

# Prozeduraler vs. objektorientierter Programmieransatz im zweiten Bildungsweg

Vergleichsstudie anhand des Lehrabschluss IT-Informatiker  
für Erwachsene mit Reha-Hintergrund

## DIPLOMARBEIT

zur Erlangung des akademischen Grades

### Magister

im Rahmen des Studiums

### Informatikmanagement

eingereicht von

**Christian Kloiber**

Matrikelnummer 0225345

an der

Fakultät für Informatik der Technischen Universität Wien

Betreuung

Betreuerin: Ass. Prof. D.I. Dr. techn. Monika Di Angelo

Wien, 20.10.2015

\_\_\_\_\_

(Unterschrift Verfasser/in)

\_\_\_\_\_

(Unterschrift Betreuer/in)



Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

## **Diplomarbeit**

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Langenzersdorf, im Oktober 2015

---

Christian Kloiber

# Danksagung

Diese Arbeit widme ich meiner Familie und Freundin, die mich tatkräftig während der Jahre meines Studiums unterstützt haben.

Während meines Informatikmanagement Studiums nahm ich schon des Öfteren bei Debatten über Vor- und Nachteile eines prozeduralen vs. objektorientierten Lehransatzes zuerst teil. Als sich diese Frage nun auch in meiner Funktion als Programmiertrainer gestellt hat, habe ich die Gelegenheit genutzt und dieses Thema zum Fokus meiner Arbeit gemacht. Im Speziellen fand ich es interessant, dass ich noch keine wissenschaftlichen Erhebungen zu diesem Thema in Bezug auf Reha-Maßnahmen für Erwachsene gefunden hatte. Besonders möchte mich an dieser Stelle auch bei meiner Betreuerin Frau Ass. Prof. D.I. Dr. techn. Di Angelo bedanken, die mich während meines Informatikstudiums erst auf den Gedanken gebracht hat überhaupt eine unterrichtende Tätigkeit in Betracht zu ziehen. Ohne ihr Engagement und den lehrreichen, aber doch unterhaltsamen, Übungen an der TU Wien, hätte ich kein Interesse an einem lehrenden Beruf entwickelt. Außerdem möchte ich mich für ihre Geduld und fachkundige Unterstützung die mir während der Betreuung dieser Arbeit entgegengebracht wurde bedanken.

Abschließend danke ich auch allen anderen Personen, die die Umsetzung der Kurse überhaupt erst möglich gemacht haben und mich bei deren Durchführung tatkräftig unterstützt haben.

## Kurzfassung

Umschulungen haben bei Instituten wie AMS, PVA, AUVA oder ähnlichen Institutionen einen hohen Stellenwert. Deshalb spezialisieren sich verschiedene Bildungsinstitutionen darauf, Menschen den Wiedereinstieg in das Berufsleben durch Umschulungen zu erleichtern. Eine dieser Umschulungen ist der Lehrabschluss für Informatiker/innen, dessen Hauptfokus auf dem Programmierunterricht liegt.

Eine besondere Zielgruppe sind erwachsene Personen mit Reha-Hintergrund. Dieser Zielgruppe wird über den zweiten Bildungsweg eine Umschulung angeboten, welche sie innerhalb von 90 Wochen fit für den IT-Arbeitsmarkt machen soll. Doch muss diese Zielgruppe nicht nur die allgemeinen Herausforderungen des Erlernens einer Programmiersprache meistern, sondern hat zumeist auch eine physische oder psychische Vorgeschichte zu bewältigen. Diese zusätzlichen Herausforderungen beeinflussen den Ausbildungsverlauf maßgeblich und müssen von Lehrenden und Lernenden ständig berücksichtigt werden.

Bei dieser Evaluierung muss noch mehr als in Schulen ein Spagat zwischen Menschlichkeit und Wirtschaftlichkeit gemeistert werden. Daher ist es hilfreich, im Rahmen einer Erhebung festzustellen, welcher Unterrichtsansatz im Hauptfach Programmieren die Teilnehmer/innen entlasten kann, aber dennoch zum gewünschten Ziel – der beruflichen Rehabilitation in Form eines IT-Lehrabschlusses – führt.

Im Rahmen dieser Arbeit werden im Programmierunterricht einerseits prozedurale und andererseits objektorientierte Lehransätze verwendet, welche auf Ihre Auswirkungen auf die Kursteilnehmer/innen untersucht werden.

Hierfür wird ein eigenes Unterrichtskonzept für den Programmierunterricht erstellt. Dieses Konzept wird in vier Gruppen, jeweils zwei pro verwendetem Lehransatz, praktisch umgesetzt. Während und nach der Unterrichtseinheit werden die Trainer/innen die Reaktionen der Kursteilnehmer/innen erfassen. Am Beginn und Ende des Kurses werden mittels Fragebogen die persönlichen Eindrücke der Kursteilnehmer/innen erhoben. Anhand dieser Ergebnisse und weiterer Unterlagen wie z.B. Tests, Fragebögen usw. wird evaluiert, inwieweit die verschiedenen Lehransätze geeignet sind um das benötigte Wissen zu vermitteln.

## **Abstract**

For institutes such as the AMS, PVA, AUVA vocational rehabilitation has a top priority. Therefore various educational institutions nowadays specialize on occupational retraining in order to facilitate reintegration for different people into the labour market. Amongst these trainings the apprenticeship for computer scientists, focusing on programming instruction, can be found.

A particular target group for this type of education are adults with a rehab background. They are offered a second-chance education and after a 90 weeks training they should be ready for the IT job market. But they not only have to face the usual challenges of learning a programming language. The majority of them have also some kind of physical or psychological history not yet overcome. These additional challenges have to be taken in consideration by the trainer and the trainees throughout the course.

Question arises, what way to teach a programming language is better for this special target group: The procedural or the object-oriented approach first. So these two are examined about their impact on the students.

For both approaches a teaching concept is created and put into practice with two groups for each method. During and after the lessons the reactions of students are evaluated by the trainer. At the end of the course the motivation and the personal impressions of the students are collected in a questionnaire. Based on these findings and other documents such as tests, questionnaires, etc. it will be evaluated to what extent the various teaching approaches are suitable to convey the necessary knowledge.

# Inhaltsverzeichnis

1	Einleitung .....	8
2	Rahmenbedingungen der Ausbildung.....	10
2.1	Ausbildungsweg Informatik laut bmwfw .....	10
2.2	Ausbildungsweg Informatik laut Bildungsinstitut .....	18
2.3	Lehrabschlussprüfung für Informatiker/innen.....	30
3	Lerntheoretische und Didaktische Grundlagen .....	34
3.1	Didaktische Grundlagen in der Informatik.....	34
3.2	Konstruktivismus .....	37
3.3	Verwendete Methoden .....	38
4	Unterrichtskonzept .....	42
4.1	Rahmenbedingungen.....	42
4.2	Kursaufbau.....	44
4.3	Lernergebnisse .....	46
4.4	Methoden .....	52
4.5	Vorkenntnisse .....	52
4.6	Kursvorbereitungen.....	53
4.7	Lehreinheiten .....	54
5	Evaluierung .....	85
5.1	Definitionen .....	85
5.2	Die zehn Empfehlungen .....	86
5.3	Untersuchungsmethoden der Evaluation.....	88
5.4	Praktische Umsetzung .....	90
5.5	Beobachtungsergebnisse Kursverlauf .....	90
5.6	Beobachtung darauffolgender Module.....	106
5.7	Analyse Kursunterlagen und Tests.....	106
5.8	Auswertung des Fragebogens.....	116
5.9	Überprüfung der Hypothese .....	139
6	Schlussfolgerungen.....	143
7	Literaturverzeichnis .....	145
8	Abbildungsverzeichnis.....	147
9	Tabellenverzeichnis.....	148

# 1 Einleitung

Ziel dieser Arbeit ist es herauszufinden, welcher Lehransatz – prozeduraler oder objektorientierter – für das Verständnis der objektorientierten Programmierung für Erwachsene in einem Umschulungsszenario zielführender ist. Im Allgemeinen soll gezeigt werden ob die verschiedenen Lehransätze das Verständnis für die Informatik im Allgemeinen und im Fach Programmieren im Speziellen maßgeblich beeinflussen.

Die Idee hinter dieser Arbeit entstand einerseits durch die persönliche berufliche Erfahrung an einem Bildungsinstitut, welches auf Umschulungen für Erwachsene spezialisiert ist. Zudem behandeln diverse Studien (Brilliant und Wiseman, 1996; Ehlert und Schulte, 2009; Hu, 2004; Vilner, Zur, und Gal-Ezer, 2007) diese Thematik für Schulen und Universitäten, aber nicht für Erwachsene mit Reha Hintergrund. Da diese Zielgruppe aber weiter wächst und Kurse zu Umschulungen in arbeitsmarktrelevante Maßnahmen, wie z.B. Umschulungen zu Informatiker/innen, sehr beliebt sind, wird dieses Feld immer wichtiger für die weitere berufliche Zukunft vieler Erwachsenen.

Im Gegensatz zu Schüler/innen und Student/innen haben diese Erwachsenen meist noch mehr Herausforderungen abseits der Ausbildung zu meistern, wodurch die persönliche Motivation der Teilnehmer/innen oft ausschlaggebend ist um eine positive Entwicklung im Informatiksektor zu ermöglichen. Außerdem ist der große Umstieg vom Berufsleben wieder zurück in ein schulisches System ein weiterer Aspekt, der bei der Ausbildung berücksichtigt werden sollte. Viele Erwachsene sind es nicht mehr gewohnt, den ganzen Tag über in einer Klasse zu sitzen und nur Vorträge zu hören.

Da der Faktor persönliche Motivation nur schwer messbar ist, wird dieser zwar des Öfteren erwähnt, aber ausschlaggebend für die Auswertung der Unterschiede der Lehransätze dieser Arbeit sind die Fakten, die sich durch Tests, Mitarbeitsnoten, Fragebögen und Beobachtungen festhalten lassen. In dieser Arbeit wird das Wort Motivation als Synonym für Einsatz und Interesse am Unterricht gesehen.

