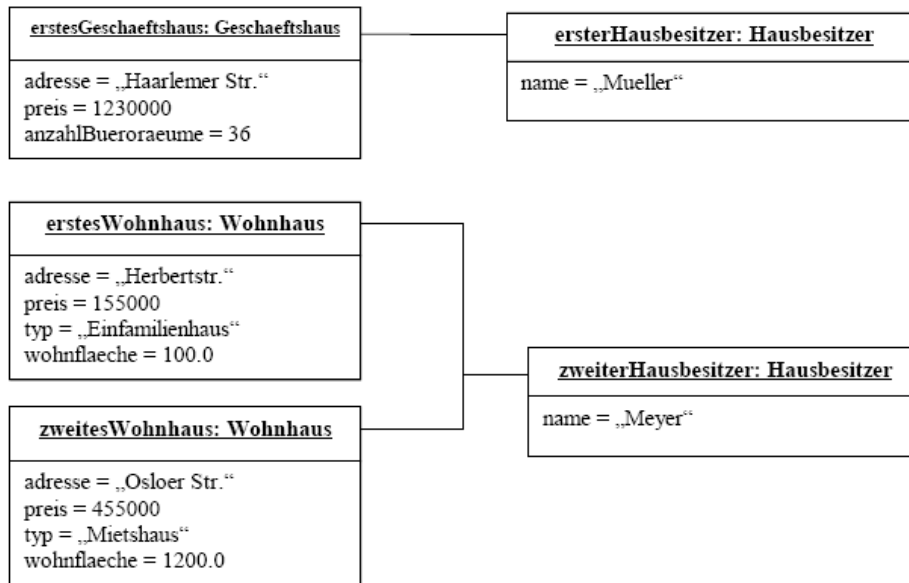
Name:	Klasse:	Datum:	Blatt Nr.: 28/32
-------	---------	--------	------------------

**UML-Objektdiagramm**

Neben den UML-Klassendiagrammen gibt es die UML-Objektdiagramme.

Sie stellen einen Realitätsausschnitt konkret dar, z.B.:

- Herr Müller besitzt ein Geschäftshaus,
- Herr Meyer besitzt zwei Wohnhäuser sein Eigentum.



Objektdiagramme sind zeitlich abhängig. Durch einen Immobilienverkauf oder –ankauf, bzw. auch bei einer Renovierung ändern sich die Zustände der Objekte.

Bei der Klassen-Modellierung muss also geguckt werden, ob alle in Frage kommenden Zustände der Objekte durch das Klassendiagramm berücksichtigt sind!

Name:	Klasse:	Datum:	Blatt Nr.: 29/32
-------	---------	--------	------------------

**Übung OOP6**

- a) Erstellen Sie mit Hilfe eines UML-Tools das **OOD-UML-Klassendiagramm**.
- b) Erstellen Sie mit Hilfe des UML-Tools das **UML-Objektdiagramm**.
- c) Falls das UML-Tool es zulässt, generieren Sie automatisch den Code und geben Sie eine Einschätzung über die Güte der Code-Fragmente.
- d) Gegeben ist ein Teil der Implementierung des Anwendungsprogrammes (**OOP6.java** bzw. **HausbesitzerTest.java**), welches das vorgegebene Objektdiagramm realisiert:

```
public class OOP6 {

    public static void main(String[] args) {

        Wohnhaus      erstesWohnhaus      =
            new Wohnhaus("Herbertstr.",155000,"Einfamilienhaus",100.0f);
        Wohnhaus      zweitesWohnhaus     =
            new ...
        Geschaeftshaus erstesGeschaeftshaus =
            new ...
        Hausbesitzer  ersterHausbesitzer  = new Hausbesitzer("Mueller");
        Hausbesitzer  zweiterHausbesitzer = new ...

        ersterHausbesitzer.setLinkHaus(erstesGeschaeftshaus);
        ...
        ...

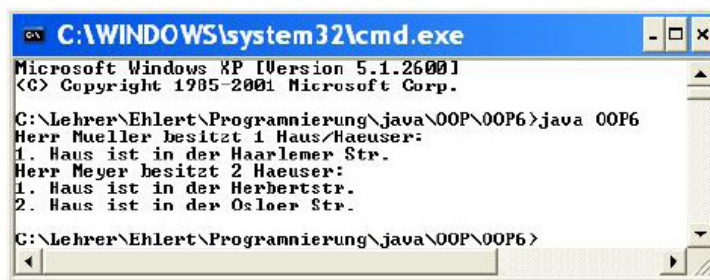
        System.out.println("Herr " + ersterHausbesitzer.getName() + " besitzt " +
            ersterHausbesitzer.getAnzahlHaeuser() + " Haus/Haeuser: ");
        for (int i = 1; i <= ersterHausbesitzer.getAnzahlHaeuser(); i++) {
            System.out.println(i + ". Haus ist in der " +
                ersterHausbesitzer.getLinkHaus(i-1).getAdresse());
        } // for

        System.out.println("Herr " + zweiterHausbesitzer.getName() + " besitzt " +
            zweiterHausbesitzer.getAnzahlHaeuser() + " Haeuser: ");
        for (int i = 1; i <= zweiterHausbesitzer.getAnzahlHaeuser(); i++) {
            System.out.println(i + ". Haus ist in der " +
                zweiterHausbesitzer.getLinkHaus(i-1).getAdresse());
        } // for

    } // main
} // OOP6
```