Die Hypothese dieser Arbeit lautet: Der prozedurale Lehransatz ist zwar am Anfang leichter zu verstehen, doch der objektorientierte Lehransatz bringt schlussendlich die besseren Ergebnisse in Bezug auf Projektqualität und tieferes Verständnis für objektorientierte Programmierung.

Die Variante mit prozeduralem Lehransatz zuerst bedeutet nicht, dass keine objektorientierte Programmierung unterrichtet wird, sondern bedeutet nur, dass zuerst die Themen der prozeduralen Programmierung und danach die Themen der objektorientierten Programmierung behandelt werden. Dieser unterschiedliche Ansatz wird in der ersten Phase der Ausbildung umgesetzt und beeinflusst dadurch indirekt die weitere Ausbildung.



Die Hypothese wird anhand von Beobachtung und Fragebogen, sowie den erbrachten Leistungsnachweisen untersucht und die Ergebnisse am Ende der Arbeit bewertet und zusammengefasst.

Der erste Teil der Arbeit stellt zuerst die Rahmenbedingungen der Ausbildung vor. In darauf folgenden Kapiteln werden Lerntheoretische und Didaktische Grundlagen der Lehransätze vorgestellt. Im folgenden Kapitel Unterrichtskonzept, wird im Detail erläutert, wie der Kursaufbau und die Lehreinheiten geplant sind. Zum Abschluss hin werden im Kapitel Evaluierung, sowie im Kapitel Schlussfolgerung, die Ergebnisse der Arbeit präsentiert.

## **2 Rahmenbedingungen der Ausbildung**

Die gesamte Ausbildung hat den Lehrabschluss Informatik als Ziel, wobei vom Bundesministerium für Wissenschaft, Forschung und Wirtschaft – im Folgenden mit bmwfw abgekürzt – die Rahmenbedingungen hierfür vorgegeben werden. Die größten Unterschiede zu einer typischen Lehre im Bereich Informatik ergeben sich durch die verkürzte Kurszeit. Laut bmwfw dauert die vorgesehene Lehrzeit ca. 3,5 Jahre, doch im Bildungsinstitut dauert ein Kurszeitraum ca. 90 Wochen, was ca. 1,7 Jahren entspricht. Der zweite Unterschied besteht darin, dass eine Lehre durchgehend einen praktischen Teil in einer Firma hat, und im Bildungsinstitut dieser praktische Teil während des Ausbildungszeitraums mit nur 10 Wochen, in Form eines Praktikums, stattfindet.

Die Verordnung II149/06 (bmwfw, 2006), in welcher das Berufsbild und die Modalitäten zur Lehrabschlussprüfung geregelt sind, gilt für die typische Lehre wie auch für das Bildungsinstitut.

Somit gibt es für das Bildungsinstitut keine Sonderregelung in Bezug auf das Ziel der Ausbildung, doch der Weg dorthin unterscheidet sich sehr stark und wird in den folgenden Kapiteln näher erläutert.

### **2.1 Ausbildungsweg Informatik laut bmwfw**

In der 149. Verordnung des Bundesministers für Wirtschaft und Arbeit über die Berufsausbildung in der Informationstechnologie (Informationstechnologie- Ausbildungsordnung) (bmwfw, 2006) werden zwei Lehrberufe näher erläutert:

- Informationstechnologie – Informatik
- Informationstechnologie – Technik

Da das Ziel dieser Arbeit ein Vergleich der verschiedenen Programmieransätze ist, konzentriert sich diese Diplomarbeit auf die Ausbildung zum/r Informatiker/in.

Im Verlauf dieses Kapitels werden die Unterschiede und Gemeinsamkeiten in der Ausbildung zum/r Informatiker/in zwischen der klassischen Lehre und der Ausbildung am zweiten Bildungsweg aufgedeckt.

#### **2.1.1 Ausbildungszeitraum**

Wie schon in den einführenden Worten erwähnt, unterscheidet sich die klassische Lehre von der Ausbildung am Bildungsinstitut stark hinsichtlich des Ausbildungszeitraums. Laut Bundesgesetzblatt dauert die Lehre ca. 3,5 Jahre. Im Gegenzug dazu dauert die Ausbildung am Bildungsinstitut nur ca. 1,7 Jahre. Diese Abweichung im Bildungszeitraum kann am Bildungsinstitut nur durch eine Reduktion auf die wichtigsten Themengebiete und eine Komprimierung der Praxis wettgemacht werden. Dieser

Unterschied bezüglich des verkürzten Ausbildungszeitraums betrifft im Allgemeinen sehr viele Schulungen in der Erwachsenenbildung, sollte aber keineswegs in Hinblick auf die Kursplanung ignoriert werden. Im Detail kann man diese Unterschiede und Gemeinsamkeiten der Ausbildung in den Kapiteln 2.1 Ausbildungsweg Informatik laut bmwfw und 2.2 Ausbildungsweg Informatik laut Bildungsinstitut nachlesen.

### **2.1.2 Berufsprofil**

Das Berufsbild laut Bundesgesetzblatt (bmwfw, 2006) im Lehrbetrieb und der Berufsschule beschreibt folgende berufliche Tätigkeiten, die fachgerecht, selbstständig und eigenverantwortlich durchzuführen sind, wie folgt:

1. *„Lesen und Anwenden von technischen Unterlagen auch in englischer Sprache,*
2. *Festlegen der Arbeitsschritte, Arbeitsmittel und der Arbeitsmethoden,*
3. *Fachgerechtes Auswählen, Beschaffen, Überprüfen der erforderlichen Betriebsmittel, Materialien und der elektronischen Datenverarbeitungsprogramme,*
4. *Erfassen von technischen Daten über den Arbeitsablauf und die Arbeitsergebnisse,*
5. *Ausführen der Arbeiten unter Berücksichtigung der einschlägigen Qualitäts-, Sicherheits- und Umweltstandards,*
6. *Kundenorientiertes Erstellen von Anforderungsanalysen und Konzepten für Programme und Bedieneroberflächen,*
7. *Erstellen und Testen von Programmen entsprechend den inhaltlichen und wirtschaftlichen Anforderungen,*
8. *Erstellen von Bedieneroberflächen,*
9. *Installieren, Konfigurieren und Prüfen von Datenverarbeitungsprogrammen,*
10. *Analysieren, Eingrenzen und Beheben von Fehlern und Störungen,*
11. *Beraten und Schulen der Anwender,*
12. *Verwalten und Sichern von Daten,*
13. *Erstellen von Dokumentationen und Erfassen von technischen Daten über die Einrichtung der Hardware und der Software für den Kunden,*
14. *Analysieren von Kundenanforderungen und Erarbeiten von Lösungsvorschlägen,*
15. *Anbieten von Service- und Betreuungskonzepten,*
16. *Annehmen, Kontrollieren, Lagern, Pflegen und Ausliefern der Waren, Inventarisieren der Bestände.“*

Die gleichen Fähigkeiten müssen die Auszubildenden des Bildungsinstitutes auch erwerben und stehen deswegen dem Lehrberuf in nichts nach. Der Weg, wie diese Fähigkeiten vermittelt und erlernt werden, unterscheidet sich aber sehr stark. In der Lehre werden diese größtenteils in der Firma erlangt und sind deswegen sehr praxisorientiert. Im Bildungsinstitut hingegen werden diese Fähigkeiten in einem schular-

tigen Umfeld erlangt, wodurch sehr viel Theorie, aber sehr wenig Praxiserfahrung nach der Ausbildung vorhanden ist.

Im Unterrichtsfach Programmieren sind gerade die Punkte

- *1. Lesen und Anwenden von technischen Unterlagen auch in englischer Sprache,*
- *7. Erstellen und Testen von Programmen entsprechend den inhaltlichen und wirtschaftlichen Anforderungen und*
- *10. Analysieren, Eingrenzen und Beheben von Fehlern und Störungen*

sehr stark im Unterricht vertreten. Im Rahmen des Aktiven Kompetenz Trainings (AKT) und des Praktikums wird versucht gerade diese Fähigkeiten so weit wie möglich auch praktisch zu trainieren. Das Aktive Kompetenz Training kann man mit einer Übungsfirma vergleichen, in der die Teilnehmer/innen von einem/r Auftraggeber/in einen Projektauftrag erhalten und diesen im Rahmen ihrer Kompetenzen weitgehend unabhängig realisieren müssen.

### **2.1.3 Berufsbild**

Das Berufsbild laut Bundesgesetzblatt (bmwfw, 2006) sieht folgend aus:

Pos.	1. Lehrjahr	2. Lehrjahr	3. Lehrjahr	4. Lehrjahr
1	Kenntnis über die Aufgaben und den organisatorischen Aufbau des Betriebes sowie über die betrieblichen Arbeitsabläufe			
2	Kenntnis und Anwendung der betrieblichen Einrichtungen und der erforderlichen Hilfsmittel			
3	Kenntnis über Marktstellung und Organisation des Betriebes (Betriebsbereiche) sowie über Warensortiment, betriebsspezifische Angebote und Produkte			
4	Kenntnis der handels- und branchenüblichen Warenbezeichnungen und Fachausdrücke			
5	Fachgerechtes und ergonomisches Vorbereiten des Arbeitsplatzes			
6	Kenntnis über Arbeitsorganisation und Arbeitsplanung sowie Teamarbeit			
7	Kenntnis und Anwendung englischer Fachausdrücke			
8	Lesen und Anwenden technischer Unterlagen auch in englischer Sprache			
9	Einfaches Zerlegen, Warten und Zusammenbauen von mechanischen Teilen			
10	Grundkenntnisse über Aufbau und Arbeitsweise von Mikrocomputersystemen			

11	Grundkenntnisse der Elektrotechnik und Elektronik			
12	Kenntnis und Anwendung der einschlägigen elektrotechnischen Sicherheitsvorschriften und Schutzmaßnahmen			
13	Kenntnis über technische Maßnahmen zur Datensicherheit und zum Datenschutz	Anwenden von technischen Maßnahmen zur Datensicherheit und zum Datenschutz		
14	Grundkenntnisse über die berufsspezifischen rechtlichen Grundlagen wie Datenschutzrecht, Lizenzrecht, Gewährleistungsrecht, Garantie, Versicherungsrecht und Internetrecht			
15	Grundkenntnisse von Betriebssystemen und Bedieneroberflächen	Kenntnis und Nutzung von unterschiedlichen Betriebssystemen und Bedieneroberflächen		
16	Installation von EDV-Software			
17	Grundkenntnisse des Programmierens und Erstellen einfacher Programme			
18		Erstellen von Pflichtenheften in Zusammenarbeit mit dem Kunden		
19		Erstellen und Modifizieren von Software mit professionellen Methoden (inkl. Testkonzepte)		
20		Festlegen von Datenmodellen, Datenstrukturen und Schnittstellen zum Datenaustausch		
21			Kenntnis und Verwendung von Datenbanken	

22		Erstellen von Programmen mit aktuellen Programmiersprachen		
23			Fehlersuche und Fehlerbehebung in Softwareapplikationen	
24		Erstellen von Bedieneroberflächen mit Zugriff auf Datenbanken auf lokalen Rechnern und in Netzwerkkumgebung		
25	Grundkenntnisse von Netzen und Netzwerktechnik sowie der Datenübertragung	Kenntnis von Netzwerken		
26	Kaufmännische Grundkenntnisse (z.B. Kalkulation, Anbot, Lieferung, Rechnung, Produktmarkt, Trends)	Kenntnis der berufsspezifischen kaufmännischen Grundlagen einschließlich des Zahlungsverkehrs		
27	Einschlägige rechnergestützte Schriftverkehrsarbeiten (Arbeiten mit Formularen und Erstellen von Kundendokumentationen)			
28	Lagern, Verwalten, Kontrollieren auf Mängel und Inventarisieren von Waren und Lizenzen			
29	Kenntnis des kundengerechten Verhaltens und der kundengerechten Kommunikation			
30		Kundenorientiertes Verhalten bei der Beratung und Auftragsabwicklung		
31		Produktpräsentation und Marketing		
32		Kenntnis der verkaufsbezogenen rechtlichen Bestimmungen		
33		Anbieten und Durchführen von betrieblichen Serviceleistungen		

34			Erstellen kundenorientierter Konzepte
35		Beratung und Schulung von Kunden und Anwendern	
36		Führen von Verkaufsgesprächen; Bedarf und Wünsche der Kunden ermitteln, Verkaufsargumente ableiten, Fragen und Einwände der Kunden berücksichtigen	
37		Führen von technischen Beratungsgesprächen	
38			Kenntnis über Verhalten bei Reklamationen, Bearbeiten von Reklamationsfällen
39	Grundkenntnisse über Qualitätssicherung und Qualitätskontrolle		
40		Kenntnis und Anwendung des betriebsspezifischen Qualitätsmanagements einschließlich Dokumentation	
41		Mitarbeit bei der Qualitätssicherung	
42		Grundkenntnisse über das Projektmanagement (Projektmethoden, Tools, Projektdefinition, Projektplanung und Projektkontrolle)	
43			Mitarbeit an Projekten (Projektdefinition, Projektplanung und Projektkontrolle)



44	Kenntnis über berufliche Aus- und Weiterbildungsmöglichkeiten
45	Kenntnis der einschlägigen Sicherheitsvorschriften und Schutzmaßnahmen sowie der sonstigen in Betracht kommenden Vorschriften zum Schutze des Lebens und der Gesundheit
46	Kenntnis der Unfallgefahren und von Erste-Hilfe-Maßnahmen
47	Die für den Beruf relevanten Maßnahmen und Vorschriften zum Schutz der Umwelt: Grundkenntnisse über die betrieblichen Maßnahmen zum sinnvollen Energieeinsatz; Grundkenntnisse über die im Arbeitsbereich anfallenden Reststoffe und über deren Trennung, Verwertung sowie über die Entsorgung des Abfalls
48	Kenntnis der sich aus dem Lehrvertrag ergebenden Verpflichtungen (§§ 9 und 10 des Berufsausbildungsgesetzes)
49	Grundkenntnisse der aushangpflichtigen arbeitsrechtlichen Vorschriften

**Tabelle 1: Berufsbild laut (bmwfw, 2006)**

Diese Aufteilung kann man schon rein zeitlich nicht 1:1 auf das Bildungsinstitut übernehmen. Im Kapitel 2.2.5 Bezug auf das Berufsbild wird näher auf die Gemeinsamkeiten und Unterschiede zwischen der Ausbildung am Bildungsinstitut und der Lehre eingegangen.

## **2.2 Ausbildungsweg Informatik laut Bildungsinstitut**

### **2.2.1 Broschüre des Bildungsinstituts**

Die Broschüre („Ausbildung IT-InformatikerIn,“ 2015) des Bildungsinstituts beschreibt, welche Zielgruppe angesprochen werden soll und welche Ziele der Kurs verfolgt. Ein Auszug dieser wird im Folgenden beschrieben.

#### **2.2.1.1 Zielgruppe**

Die Zielgruppe der Ausbildung beschreibt erwachsene Personen, die aufgrund einer Krankheit oder eines Unfalls ihre bisherige berufliche Tätigkeit nicht mehr ausüben können. Dies hat zur Folge, dass diese Menschen aus den unterschiedlichsten Gewerben kommen können und meist längere Zeit von der Berufswelt abgeschnitten waren. Deswegen müssen alle Kursteilnehmer/innen zuerst einen REHA-Plan und/oder eine REHA-Kombination durchlaufen. Dies sind Maßnahmen der Stabilisierung, damit Kursteilnehmer/innen sich hauptsächlich auf die Ausbildung konzentrieren können und Altlasten so weit wie möglich vor der Ausbildung behandelt und abgearbeitet sind.

#### **2.2.1.2 Anforderungen und Ausbildungsinhalte**

Die Anforderungen und Ausbildungsinhalte sollen den Teilnehmer/innen einen Eindruck geben, was sie sich unter der Bezeichnung Informatiker/in vorstellen können.

Folgende Anforderungen laut („Ausbildung IT-InformatikerIn,“ 2015) sind aufgelistet:

- *„Interesse für IT-Themen und die aktuellen Marktentwicklungen*
- *Bereitschaft zum lebenslangen Lernen*
- *analytisches Denken und strukturiertes Arbeiten*
- *einfühlbarer und höflicher Umgang mit KundInnen*
- *Kostenbewusstsein und Flexibilität,*

Gerade das Interesse für aktuelle Marktentwicklungen in der IT und die Bereitschaft zum lebenslangen Lernen, sowie eine Begabung für analytisches Denken und strukturiertes Arbeiten sind wichtige Grundlagen für einen einfacheren Einstieg und die Durchführung der Ausbildung des IT-Informatikers.

Außerdem beschreibt die Broschüre für („Ausbildung IT-InformatikerIn,“ 2015) folgende Ausbildungsinhalte:

- Microsoft Desktop Applikationen
- Programmieren
- Analytik
- Datenorganisation
- Systemtechnik
- Fachenglisch

- Rechnungswesen
- Projektmanagement
- Softskills
- Bewerbungstraining

Für den Vergleich des prozeduralen vs. objektorientierten Programmieransatzes wird hauptsächlich das Unterrichtsfach Programmieren herangezogen. Die Fächer Analytik, Datenorganisation, Projektmanagement, Systemtechnik und Fachenglisch überschneiden sich aber immer wieder mit dem Unterrichtsfach Programmieren, wodurch auch diese Fächer teilweise die Lehrinhalte von Programmieren beeinflussen.

### 2.2.1.3 Sonstiges

Der pädagogische Grundsatz eines modularen Aufbaus der Ausbildungsinhalte ("Ausbildung IT-InformatikerIn," 2015) hat mehrere Gründe:

- **Für das Unternehmen** bringt es den Vorteil, dass es nach einem Modul, falls notwendig, die Vortragenden aus einzelnen Fächern austauschen kann.
- **Für die Teilnehmer/innen** bringt es den Vorteil, dass diese bei einem längeren Ausfall die Ausbildung nicht von Beginn an wieder durchführen müssen, sondern im schlimmsten Fall nur das Modul wiederholen müssen, wo er/sie ausgefallen ist. Somit sparen die Teilnehmer/innen Zeit und können möglichst bald die Ausbildung abschließen und in den beruflichen Alltag zurückkehren.
- **Für den Zahlungsträger** ergibt sich durch den einfachen Einstieg in einzelne Module eine günstigere Variante, seine Klient/innen wieder in die Ausbildung zu integrieren, und damit ergibt sich eine verkürzte Zahlungsdauer als bei einem Totalausfall und einem Wiedereinstieg von Beginn an.

Der genaue Aufbau dieser Module wird in späteren Kapiteln wie zum Beispiel den Kapiteln 2.2.3 Übersicht Modulaufteilung und 2.2.4 Übersicht Studententafel noch näher erläutert.