Implementieren Sie den gesamten Code für die Klasse **Hausbesitzer** und das Anwendungsprogramm **OOP6**.

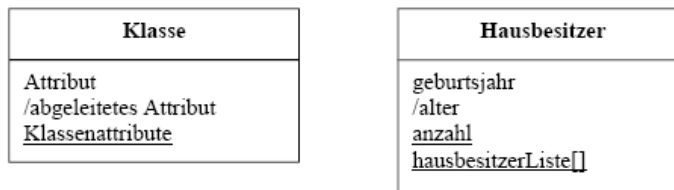
Die Klassen **Immobilie**, **Geschaeftshaus** und **Wohnhaus** können Sie aus OOP5 übernehmen!



## OOB7: Klassenattribute und Klassenmethoden

### Attribute:

Ein **Attribut** beschreibt eine Eigenschaft einer Klasse bzw. ein Datum eines Objekts der Klasse. Der Wert eines **abgeleiteten Attributs** kann jederzeit aus anderen Attributwerten berechnet werden. Ein **Klassenattribut** liegt vor, wenn nur ein Attributwert für alle Objekte einer Klasse existiert.



Die Anzahl soll die „Anzahl der Hausbesitzer“ sein. Dieses Attribut ist somit nicht jedem „Objekt Hausbesitzer“ zugeordnet, sondern nur der „Klasse Hausbesitzer“! Genauso wie die „Liste der Hausbesitzer“.

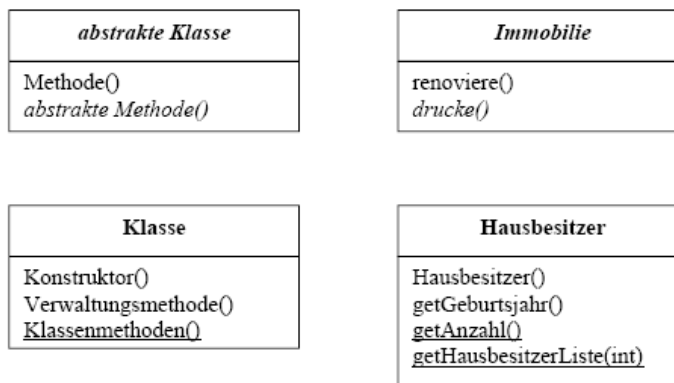
### Methoden:

Eine **Methode** beschreibt das Verhalten eines Objekts bzw. einer Klasse.

Eine **Klassenmethode** ist eine Operation, die der jeweiligen Klasse zugeordnet ist und nicht auf ein einzelnes Objekt der Klasse angewendet werden kann.

**Verwaltungsmethoden** ( `setAttribute()`, `getAttribute()`, `setLink()`, `getLink()`, `removeLink()` etc. ) und **Konstruktormethoden** werden aus Gründen der Lesbarkeit oft nicht in das Klassendiagramm eingetragen.