Aussetzer oder Ausfälle von Teilnehmer/innen kommen immer wieder bei einzelnen Personen vor. Die genauen Gründe sind meist gesundheitlicher Natur und können deswegen in dieser Diplomarbeit nicht näher erläutert werden. Damit sich diese Ausfälle aber in Grenzen halten, gibt es neben der fachlichen Ausbildung – durch meist externe Trainer/innen die aus der Wirtschaft kommen – auch noch interne Betreuung die sich um folgendes kümmert:

- Psychologische Betreuung
- Sozialer Dienst
- Rechtsberatung
- Unterstützung in der Organisation der Ausbildung, sowie
- Kommunikation mit den Zahlungsträgern

Diese Säulen sollen allen Auszubildenden die Möglichkeit geben etwaige Altlasten oder Rückschläge auch während der Ausbildung klären zu können. Selbstverständlich wird hierbei versucht, die fachliche Ausbildung so weit wie möglich nicht zu be-

einflussen. Dies kann aber in der Praxis nicht immer gewährleistet werden. Ausfälle in der Anwesenheit der betroffenen Teilnehmer/innen beeinflussen das Unterrichtstempo für die ganze Gruppe und somit auch den Fortschritt in den einzelnen Fächern. Deswegen muss die zeitliche Planung der Lehreinheiten so weit wie möglich auf solche Verzögerungen flexibel reagieren können. Somit ergibt sich ein allgemeiner Plan für das Fach Programmieren, wo eine Richtung mit den grundlegenden Technologien vorgegeben wird, aber nicht alle Themengebiete zwingend notwendig für die Lehrabschlussprüfung sind und somit teilweise optional von den Trainer/innen zu wählen sind. Weitere Details zu der Planung der Lehreinheiten findet man im Kapitel 4.2 Kursaufbau und folgende.

Nach einer Dauer von 90 Wochen schließen die Teilnehmer/innen im besten Fall mit einer positiven Lehrabschlussprüfung als IT-Informatiker/in ab.

#### **2.2.1.4 Auftraggeber**

Die Auftraggeber sind das AMS (Wien, Niederösterreich, Burgenland), die PV und die AUVA. Diese legen auch die Rahmenbedingungen für den Kurs – wie Kursdauer, Stundenausmaß pro Woche, usw. – fest und bezahlen das Bildungsinstitut, sowie die Auszubildenden während der Ausbildung. Außerdem erhalten die Zahlungsträger nach dem Kurs weiterführende Berichte mit Vermittlungsquoten, die vom Bildungsinstitut den Auftraggebern zur Verfügung gestellt werden.

### **2.2.2 Übersicht Unterrichtsfächer**

Folgende Tabelle zeigt die Abkürzungen der einzelnen Unterrichtsgegenstände der gesamten Ausbildung wie auch eine kurze Beschreibung, was man sich unter dem entsprechenden Fach vorstellen kann. Neben diesen fix geplanten Unterrichtsfächern gibt es noch diverse Seminare, z.B. für Präsentation, Ernährung, Netzwerkcoaching, Teambuilding, usw., die einmalig für Klassen angeboten werden.

<b>Abkürzung</b>	<b>Titel</b>	<b>Kurzbeschreibung</b>
<b>PR</b>	Programmieren	Programmierunterricht in der objektorientierten Programmiersprache C#
<b>DO</b>	Datenorganisation	Im Modul 0 Spezialisierung auf MS Access für den ECDL <sup>1</sup> Standard. Ansonsten erlernen des Umgangs mit relationalen Datenbanksystemen am Beispiel MS SQL Server 2012.
<b>ST</b>	Systemtechnik	Technische Grundlagen der IT in Bezug auf Betriebssysteme und Netzwerke
<b>AKT PR</b>	Anwendungs-Kompetenz-Training Programmieren	Praktische Umsetzung der PR relevanten Themen für das AKT-Projekt, z.B. Erstellung einer Windows-Oberfläche, Anbindung an eine Datenbank, usw.
<b>AKT DO</b>	Anwendungs-Kompetenz-Training Datenorganisation	Praktische Umsetzung der DO relevanten Themen für das AKT-Projekt, z.B. Erstellung der Datenbank, Stored Procedures, usw.
<b>AKT PV</b>	Anwendungs-Kompetenz-Training Prozessverantwortlicher	Praktische Umsetzung der Dokumentation, z.B. Erstellung des Pflichtenhefts, Projekthandbuchs, usw.
<b>AKT ST</b>	Anwendungs-Kompetenz-Training Systemtechnik	Praktische Umsetzung der ST relevanten Themen für das AKT-Projekt, z.B. Erstellung der Benutzer und Benutzergruppen, Netzwerkplan, usw.
<b>AN</b>	Analytik	Erlernen der theoretischen Grundlagen des Programmierens, z.B. Diagrammtypen, Sortieralgorithmen, Suchalgorithmen, usw.
<b>BU</b>	Bewerbungsunterlagen	Erstellen von Bewerbungsunterlagen für zukünftige Arbeitgeber
<b>E</b>	Englisch	Allgemeiner Englischunterricht
<b>EBCL<sup>2</sup></b>	Europäischer Wirtschaftsführerschein	Unterricht zur Erlangung des EBCL, wobei Grundlagen zu Kennzahlen, Bilanzierung, Kostenrechnung, usw. vermittelt werden
<b>FE</b>	Fachenglisch	Englisch für Informatiker
<b>FR</b>	Fachrechnen	Mathematik und Elektrotechnik Grundlagen für die Lehrabschlussprüfung
<b>ITG</b>	IT-Grundlagen	Erlangen der Grundbegriffe der Informationstechnologie, z.B. Wie arbeitet

<sup>1</sup> <http://ecd1.at/>

<sup>2</sup> <http://www.ebcl.at/>

		ein PC, Was ist ein Byte, binäres Rechnen, Aussagenlogik, usw.
<b>KV</b>	Klassenvorstand	Zur Klärung organisatorischer Anliegen wie z.B. Kontrolle der Anwesenheit, Krankenstände, Klärung von klasseninternen Konflikten, usw.
<b>MS</b>	Microsoft	Grundlagen des Betriebssystems Windows 7 als Vorbereitung für den ECDL Base – Computergrundlagen
<b>MSDA</b>	Microsoft Desktop Applikationen	Module für den ECDL Standard, ausgenommen Datenbanken und Computergrundlagen
<b>PM</b>	Projektmanagement	Zur Erlangung des pm basic <sup>3</sup> Zertifikats
<b>PRAV</b>	Praktikumsvorbereitung	Vorbereitungsstunden für das Praktikum, z.B. Bewerbungen schreiben
<b>REBO</b>	Rechnungswesen und Büroorganisation	Grundlagen für das Rechnungswesen und Büroorganisation in Bezug auf die Lehrabschlussprüfung, z.B.: Bestandteile einer Rechnung, usw.
<b>VBFG</b>	Vorbereitung zum Fachgespräch	Wiederholung aller wichtigen Themen für die Lehrabschlussprüfung
<b>WPR</b>	Webprogrammieren	Grundlagen zu clientseitigen Webtechnologien wie HTML, CSS, JavaScript und XML

**Tabelle 2: Abkürzungen Unterrichtsfächer**

<sup>3</sup> <https://www.ocg.at/en/pm-basic>

### 2.2.3 Übersicht Modulaufteilung

Die Ausbildung des IT-Informatikers findet über ca. 90 Wochen statt. In dieser Zeit soll es möglich sein ohne Vorkenntnisse alles Wissen erlangen zu können um die Lehrabschlussprüfung Informatik am Ende des Zeitraumes zu bestehen. Die Ausbildung ist in mehrere Module aufgeteilt, welche in diesem Abschnitt mitsamt ihrer Unterrichtsgegenstände näher erklärt werden.

Die Einführungsphase spielt in der fachlichen Ausbildung so gut wie keine Rolle, da es hierbei um absolute Grundkenntnisse der Informatik geht und der Schwerpunkt eigentlich auf der Vorstellung der Betreuer/innen und das Kennenlernen innerhalb der Gruppe liegt. Diese Phase zählt zwar schon zum Ausbildungszeitraum, hat aber keinen Bezug zum Unterrichtsfach Programmieren.

Das Modul 0 hat als Schwerpunkt den ECDL. Hierbei sollen innerhalb von 10 Wochen der ECDL Standard<sup>4</sup> und der ECDL Advanced<sup>5</sup> Datenbank absolviert werden. Der ECDL ist der Europäische Computerführerschein und soll sicherstellen, dass alle Teilnehmer/innen ab dem Modul 1 die Grundlagen eines Computers und seines Betriebssystems beherrschen.

In den Modulen 1 und 2 werden die Grundlagen der einzelnen Fachgebiete behandelt. Diese sind auch die interessantesten Module für den Vergleich des prozeduralen vs. objektorientierten Programmieransatzes und werden am genauesten in dieser Arbeit behandelt, da nur hier die unterschiedlichen Lehransätze praktiziert werden.

Die Module 1 und 2 bilden somit die Grundlage dieser Arbeit für den Vergleich des prozeduralen vs. objektorientierten Ansatzes. Gerade da es um die Anfänge in der Programmierung geht bilden diese Module die Basis und höhere Module bauen nur auf dieser auf. In den höheren Modulen werden hauptsächlich zusätzliche Programmiertechniken in Bezug auf diverse Projekte unterrichtet und setzen den objektorientierten Ansatz voraus.

Das Modul 3 soll auf den erlangten Grundlagen aufbauen und Platz für erweiterte Techniken in den einzelnen Fachgebieten schaffen. Im Unterrichtsfach Programmieren beschäftigt man sich hierbei mit der Erschaffung von Benutzeroberflächen mittels Microsofts Windows Presentation Foundation (WPF).

Die Module 4 und 5 beschäftigen sich als Kernthema mit dem Aktiven Kompetenz Training (AKT). Hier sollen die theoretischen Grundlagen, die im Unterricht vorgetragen wurden, in einem praktischen Projekt umgesetzt werden. Diese Umsetzung soll soweit es geht sehr selbstständig von den Teilnehmer/innen erfolgen und stellt somit eine Art Übungsfirma dar. Die Lehrkräfte übernehmen hierbei immer mehr die Rolle von Berater/innen und weniger die Rolle von Unterrichtenden.