**Abstrakte Operationen** sind in der Oberklasse nicht implementiert und werden durch die Unterklassen überschrieben. Abstrakte Operationen gibt es nur in abstrakten Klassen.





**Klassenattribute und Klassenmethoden in Java**

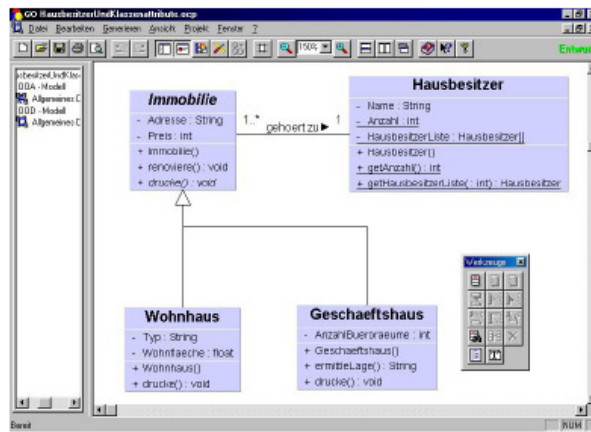
Bei der letzten Übung musste für jeden Hausbesitzer ein eigener Ausgabealgorithmus geschrieben werden. Viel besser wäre es, wenn ein Algorithmus für alle Hausbesitzer genügen würde:

```

for (int i = 1; i <= Hausbesitzer.getAnzahl(); i++) {
    System.out.println("Herr/Frau " +
        Hausbesitzer.getHausbesitzerListe(i-1).getName() + " besitzt " +
        Hausbesitzer.getHausbesitzerListe(i-1).getAnzahlHaeuser() + " Haus/Haeuser: ");
    for (int j = 1; j <= Hausbesitzer.getHausbesitzerListe(i-1).getAnzahlHaeuser(); j++) {
        System.out.println(j + ". Haus ist in der " +
            Hausbesitzer.getHausbesitzerListe(i-1).getLinkHaus(j-1).getAdresse());
    } // for
} // for

```

Die Klasse Hausbesitzer stellt hierbei zwei sog. **Klassenmethoden** (getAnzahl() und getHausbesitzerListe(int)) zur Verfügung. Sie beziehen sich auf zwei sog. **Klassenattribute** (anzahl und hausbesitzerListe[]). Ein Klassenattribut hat für alle Objekte den gleichen Wert, da es nur zur Klasse zählt und damit nur einen Speicherplatz belegt. In der UML-Darstellung werden Klassenattribute und Klassenmethoden durch einen Unterstrich gekennzeichnet:



Die Implementierung ist wie folgt:

```

private static int anzahl = 0; // Klassenattribut: Anzahl der Hausbesitzer
private static Hausbesitzer hausbesitzerListe[] = new Hausbesitzer[10];

public Hausbesitzer(String name) {
    this.name = name;
    anzahlHaeuser = 0;
    hausbesitzerListe[anzahl] = this;
    anzahl++;
} // parametrisierter Konstruktor

public static int getAnzahl() {
    return anzahl;
} // Klassenmethode getAnzahl

public static Hausbesitzer getHausbesitzerListe(int nummer) {
    return hausbesitzerListe[nummer];
} // Klassenmethode getHausbesitzerListe

```

Name:

Klasse:

Datum:

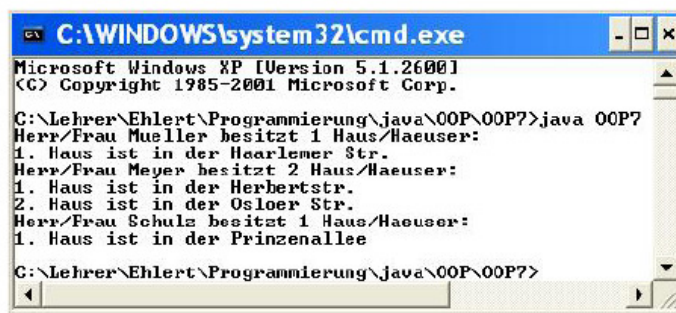
Blatt Nr.: 32/32

**Übung OOP7**

Gegeben ist das **Objektdiagramm** aus der OOP6!