---

<sup>4</sup> <http://ecdl.at/de/ecdl-standard>

<sup>5</sup> <http://ecdl.at/de/ecdl-advanced>

Das Modul 6 findet bei einem Unternehmen statt und ist mit einem Berufspraktikum gleichzusetzen, wie es z.B. in einer höheren technischen Lehranstalt (HTL) Pflicht ist.

Die Module 7 und 8 haben nur mehr die Lehrabschlussprüfung als Ziel und fast alle Unterrichtsgegenstände fokussieren sich auf diese.

<b>Modul</b>	<b>Dauer in Wochen</b>	<b>Beschreibung</b>
Einführungsphase	2	Unternehmen und seine Gepflogenheiten kennen lernen. Übungen zum Teambuilding und dergleichen. Computer, Maus und Tastatur kennenlernen
Modul 0	10	ECDL Standard, Einführung ins Programmieren
Modul 1	10	Grundlagenwissen Fachausbildung
Modul 2	10	Grundlagenwissen Fachausbildung
Modul 3	10	Vertiefendes Wissen Fachausbildung
Modul 4	10	Start AKT-Phase
Modul 5	10	Abschluss AKT-Phase
Modul 6	10	Praktikum
Modul 7	10	Start Lehrabschlussprüfungsprojekt
Modul 8	8	Abschluss des Lehrabschlussprüfungsprojektes, sowie Lehrabschlussprüfung

**Tabelle 3: Modulaufteilung**

Die Wochendauer kann in der Einführungsphase und gerade im Modul 8 von der Vorgabe leicht abweichen, da es hierbei von den Zeiten abhängt, wann ein Kurs beginnt und wann die nächste Lehrabschlussprüfung stattfindet. In der Regel kommt man aber auf die geplanten 90 Wochen für die Ausbildung zum/r Informatiker/in.

## 2.2.4 Übersicht Stundentafel

In diesem Abschnitt wird eine Übersicht über den Lehrplan gegeben, um einen Gesamteindruck bezüglich der Ausbildung zu vermitteln. Das Unterrichtsfach Programmieren, für die relevanten Module 1 und 2, wird im Kapitel 4.2 Kursaufbau noch genauer beschrieben.

Die Abkürzungen zu den Unterrichtsfächern wurden im Kapitel 2.2.2 Übersicht Unterrichtsfächer aufgelistet und erklärt. Unter der Abkürzung für das Unterrichtsfach steht die Anzahl der Einheiten pro Woche die dieses Fach unterrichtet wird. Die Anzahl der Einheiten kann sich abhängig vom Modul ändern. Eine Einheit entspricht 45 Minuten Unterrichtszeit.

Da sich das Modul 0 sehr stark von den anderen Modulen abhebt, wird dieses hier extra ausgewiesen.



Die Module 1 bis 8 bilden den Kern der Ausbildung, wobei das Modul 6 nicht im Bildungsinstitut stattfindet, sondern den Teilnehmer/innen die Möglichkeit geboten wird, ein Praktikum bei einer Firma zu absolvieren.

Die wichtigen Module 1 und 2 im Unterrichtsfach Programmieren haben jeweils 9 Einheiten pro Woche veranschlagt. Dadurch ergibt sich, dass der Vergleich zwischen prozeduralem und objektorientierten Lehransatz auf insgesamt 180 Einheiten zu je 45 Minuten basiert. Dies ergibt sich aus den zwei Modulen mal zehn Wochen zu je 9 Einheiten pro Woche.

Modul	MS	ITG	PR	FE	E	MSDA	DO	ST	KV
0	2	5	4	2	2	12	9	3	1

Tabelle 4 Grundlagenmodul für Informatiker/innen

M o d u l	P R A D W E F S P M M K AKT AKT AKT AKT R P B F V	A N O P R C L	D O P B C L	W P B C L	E R C L	F E T	S T M	P M	M S D A	K V	AKT S T	AKT P R	AKT P V	AKT D O	R E B O	P R A V	B U	F R	V B F G
1	9	2	9	4	3	2	5	2	3	1									
2	9	2	9	4	3	2	5	2	3	1									
3	9	2	9	4	3	2	5	2	3	1									
4	5		5	4		2				1	5	5	3	5	3	2			
5	5		5	4		2				1	5	5	3	5	3	2			
6	Praktikum																		
7	12		7				5								3		2	6	5
8	12		7				5								3		2	6	5

Tabelle 5: Lehrplan IT-Informatiker/innen

## 2.2.5 Bezug auf das Berufsbild

Folgende tabellarische Abbildung zeigt eine Zuordnung der Positionen, laut Kapitel 2.1.3 Berufsbild, auf die Module und Unterrichtsfächer des Bildungsinstituts sofern diese möglich ist. Hiermit sollen in kompakter Form die Gemeinsamkeiten und Unterschiede zur Vorgabe des bmwfw aufgelistet werden.

Pos	Fach	Modul	Anmerkung
1	Praktikum	6	Mit Hilfe des Praktikums hat ein/e Teilnehmer/in das erste Mal eine Chance den beruflichen Alltag in einem IT-Unternehmen zu erleben. Dieses Praktikum kann nur einen Einblick gewähren und ist nicht so integriert wie eine Lehre bei einem Lehrbetrieb über mehrere Jahre.
2	Praktikum	6	
3	Praktikum	6	
4	REBO	4	In der Vorbereitung zu der Firmengründungsphase des AKT-Projekts werden wichtige Grundlagen diesbezüglich besprochen. In VBFG wird dieses Wissen in Form einer Prüfungssimulation wiederholt und abgefragt.
4	AKT	4, 5	
4	VBFG	7, 8	
5	MSDA	0	Im Rahmen des ECDL werden ergonomische Grundlagen besprochen.
6	AKT Allg.	4, 5	Ziel des AKT-Projekts ist die Förderung von Teamarbeit und Selbstorganisation.
7	E	0	Grundlegende Vokabel und Grammatik.
7	FE	0-5	Ständiger Bezug auf den IT-Sektor.
8	FE	0-5	Häufig in Anlehnung an den Unterrichtsstoff anderer Fächer wie PR, ST, DO, usw.
9	AKT/ST	4	Optional im Rahmen des AKT, sofern Zeit vorhanden ist oder ein Projektbezug besteht.
10	-	-	Wird in der Regel nicht unterrichtet, kann aber bei entsprechendem AKT-Projekt erwähnt werden.
11	VBFG	7, 8	Im Rahmen der Vorbereitung für die Lehrabschlussprüfung werden alle relevanten Grundlagen durchgenommen.
11	FR	7, 8	
12	VBFG	7, 8	Wird im Rahmen eines Fachgesprächs durchgenommen.
13	ITG	0	Grundlagen im Laufe des ECDL.
13	ST	5	Erweiterte Prinzipien in Bezug auf Serverlandschaften.
14	REBO	4	Grundlagen in Bezug auf das AKT-Projekt. Wiederholung und Vertiefung im Laufe der VBFG Einheiten als Vorbereitung für die Lehrabschlussprüfung.
14	VBFG	7, 8	
15	MS	0	Grundlagen Filesystem, Dateimanagement im Rahmen des ECDL.
15	ST	1, 2	Vertiefung der Windows-Grundlagen. Andere Betriebssysteme werden nur am Rande erwähnt.
15	PR	3	Erstellung einfacher Oberflächen mittels WPF des .NET

			Frameworks. Grundlagen im Zusammenspiel Programm und Betriebssystem.
16	ST	1, 2	Im Zusammenhang mit der Erkundung des Betriebssystems, siehe Pos. 15.
17	PR	0-2	Dieser Teil ist der wichtigste der Diplomarbeit. Im Modul 0 wird anhand der Programmiersprache VB.NET der erste Schritt gewagt. Ab Modul 1 werden prozedurale und objektorientierte Ansätze unterrichtet. Die Reihenfolge hängt davon ab ob man in der prozeduralen oder objektorientierten Gruppe ist – siehe Kapitel 4.7 Lehreinheiten.
18	EBCL	3	Wirtschaftliche Grundlagen für ein Pflichtenheft.
18	PR	3	IT-Grundlagen für die Gestaltung eines Pflichtenhefts.
18	REPO	4	Gestaltung eines Pflichtenhefts anhand des AKT-Projekts.
18	AKT/PR	4	
19	PR	3-5, 7, 8	Grundlagen zu Testen sowie praktische Komponententests (Unit-Tests) werden besprochen und programmiert. Außerdem wird sofern die Klasse Kapazitäten hat das Thema Design-Pattern. Die erste aufwendige Applikation die von den Teilnehmern fast vollkommen selbstständig erstellt wird ist das LAP-Projekt.
19	AKT/PR	4, 5	Das erste Projekt, welches als Gruppe erstellt wird und fiktiv einem Kunden präsentiert wird.
20	AN	1-3	Alle theoretischen Grundlagen werden besprochen. Besonders in Bezug auf Such- und Sortieralgorithmen.
20	PR	1, 2, 3	Alle relevanten praktischen Grundlagen und Verhaltensregeln werden im Unterricht vermittelt. Schnittstellen werden nur bei entsprechenden Kapazitäten im Detail besprochen.
20	AKT/PR	4, 5	Teilnehmer/innen setzen die Grundlagen im Rahmen des AKT-Projekts um und werden vom Trainerstab betreut.
21	DO	0-5, 7, 8	Grundlagen bis erweiterte Kenntnisse werden in Bezug auf SQL, T-SQL und dem MS SQL Server 2012 besprochen.
21	AKT/DO	4, 5	Umsetzung in Form des AKT-Projekts.
22	PR	0-5, 7, 8	Im Unterricht werden Programme in Form von Übungsaufgaben selbstständig erstellt. Diese sind meist sehr überschaubar um den zeitlichen Rahmen nicht zu sprengen. Im Modul 7 und 8 findet dies in größerer Form als LAP-Projekt statt.
22	AKT/PR	4, 5	Das erste Gruppenprojekt das von der Planung bis zum Endprodukt von der Klasse zumeist selbstständig erstellt wird.
23	PR	0 - 5, 7, 8	Unterricht und Wiederholung in Bezug auf Fehlersuche und Fehlerbehebung wird in den regulären Unterricht mit

			eingebaut. Vertiefend wird dieses Thema aber erst ab dem 4. Modul behandelt.
23	AKT/PR	5	Während des Projekts wird speziell auf die Fehlerbehandlung eingegangen.
24	PR	4, 5, 7, 8	Ab dem 3. Modul werden nur mehr Applikationen mit einer Windows Oberfläche erstellt. Ab dem 4. Modul werden diese Oberflächen mit Datenbanken in Verbindung gebracht.
24	AKT/PR	4, 5	Ein AKT-Projekt muss immer aus einer Windows Benutzeroberfläche und einer Datenbankanbindung bestehen.
25	ST	3-5	OSI Modell, Netztopologie und dergleichen wird besprochen.
26	EBCL	1-3	Erlangung von wirtschaftlichen Grundlagen.
26	REBO	4, 5, 7, 8	Erweiterungswissen und praktische Übungen zu wirtschaftlichen Themengebieten.
27	AKT/PV	4, 5	Im Rahmen des AKT-Projekts werden Dokumentenvorlagen und Formulare für Rechnungslegung aufgebaut.
28	VBFG	7, 8	Arten von Mängel werden besprochen.
29	EBCL	3	Theoretisches Vorgehen wird unterrichtet.
29	REBO	4, 5	Praktische Erprobung im Rahmen des AKT-Projekts.
30	AKT/PV	4, 5	
31	AKT/PV	5	Endpräsentation im Rahmen des AKT-Projekts vor dem Kunden.
32	REBO	4, 5, 7, 8	Grundlagenwissen in Bezug auf die Lehrabschlussprüfung.
33	-	-	Kann nur im Rahmen des Praktikums stattfinden – ist aber unwahrscheinlich.
34	AKT Allg.	4, 5	Im Rahmen des AKT-Projekts, sofern es das Produkt zulässt.
35	AKT Allg.	5	
35	PR	7, 8	Im Rahmen des LAP-Projekts, sofern die Prüfungskommission daran interessiert ist.
36	AKT Allg.	4, 5	Im Rahmen des AKT-Projekts.
37	AKT Allg.	4, 5	Nur ansatzweise im Laufe des AKT-Projekts.
38	VBFG	7, 8	Theoretisches Wissen in Form eines Fachgespräches als Vorbereitung für die Lehrabschlussprüfung.
39	-	-	Wird nur sehr rudimentär während des Unterrichts behandelt.
40	-	-	Nicht möglich.
41	-	-	Nicht möglich
42	PM	1-3	Im Rahmen der Zertifizierung.
43	AKT Allg.	4, 5	Durch das AKT-Projekt soll jede/r Teilnehmer/in die Möglichkeit bekommen ein Softwareprojekt vom Anfang bis zum Ende zu begleiten. Projektmanagement Skills werden hier praktisch erprobt.

<b>44</b>	BU	7, 8	Beratungsgespräche und weiterführende Betreuung.
<b>45</b>	VBFG	7, 8	Grundkenntnisse für das Lehrabschlussgespräch.
<b>46</b>	VBFG	7, 8	
<b>47</b>	ITG	0	Grundlagen im Rahmen des ECDL.
<b>48</b>	-	-	Keine Notwendigkeit in Bezug auf die Lehre. Ähnlich in der Form der Pflichten in Bezug auf den Zahlungsträger und die Ausbildungsstätte.
<b>49</b>	-	-	Wird nicht behandelt.

**Tabelle 6: Zuordnung Bildungsinstitut und Berufsbild**

Anhand dieser Tabelle kann man erkennen, welche Positionen überhaupt am Bildungsinstitut umgesetzt werden und in welche Fächer und Zeiträume diese aufgeteilt werden.

Wie auch schon im Kapitel 2.1.2 Berufsprofil erwähnt, unterscheidet sich die Ausbildung am Bildungsinstitut von der Lehre hauptsächlich in Bezug darauf, dass diese einen eher schulischen und wenig praxisorientierten Ansatz hat. Bestimmte Kenntnisse, die besonders auf einen Betrieb zugeschnitten sind, können auch gar nicht vom Bildungsinstitut behandelt werden.

Auch hier sieht man, dass laut Bundesgesetzblatt ca. ein Fünftel aller Positionen relevant für das Unterrichtsfach Programmieren sind. Genauer gesagt sind es die Positionen 15, 17, 18, 19, 20, 22, 23, 24 und teilweise 35. Von diesen Positionen sind die Positionen 17, 20 und 22 für den Vergleich der verschiedenen Lehransätze relevant. Obwohl es nur wenige Positionen sind, kann man in der Studentafel von Kapitel 2.2.2 Übersicht Unterrichtsfächer sehr gut erkennen, dass das Fach Programmieren neben Datenorganisation eines der Hauptfächer ist und deswegen sehr viel Zeit zugeordnet bekommt.

## **2.3 Lehrabschlussprüfung für Informatiker/innen**

Laut § 5 der 149. Verordnung für Informationstechnologie-Ausbildungsverordnung (bmwfw, 2006) gliedert sich die Lehrabschlussprüfung in eine praktische und in eine theoretische Prüfung.

Die praktische Prüfung umfasst hierbei eine Prüfarbeit, sowie ein dazugehöriges Fachgespräch. Die Prüfarbeit beschreibt ein praktisches Projekt das von allen Lehrlingen umgesetzt werden muss. In der Regel umfasst dieses Projekt eine Webanwendung mit einer Datenbankanbindung. Die theoretische Prüfung umfasst die Gegenstände Angewandte Physik, Angewandte Mathematik und Informatik.

Laut Punkt 4 des § 5 kann die theoretische Prüfung bei Erreichen des Lehrzieles der letzten Klasse der Berufsschule auch entfallen.

Da das Bildungsinstitut der Berufsschule nicht gleichgesetzt ist, entfällt der Punkt 4 für die Auszubildenden, und somit müssen diese auf jeden Fall die praktische und theoretische Prüfung absolvieren.

### **2.3.1 Praktische Prüfung**

Laut § 6 der 149. Verordnung für Informationstechnologie-Ausbildungsverordnung (bmfw, 2006) kann die praktische Prüfung auf zwei unterschiedliche Arten absolviert werden. Diese Varianten werden im Laufe dieser Diplomarbeit nun kurz vorgestellt, sind aber in den §§ 7 und 8 der Ausbildungsverordnung genau beschrieben.

Die erste Variante laut § 7 umfasst einen Arbeitsauftrag, der entweder eine Bedieneroberfläche oder eine Datenbank anpasst. Die Aufgabenstellung selbst legt hierbei die Prüfungskommission fest. Die Prüfung darf acht Stunden nicht überdauern, und die Bewertungskriterien umfassen die fachgerechte Arbeitsweise, richtige und zweckentsprechende Funktion, anwenderfreundliche Konfiguration und fachgerechtes Verwenden von Hilfsmittel.

Die zweite Variante laut § 8 beschreibt ein betriebliches Projekt, welches einen betrieblichen Auftrag umfasst. Dieses wird nach Genehmigung der Lehrlingsstelle von den Lehrlingen fachbezogen dokumentiert und am Ende vor einer Prüfungskommission präsentiert. Die fachbezogene Präsentation soll 15 Minuten dauern, und prozessrelevante Qualifikationen des Prüflings sollten dabei herausstechen. Für die Bewertung werden fachgerechte Arbeitsweise, richtige und zweckentsprechende Funktion, anwenderfreundliche Konfiguration, fachgerechtes Verwenden von Hilfsmittel, sowie praxisbezogene Dokumentation und Präsentation herangezogen.

Neben dieser Prüfarbeit die laut § 7 oder § 8 absolviert werden muss, gibt es noch das Fachgespräch. Dieses ist laut § 9 vor der gesamten Prüfungskommission abzuhalten. Laut § 9 Punkt 2 ergibt sich das Gespräch aus dem praktischen Teil, und es wird das Wissen aus dem praktischen und kaufmännischen Bereich, sowie die Fähigkeit ein Kundengespräch zu führen, überprüft. Laut § 9 Punkt 3 soll sich dieses als lebendiges Gespräch entwickeln und somit Schilderungen von Situationen und Problemen beinhalten, um auch weitere Themengebiete wie z.B. Unfallverhütung abzudecken.

Das Fachgespräch an sich soll 15 Minuten dauern, wobei im Zweifelsfall dieses um zehn Minuten verlängert werden darf.

### **2.3.2 Theoretische Prüfung**

Laut § 10 Punkt 1 der 149. Verordnung für Informationstechnologie-Ausbildungsverordnung (bmfw, 2006) hat die theoretische Prüfung schriftlich zu erfolgen. Die theoretische Prüfung ist auch vor der praktischen Prüfung abzuhalten und die Aufgaben sollen den Anforderungen der Berufspraxis entsprechen.

Die Aufgabenstellungen der Gegenstände Angewandte Physik, Angewandte Mathematik und Informatik sollen innerhalb von 60 Minuten durchgeführt werden können, wobei die Prüfung nach 80 Minuten beendet wird. Jeder Gegenstand hat Bereiche aus denen je eine Aufgabe ausgewählt werden soll. Nähere Beschreibungen entnimmt man den §§ 11, 12 und 13 (bmfw, 2006).

Die §§ 14 – 17 (bmwfw, 2006) beschreiben Wiederholungsprüfung, eingeschränkte Zusatzprüfung, Übergangsbestimmungen und Schlussbestimmungen, die für diese Arbeit nicht relevant sind und deswegen hier nicht weiter erläutert werden.