Zusätzlich gibt es einen dritten Hausbesitzer („Schulz“), dem das dritte Wohnhaus gehört („Prinzenallee“, 350000, „Mietshaus“, 800.0).

- a) Implementieren Sie in der **Hausbesitzer** Klasse die beiden **Klassenattribute** und die zwei **Klassenmethoden**. Ergänzen Sie beide Konstruktoren.
- b) Implementieren Sie ein **Anwendungsprogramm (OOP7.java)**, welches die 3 Hausbesitzer, die 4 Häuser und die entsprechenden Assoziationen bildet.
- c) Geben Sie am Ende des Anwendungsprogramms alle Hausbesitzer mit ihren Häusern mit Hilfe eines einzigen Ausgabealgorithmus' aus.



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

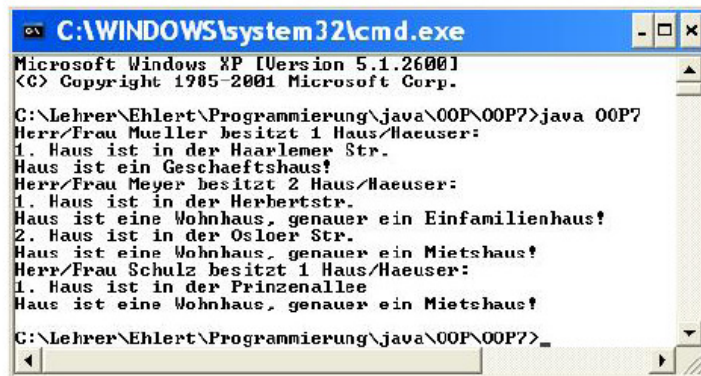
C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP7>java OOP7
Herr/Frau Mueller besitzt 1 Haus/Haeuser:
1. Haus ist in der Haarlemer Str.
Herr/Frau Meyer besitzt 2 Haus/Haeuser:
1. Haus ist in der Herbertstr.
2. Haus ist in der Osloer Str.
Herr/Frau Schulz besitzt 1 Haus/Haeuser:
1. Haus ist in der Prinzenallee

C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP7>

```

- d) Versuchen Sie noch einen Hinweis darauf zu geben, ob die Immobilie ein Wohnhaus (spezifiziert nach Typ) oder ein Geschäftshaus ist:

```
if (einHaus instanceof Wohnhaus) { ... }
```



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP7>java OOP7
Herr/Frau Mueller besitzt 1 Haus/Haeuser:
1. Haus ist in der Haarlemer Str.
Haus ist ein Geschaeftshaus!
Herr/Frau Meyer besitzt 2 Haus/Haeuser:
1. Haus ist in der Herbertstr.
Haus ist eine Wohnhaus, genauer ein Einfamilienhaus!
2. Haus ist in der Osloer Str.
Haus ist eine Wohnhaus, genauer ein Mietshaus!
Herr/Frau Schulz besitzt 1 Haus/Haeuser:
1. Haus ist in der Prinzenallee
Haus ist eine Wohnhaus, genauer ein Mietshaus!

C:\Lehrer\Ehlert\Programmierung\java\OOP\OOP7>_

```

## U) Google-Ergebnisse: OOP first und OOP later (14.01.2010)

The screenshot shows a Mozilla Firefox browser window with the title "OOP first - Google-Suche - Mozilla Firefox". The address bar contains the URL "http://www.google.de/search?q=OOP+first&". The search bar has "OOP first" entered. Below the search bar, there are radio buttons for "Suche: Das Web", "Seiten auf Deutsch", and "Seiten aus Deutschland". The search results are displayed below, showing a list of 1-10 results out of approximately 47,800,000 for "OOP first" in 0.31 seconds. The results include:

- OOP im Anfängerunterricht von Anfang an?**  
Dateiformat: PDF/Adobe Acrobat - [Schnellansicht](#)  
Die Themen von OOP-First sind vergleichbar mit den Themen von OOP-Later, ...  
Sequence of topics taught so far in the OOP-first group: ...  
[www.ehlerl.oszimt.de/dokumente/INF OS-2009-Ehlerl.pdf](http://www.ehlerl.oszimt.de/dokumente/INF OS-2009-Ehlerl.pdf)
- OOP im Anfängerunterricht von Anfang an?**  
Dateiformat: PDF/Adobe Acrobat - [Schnellansicht](#)  
Ich möchte im folgenden den Begriff OOP-First verwenden, wenn im ..... Mit dem OOP-First-Einstieg wird (kann) in der Regel schneller unterrichtet. (werden)! ...  
[www.ehlerl.oszimt.de/dokumente/FU-2008-Ehlerl3.pdf](http://www.ehlerl.oszimt.de/dokumente/FU-2008-Ehlerl3.pdf)
- [Weitere Ergebnisse anzeigen von www.ehlerl.oszimt.de](#)
- Object-oriented programming - Wikipedia, the free encyclopedia** - [ [Diese Seite übersetzen](#) ]  
The Simula programming language was the first to introduce the concepts underlying object-oriented programming (objects, classes, subclasses, ...  
[en.wikipedia.org/wiki/Object-oriented\\_programming](http://en.wikipedia.org/wiki/Object-oriented_programming) - [Im Cache](#)
- OOPHub.com: Learn OOP First, UML Afterwards** - [ [Diese Seite übersetzen](#) ]  
28 Jul 2009 ... If you are relatively new to software development, you may ask what you shall learn first OOP or UML. This is somewhat like the chicken or ...  
[www.oophub.com/.../learn-oop-first-uml-afterwards.html](http://www.oophub.com/.../learn-oop-first-uml-afterwards.html) - [Im Cache](#) - [Ähnlich](#)
- WikiAnswers - When and where was the first ally oop play** - [ [Diese Seite übersetzen](#) ]  
I believe the first ally oop play occurred in 1945 during a game between Savannah, Tn., and Selmer Tn. high schools. Can anyone.  
[wiki.answers.com/.../When\\_and\\_where\\_was\\_the\\_first\\_ally\\_ooop\\_play](http://wiki.answers.com/.../When_and_where_was_the_first_ally_ooop_play) - [Im Cache](#) - [Ähnlich](#)
- First Look: Object Oriented CSS** - [ [Diese Seite übersetzen](#) ]  
16 Jun 2009 ... I first did OOP in 1986 and I've used it intermittently ever since. I love it! When I saw the title of the article, the excitement welled up ...  
[www.sitepoint.com/.../first-look-object-oriented-css/](http://www.sitepoint.com/.../first-look-object-oriented-css/) - [Vereinigte Staaten](#) - [Im Cache](#) - [Ähnlich](#)
- Amazon.com: Head First Object-Oriented Analysis and Design ...** - [ [Diese Seite übersetzen](#) ]  
Head First Object-Oriented Analysis & Design shows you how to analyze, design, and write serious object-oriented software: software that's easy to reuse, ...  
[www.amazon.com](http://www.amazon.com) > ... > [Methodology](#) - [Im Cache](#) - [Ähnlich](#)
- [Tutor] PYTHON OOP First Project - Coke Machine** - [ [Diese Seite übersetzen](#) ]  
8 Jun 2000 ... For my first Python > project, I am embarking on a OOP project using > Python to simulate a Coca-Cola (or Pepsi :-)) > vending machine. ...  
[mail.python.org/pipermail/tutor/2000-June/001621.html](http://mail.python.org/pipermail/tutor/2000-June/001621.html)  
[\[Tutor\] Help\(\) with OOP/Composition from "Learning Python"](#) - 21. Okt. 2005  
[\[Edu-sig\] Best approach to teaching OOP and graphics](#) - 23. März 2005  
[\[Tutor\] First OOP Attempt](#) - 31. Jan. 2001  
[Weitere Ergebnisse von mail.python.org >](#)
- eBay Österreich: BIG IDEA - S/T FIRST - Canada-AOR/MR PARADOX OOP ...**  
11. Nov. 2009 ... Finden Sie BIG IDEA - S/T FIRST - Canada-AOR/MR PARADOX OOP RARE!! in Musik , CDs, Rock Underground, Hardrock bei eBay.at.  
[cgi.ebay.at/ws/eBayISAPI.dll?ViewItem&item...](http://cgi.ebay.at/ws/eBayISAPI.dll?ViewItem&item...) - [Im Cache](#)
- Amazon.de: Ein Kundess Rezension von Thinking in Java**  
Again however you need a basic understanding of OOP first. (C++ and Java syntax are not enough, this book really goes into the OOP stuff pretty detailed and ...  
[www.amazon.de/review/R1NYOJ2MCEUQKZ](http://www.amazon.de/review/R1NYOJ2MCEUQKZ) - [Im Cache](#)

The browser status bar at the bottom shows "Fertig".

OOM later - Google-Suche - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://www.google.de/search?q=OOM+later

OSZ IMT Stundenpläne Mail BSCW Fortbildungen Cisco Academy Sun Initiative

intern © OSZ IMT - Oberstufenzentrum... OOM later - Google-Suche

Web Bilder Videos Maps News Shopping E-Mail Mehr Webprotokoll Sucheinstellungen Anmelden

Google OOM later Suche [Erweitere Suche](#)

Suche:  Das Web  Seiten auf Deutsch  Seiten aus Deutschland

Web [Optionen anzeigen...](#) Ergebnisse 1 - 10 von ungefähr 29.800.000 für OOM later. (0,29 Sekunden)

**OOM im Anfängerunterricht von Anfang an?**  
 Dateiformat: PDF/Adobe Acrobat - [Schnellansicht](#)  
 Die Themen von OOP-First sind vergleichbar mit den Themen von OOP-Later, ...  
 Sequence of topics taught so far in the OOP-later group: ...  
[www.ehlert.oszimt.de/dokumente/INFOS-2009-Ehlert.pdf](http://www.ehlert.oszimt.de/dokumente/INFOS-2009-Ehlert.pdf)

**OOM im Anfängerunterricht von Anfang an?**  
 Dateiformat: PDF/Adobe Acrobat - [Schnellansicht](#)  
 Dagegen setze ich den Begriff OOP-Later, wenn ein Teil der prozeduralen ..... Der  
 Einstieg in die OOP mit BlueJ fällt OOP-Later-Schülern leichter als OOP- ...  
[www.ehlert.oszimt.de/dokumente/FU-2008-Ehlert3.pdf](http://www.ehlert.oszimt.de/dokumente/FU-2008-Ehlert3.pdf)

[Weitere Ergebnisse anzeigen von www.ehlert.oszimt.de](#)

**MNUBerlin08Informatikprogramm - DDI - Twiki**  
 OOP-First-Einstieg mit dem OOP-Later-Einstieg. Dabei werden zwei Schulklassen ein  
 Jahr lang in zweierlei Hinsicht untersucht: ...  
<https://www.inf.fu-berlin.de/.../MNUBerlin08Informatikprogramm> - [Im Cache](#)

**PubZone - Unterschiede im Lernerfolg von Schülerinnen und Schülern ...**  
 Unterschiede im Lernerfolg von Schülerinnen und Schülern in Abhängigkeit von der  
 zeitlichen Reihenfolge der Themen (OOP-First bzw. OOP-Later). ...  
[www.pubzone.org/dblp/conf/schule/EhlertSD9](http://www.pubzone.org/dblp/conf/schule/EhlertSD9) - [Im Cache](#)

**OOPPhoto - Moving To OOP - Adding Service Objects** - [ [Diese Seite übersetzen](#) ]  
 18 Jul 2008 ... And, while this is an attempt at learning OOP, phase one consisted of a  
 working ... and then worry about refactoring it into OOP later. ...  
[www.bennadel.com/index.cfm?dax=blog:1292...](http://www.bennadel.com/index.cfm?dax=blog:1292...) - [Im Cache](#) - [Ähnlich](#)

**Oops I'm Late!** - [ [Diese Seite übersetzen](#) ]  
 Oops I'm Late! is software for mobile phones that sends notifications based ..... Oops  
 I am Late! - приложение, которое сообщит коллегам о вашем опоздании ...  
[www.oopsimlate.com/](http://www.oopsimlate.com/) - [Im Cache](#) - [Ähnlich](#)