### **2.3.3 Anwendung am Bildungsinstitut**

Da die Variante laut § 8 (bmwfw, 2006) der praktischen Prüfung aus dem Kapitel 2.3.1 Praktische Prüfung auf das Bildungsinstitut nicht anwendbar ist, bleibt für das Bildungsinstitut nur mehr die praktische Prüfung laut § 7 (bmwfw, 2006).

Folgend wird die Umsetzung dieses Prüfungsszenarios für das Bildungsinstitut näher beschrieben.

#### **2.3.3.1 Vorbereitungsphase**

Wie schon im Kapitel 2.2.3 Übersicht Modulaufteilung kurz erwähnt, dienen die Module 7 und 8 nur der Vorbereitung der Lehrabschlussprüfung.

Die Fächer Vorbereitung auf das Fachgespräch (VBFG) und Fachrechnen (FR) beschäftigen sich ausschließlich mit Themen, die die Lehrabschlussprüfung betreffen. Gerade in Hinblick auf die theoretische Prüfung ist FR ein essenzielles Fach, um den Teilnehmer/innen wieder die Grundzüge der Angewandten Mathematik und Angewandten Physik ins Gedächtnis zu rufen.

In den Fächern Programmieren (PR) und Datenorganisation (DO) wird im Laufe dieser Module ein Projekt entwickelt, welches dann später für die Lehrabschlussprüfung als Grundlage dient. Am Bildungsinstitut ist dieses ein Webprojekt mit Datenbankbindung. Jede/r Teilnehmer/in bekommt hierfür im Modul 7 ein Pflichtenheft, welches die Rahmenbedingungen für das Projekt abstecken.

Bis zum Termin der Lehrabschlussprüfung ist es das Ziel aller Teilnehmer/innen, dieses Projekt so eigenständig wie möglich, aber im Unterricht mit Unterstützung der Trainer/innen, umzusetzen. Das Pflichtenheft wird auch an die Prüfungskommission geschickt, damit diese einen Einblick in das Projekt bekommt und daraus Prüfungsfragen ableiten kann.

#### **2.3.3.2 Prüfungsphase – Tag 1: theoretische Prüfung**

Die theoretische Prüfung findet außerhalb des Bildungsinstituts bei der Wirtschaftskammer statt. Hierbei werden die Gegenstände Angewandte Mathematik, Angewandte Physik und Informatik unter Aufsicht eines Prüfungsorgans in drei separaten Prüfungsbögen abgearbeitet.

#### **2.3.3.3 Prüfungsphase – Tag 2: praktische Prüfung Projektarbeit**

Das erstellte Projekt, wie in Kapitel 2.3.3.1 Vorbereitungsphase beschrieben, wird als Grundlage für die Aufgabenstellungen der praktischen Prüfung herangezogen. Hierbei werden von der Prüfungskommission mehrere von ihr gewählte Aufgaben für die Erweiterung oder Änderung des Projekts definiert. Somit bildet das Projekt den Einstiegspunkt für die praktische Prüfung. Aus dem Thema des Projekts und den Aufga-



benstellungen der Prüfungskommission ergibt sich dann auch schlussendlich der Rahmen für das Fachgespräch.

#### **2.3.3.4 Prüfungsphase – Tag 3: praktische Prüfung Fachgespräch**

In der letzten Phase der Lehrabschlussprüfung findet ein Fachgespräch statt, welches sich zumeist auf das praktische Projekt bezieht, aber manchmal auch Schwächen der theoretischen Prüfung aufgreift – sofern es die Thematik zulässt. Hier wird von der Prüfungskommission versucht, in einem Gespräch unter Fachkundigen die Schwächen und Stärken der Teilnehmer/innen zu erfassen, um diese gerecht beurteilen zu können.

### 3 Lerntheoretische und Didaktische Grundlagen

Es gibt verschiedene Methoden zu lernen – indem man zuschaut und/oder zuhört, selbst ausprobiert, über getane Arbeiten reflektiert oder sich einfach nur erinnert, wie man schon zuvor ein ähnliches Problem gelöst hat und diesen Lösungsweg adaptiert.

So wie es verschiedene Methoden gibt um zu lernen, so gibt es auch verschiedene Methoden um zu lehren. Manche Lehrenden halten Vorträge, andere wiederum starten Diskussionen oder führen Demonstrationen vor. Jede Methode hat seine Vor- und Nachteile und ist somit besser für bestimmte Szenarien geeigneter als andere.

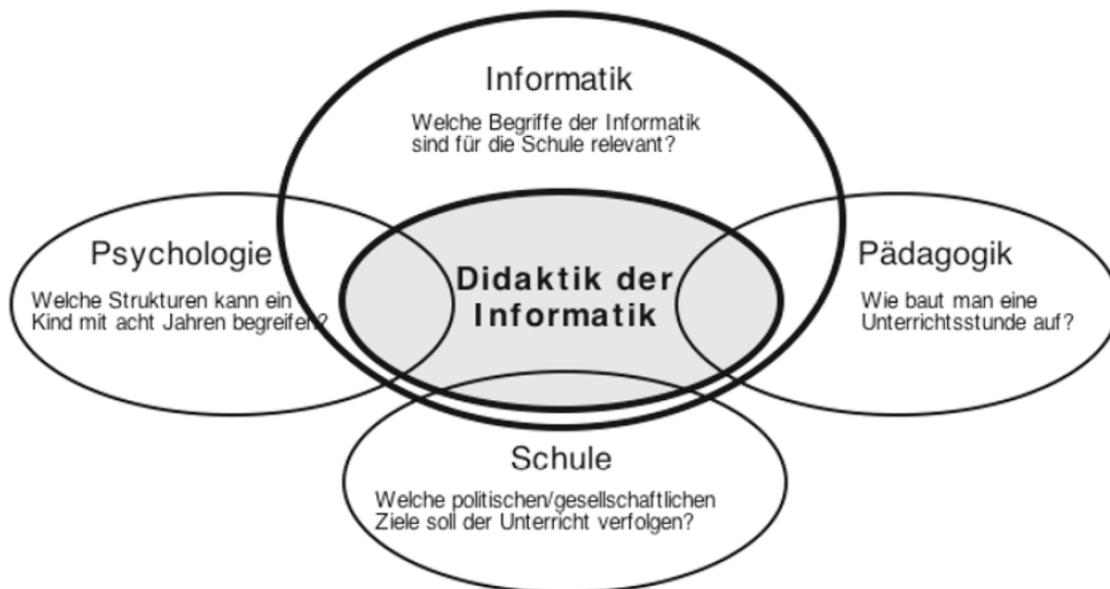
Ob die Lernmethode mit der Lehrmethode für die einzelnen Teilnehmer/innen im Einklang stehen, kann nur in der Praxis ausprobiert und analysiert werden. Dass das Zusammenspiel dieser Komponenten auch ausschlaggebend für den Lernerfolg ist, wurde schon von (Felder und Silverman, 1988) folgend beschrieben: *„How much a given student learns in a class is governed in part by that student’s native ability and prior preparation but also by the compatibility of his or her learning style and the instructor’s teaching style.“*

Um diesen Faktor der Kompatibilität zwischen Lerntyp und Unterrichtsstil geht es in diesem Kapitel, wo die verschiedenen didaktischen Grundlagen, die für den Kurs verwendet werden, näher gebracht werden.

#### 3.1 Didaktische Grundlagen in der Informatik

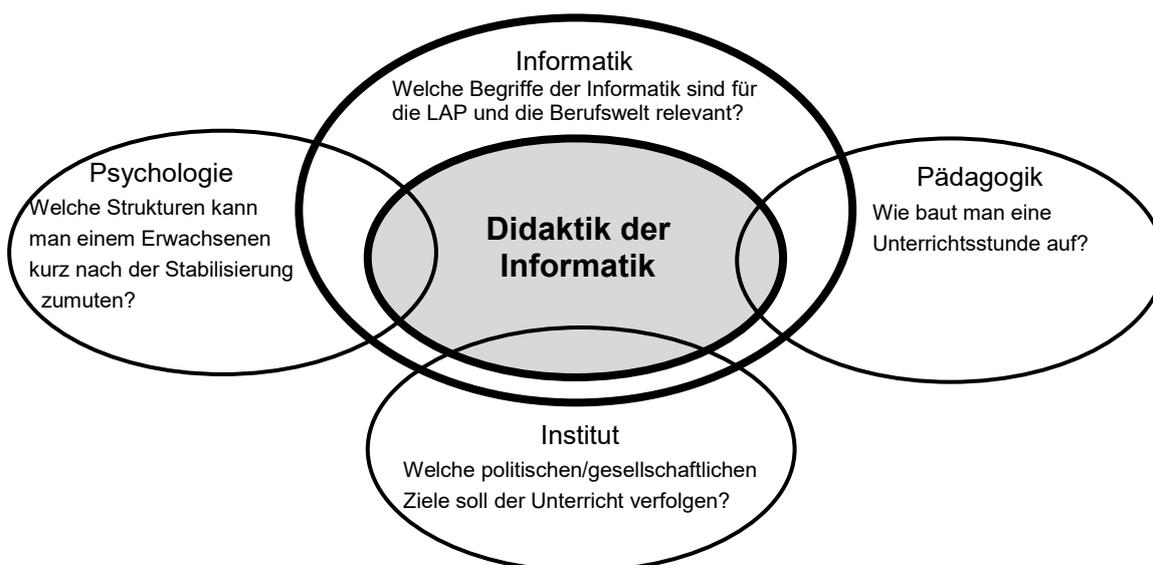
Laut (‘‘Meyers enzyklopädisches Lexikon,’’ 1974) ist Didaktik folgend definiert: *„Unterrichtslehre; Kunst des Lehrens; Wissenschaft von der Methode des Unterrichtens“*. Daraus ergibt sich, dass sich Didaktik damit beschäftigt, wie man am besten Wissen vermittelt und welche Methode hierfür die effektivste ist.