**OOP question:Late static binding and generic classes - SitePoint ...** - [ [Diese Seite übersetzen](#) ]  
 7 Einträge - 4 Autoren - Letzter Eintrag: Vor 2 Tagen  
 OOP question:Late static binding and generic classes: PHP. ... not work as expected  
 before php 5.3 which doesnt support late static binding. ...  
[www.sitepoint.com/forums/showthread.php?p=4482161](http://www.sitepoint.com/forums/showthread.php?p=4482161) - [Vereinigte Staaten](#)

**Tennis Tipping #3 - Chennai - Monday OoP up! Send picks before 7 ...** - [ [Diese Seite übersetzen](#) ]  
 5 Einträge - 5 Autoren  
 Schedule is very late today. Can anyone help me post the OoP later? Picks should  
 count for Day 1 of 2 for Both Singles and Doubles TT QF. ...  
[www.menstennisforums.com/showthread.php?p=9468962](http://www.menstennisforums.com/showthread.php?p=9468962) - [Im Cache](#)

**OOP 2010 - Sessiondetails**  
 Schnell & Direkt. Aktuelle Neuigkeiten, RSS Feed, OOP Podcasts ... S.O.L.I.D.-  
 Fifteen Years Later. 28.01.2010. Uhrzeit: 14:30 - 15:30 ...  
[www.sigs-datacom.de/.../sessiondetails.html?](http://www.sigs-datacom.de/.../sessiondetails.html?) - [Im Cache](#)

**OOM Learners Views on Objects-First and Objects-Later - Results of an ...** - [ [Diese Seite übersetzen](#) ]  
 Dateiformat: PDF/Adobe Acrobat - [Schnellansicht](#)  
 von A Ehlert - [Zitiert durch: 1](#) - [Ähnliche Artikel](#) - [Alle 3 Versionen](#)  
 Figure 5: Sequence of modules taught so far in the OOP-later group (Note: The .... For  
 the "OOM later"-group we see the following interesting aspects: ...  
[www.cs.umu.se/~jubo/Meetings/EC/OOP07/Submissions/EhlertSchulte.pdf](http://www.cs.umu.se/~jubo/Meetings/EC/OOP07/Submissions/EhlertSchulte.pdf)

Fertig

## Zusammenfassung der Ergebnisse

Die Studie erbrachte u. a. die folgenden (hier verkürzt dargestellten) Resultate [ES09c, S. 132] [ES09a, S. 24]:

- Es konnten **neun Themengebiete** generiert werden, die sowohl beim OOP-First-Ansatz als auch beim OOP-Later-Ansatz zur Geltung kamen.
- Es wurden keine signifikanten Unterschiede zwischen dem OOP-First-Vorgehen und dem OOP-Later-Vorgehen im Hinblick auf den **Lernerfolg** der Schülerinnen und Schüler festgestellt bzw. gemessen (die p-Werte für die elf abgefragten Themen liegen zwischen 0,18 und 0,83).
- Die OOP-Later-Schüler hatten ein (oft signifikant) besseres **subjektives Erleben**, sowohl in den Frage-Bereichen („Schule“, „Fach Informatik“ und „Thema“) als auch in den Erlebnis-Dimensionen (emotionales, kognitives und motivationales Erleben). Das subjektive Erleben der Themen korrelierte allerdings nur schwach (OOP-First) bis mittel (OOP-Later) mit den Ergebnissen des Nachhaltigkeitstests.
- Das **emotionale Erleben** (ermittelt durch einen „Wohlfühlfaktor“ und einen „Stressfaktor“) war in der OOP-First-Klasse nicht so gut wie in der OOP-Later-Klasse, die Unterschiede waren meist signifikant.
- Im Hinblick auf das **motivationale Erleben** war in allen drei Frage-Bereichen die OOP-Later-Klasse (signifikant) höher motiviert als die OOP-First-Klasse.
- Der Unterricht eines OOP-Later-Vorgehens scheint mehr auf die vorher unterrichteten Themen aufzubauen (ein Aspekt des **kognitiven Erlebens**) als der Unterricht eines OOP-First-Vorgehens.
- Es gab eine mittlere Korrelation bei der OOP-Later-Klasse zwischen dem **empfundenen Schwierigkeitsgrad** der Themen (ein weiterer Aspekt des **kognitiven Erlebens**) und den Ergebnissen des Nachhaltigkeitstests, also dem **tatsächlichen Schwierigkeitsgrad**. Bei der OOP-First-Klasse war die Korrelation nur schwach.
- In den beiden Klassen gab es teilweise große Unterschiede in dem **empfundenen Schwierigkeitsgrad** der einzelnen Themen, z. B. bei den Themen „Vererbung“, „Arrays und Strings“ und „Assoziation“. Der **tatsächliche Schwierigkeitsgrad** der Themen war dagegen unabhängig von einem OOP-First- bzw. OOP-Later-Vorgehen. Die schwierigsten Themen im Test waren in beiden Klassen die Themen „Assoziation“ und „Arrays“.
- Das **BlueJ-Tool** wurde von beiden Klassen positiv erlebt.
- **Objektorientierte Themen** waren nicht schwieriger als **prozedurale Themen**. In beiden Themenarten fanden sich schwere und leichte Themen wieder.
- **Objektorientierte Modellierung** konnte in den beiden OOP-Kursen gut vermittelt werden. Die Lernergebnisse bezogen auf die **OOM** waren unabhängig von einem OOP-First- oder OOP-Later-Vorgehen.
- Bei der Lerngruppe der **leistungsschwächsten Schülerinnen und Schüler** zeigten die OOP-Later-Schüler (im Nachhaltigkeitstest) einen besseren Lernerfolg als die OOP-First-Schüler. Die spezielle Präparierung leistungsschwacher Schüler auf den Vergleichstest hin zeigte eine (im Gegensatz zu den leistungsstarken Schülern) offensichtliche Auswirkung, die aber nicht nachhaltig war.
- Es gab bei der Lerngruppe der **leistungsstärksten Schülerinnen und Schüler** keine signifikanten Unterschiede im Lernerfolg in Abhängigkeit von der fachdidaktischen Vorgehensweise (der p-Wert für das Gesamtergebnis im Nachhaltigkeitstest beträgt 0,90). Für sie scheint aber im Hinblick auf das subjektive Erleben das OOP-First-Vorgehen von Vorteil zu sein.
- Für **Schülerinnen** scheint dagegen das OOP-Later-Vorgehen besser geeignet zu sein. Aber dieses Resultat sollte man angesichts der kleinen Lerngruppe von jeweils nur zwei Personen mit Vorsicht betrachten.

Ausgehend von dem Modell, dass sich ein OOP-First- von einem OOP-Later-Vorgehen nur in der zeitlichen Reihenfolge gleichartiger Themen unterscheidet, deuten die Resultate der Studie darauf hin, dass die Debatte um OO-First bzw. objects-first evtl. von geringerer Bedeutung ist, als bisher angenommen. Die Debatte sollte sich m. E. eher auf die Vor- und Nachteile jedes einzelnen Weges konzentrieren, speziell auf die Prozessunterschiede.

Die Fragestellung braucht in Zukunft nicht mehr zu lauten, mit welchem der beiden Paradigmen die Lehrerin bzw. der Lehrer in den Informatik-Anfangsunterricht einsteigt, da bezogen auf ein Schuljahr (mit ca. 100 Schulstunden) die Klassen den gleichen Lernerfolg hatten. Der Unterrichtende muss sich vielmehr fragen, warum bestimmte Themen von den Schülern als schwierig erlebt bzw. mangelhaft gelernt werden, warum bestimmte Themenübergänge Schwierigkeiten bereiten und wie das subjektive Erleben und der Lernerfolg gesteigert werden können. Auf den Informatik-Anfangsunterricht am OSZ IMT bezogen könnte man z. B. das Thema „Assoziation“ aus dem ersten Schuljahr herausnehmen und mit der gewonnenen Zeit das vertiefende bzw. längere Unterrichten des Themas „Arrays“ bewirken.



## **Lebenslauf**

Der Lebenslauf ist in der Online-Version aus Gründen des Datenschutzes nicht enthalten.





# Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine außer die von mir angegebenen Hilfsmittel, Hilfen und Quellen dazu verwendet habe (vgl. [FUB07, §7(4)]). Inhaltlich oder wörtlich entnommene Stellen aus anderen Werken habe ich als solche kenntlich gemacht.

Bestensee, den 8. August 2011

---

Albrecht Ehlert