In Bezug auf die Informatik steht nicht die Informatik alleine im Mittelpunkt (Schubert und Schwill, 2011). Mehrere Wissenschaften und Institutionen beeinflussen die Didaktik in der Informatik. (Schubert und Schwill, 2011) hat es in folgender Grafik für Schulen veranschaulicht.



**Abbildung 1: Einbettung der Didaktik laut (Schubert und Schwill, 2011)**

Bildet man diese Grafik auf das Bildungsinstitut ab, so sieht dieses in Bezug auf das Bildungsinstitut für die Diplomarbeit folgend aus:



**Abbildung 2: Einbettung der Didaktik am Bildungsinstitut**

In dieser, für die Diplomarbeit angepassten, Abbildung erkennt man, dass die wesentlichen Unterschiede von der Fragestellung her in der Psychologie und in der Informatik zu finden sind. Die politischen und gesellschaftlichen Ziele sind vielleicht nicht anhand der Fragestellung unterschiedlich, aber alleine die Tatsache, dass das Bildungsinstitut auch gewinnorientiert denken muss lässt hier Unterschiede vermuten. Folgend wird kurz auf jede dieser Komponenten, die die Didaktik in der Informatik ausmachen eingegangen.

- Pädagogik  
Hierbei geht es um den zielführendsten Aufbau einer Unterrichtsstunde am

Bildungsinstitut. Dieser Teil wird im Kapitel 4.7 Lehreinheiten für die einzelnen Kurse beschrieben.

- **Institut**  
Das Ziel des Bildungsinstituts und deren Auftraggeber ist es, die Teilnehmer/innen zum Abschluss einer Ausbildung in Form des Lehrabschlusses für Informatiker/innen laut Kapitel 2.1 Ausbildungsweg Informatik laut bmwfw zu bewegen, um den Teilnehmer/innen langfristig eine Chance auf den Arbeitsmarkt zu bieten. Außerdem muss das Bildungsinstitut zumindest auf eine kostendeckende Ausbildung achten. Die Auftraggeber haben das Ziel, dass so vielen Menschen wie möglich eine derartige Ausbildung, bei vorhandenem Interesse, geboten werden kann, aber die Qualität nicht darunter leidet.
- **Psychologie**  
Da alle Teilnehmer/innen schon am Arbeitsmarkt aktiv waren und diesen wegen eines unvorhergesehenen Einwirkens meist schlagartig für einen längeren Zeitraum verlassen mussten und die meisten auch eine Stabilisierungsmaßnahme durchlaufen mussten, sind die psychologischen Rahmenbedingungen von hieraus schon einmal anders als bei Schüler/innen. Außerdem bewirkt die Tatsache, dass es sich um Erwachsene handelt, die auch schon soziale Verantwortungen, wie z.B. für Familie und Kinder, außerhalb der Ausbildung tragen, andere Rahmenbedingungen für den Unterricht.
- **Informatik**  
Durch den verkürzten Ausbildungszeitraum und den Druck, so schnell wie möglich wieder in die Berufswelt einsteigen zu können, bildet sich in der Informatik der Schwerpunkt einer sehr berufsorientierten Ausbildung für die Teilnehmer/innen. Eine Herausforderung hierbei ist auch den besten Weg zwischen Lehrabschlussprüfung, die sehr schulisch ist, und beruflicher Praxis zu finden.

Dieser berufsorientierte Ansatz und die Tatsache, dass es sich um Menschen aus der Arbeitswelt, mit Reha-Hintergrund handelt, bestimmt die Didaktik in der Informatik in Bezug auf das Bildungsinstitut sehr stark.

Dieser praktische Ansatz deckt sich aber sehr gut mit (Schubert und Schwill, 2011), wo ein besonderer Fokus auf die Zusammensetzung der Inhalte und auf prägnante Beispiele besonderer Wert gelegt wird: *„Denn die Auswahl und Zusammensetzung von geeigneten Inhalten ist von zentraler Bedeutung für die Vermittlung von Begriffen und Prinzipien der Informatik. Besonderes Gewicht ist auf prägnante Beispiele zu legen: Sie bleiben einerseits lange haften, und sie komprimieren Information, d.h., selbst komplizierte Sachverhalte lassen sich meist außerordentlich leicht wieder über die Beispiele erschließen.“*

Mit diesem sehr praktischen Ansatz kann für die Teilnehmer/innen vielleicht die Rückkehr zu einem schulischen System, welches in manchen Fällen schon Jahrzehnte zurückliegt, erleichtert werden. Dieser Ansatz, dass man mit wenigen prägnanten Beispielen viel bewirken kann, deckt sich sehr gut mit dem begrenzten Zeitrahmen. Die größte Herausforderung hierbei ist aber, dass die Ausarbeitung der Beispiele einen nicht zu unterschätzenden Mehraufwand bei der Erstellung Planung der

Kurse bedeutet. Qualitativ hochwertige Unterrichtsbeispiele benötigen viel Erfahrung und müssen laufend an die äußeren Umstände, wie z.B. Gruppengröße und Themenschwerpunkte, angepasst werden.

## 3.2 Konstruktivismus

Dieser praktische Ansatz der Didaktik der Informatik, der im Kapitel 3.1 Didaktische Grundlagen in der Informatik eingeführt wurde, deckt sich sehr gut mit dem Konstruktivismus, bei dem die Prozessmerkmale des Lernens laut (Hubwieser, 2007) auszugswise folgend beschrieben werden *„Lernen ist nur über die **aktive** Beteiligung des Lernenden möglich. Dazu gehört, dass der Lernende zum Lernen **motiviert** ist und dass er an dem, was er tut und wie er es tut, **Interesse** hat oder entwickelt.“*

Die einzelnen Schlagwörter – aktiv, motiviert und Interesse – sind die Grundlage, um eine erfolgreiche Ausbildung und späteren beruflichen Erfolg in der Informatik zu ermöglichen. Das Interesse wird auch schon von der Broschüre des Bildungsinstituts – siehe Kapitel 2.2.1.2 Anforderungen und Ausbildungsinhalte – erwartet. Die aktive Beteiligung am Unterricht wird im Laufe dieser Diplomarbeit durch die unterrichtenden Trainer/innen mit Hilfe der Unterrichtsnotiz – siehe Kapitel 5.5.1 – erfasst und im Kapitel 5 Evaluierung näher behandelt. Die schwer messbare Motivation wird anhand des Fragebogens, im Kapitel 5.8 Auswertung des Fragebogens, und der Unterrichtsbeobachtung erfasst.

Die Informatik ist ein sehr lebendiges Gebiet, wo es ohne eigene Motivation und Interesse nicht funktioniert, am letzten Stand des Wissens zu bleiben. Das erfordert eine sehr aktive Rolle von Informatiker/innen, sich selbstständig fortlaufend weiterzubilden, um für den Arbeitsmarkt attraktiv zu bleiben.

Diese Eigenschaften finden auch während der Ausbildung eine starke Gewichtung. Die Rahmenbedingung der kurzen Ausbildungszeit setzt es schon voraus, dass sich die Teilnehmer/innen auch in ihrer Freizeit über den Unterrichtsstoff hinaus mit der Informatik beschäftigen und sich auch während des Unterrichts so weit wie möglich beteiligen, um in dieser kurzen Zeit den Input bestmöglich verarbeiten zu können. Gerade die aktive Beteiligung ist in den Modulen 4 und 5 im Rahmen des AKT-Projekts sehr wichtig, damit die Teilnehmer/innen in dieser Zeit ihr Grundwissen stärken und ausbauen können. Eine Nicht-Beteiligung am AKT-Projekt kann gerade hier den Teilnehmern/innen sehr schaden, da dies die erste Gelegenheit ist, das theoretische Wissen in einer simulierten Umgebung selbstständig anzuwenden.

Neben den anderen Prozessmerkmalen von (Hubwieser, 2007), die hier nicht näher behandelt werden, gibt es aber noch ein Prozessmerkmal, das gerade in Hinblick auf die Zielgruppe des Kurses sehr wichtig ist: *„Lernen ist schließlich immer auch ein sozialer Prozess: Zum einen sind der Lernende und all seine Aktivitäten stets soziokulturellen Einflüssen ausgesetzt, zum anderen ist jedes Lernen ein interaktives Geschehen.“* (Hubwieser, 2007).

Dies trifft gerade auf die spezielle Konstellation von Erwachsenen zu, die teilweise soziale Defizite aufweisen und somit manchmal schwer zu sozialen Kontakten, geschweige denn produktiven Partner- und Gruppenarbeiten zu bewegen sind. Wie auch die Tatsache, dass diese Erwachsenen oft noch ganz anderen äußeren Ein-

flüssen verstärkt ausgesetzt sind, wie die wären: Finanzielle Probleme, psychologische Betreuung, Medikamente, Ersatzdrogen, familiäre Probleme und dergleichen. Diese Einflüsse hindern die Teilnehmer/innen oft daran, sich auf das Lernen zu konzentrieren und machen dadurch einen Lernfortschritt fast unmöglich. Dadurch kann es leicht zu Demotivation führen – z.B. wenn man mit dem Lernen nicht mehr nachkommt – und somit zu einer Abwärtsspirale führen. Die Kurseinheiten müssen dementsprechend angepasst werden, um dieser Abwärtsspirale, soweit es im Rahmen des Unterrichts möglich ist, die Kraft zu nehmen. Diese dynamische Anpassung kann aber nur in einem gewissen Rahmen stattfinden, da das Ziel der Lehrabschlussprüfung weiterhin oberste Priorität hat. Zumeist wird der Druck von der Gruppe genommen, indem man zusätzliche Themen aus den Unterrichtsfächern reduziert oder streicht, damit man sich auf die Basis konzentrieren kann. Beispiele hierfür kann man im Kapitel 4.7 Lehreinheiten entdecken.

### **3.3 Verwendete Methoden**

Die hier kurz vorgestellten Methoden stammen aus dem Buch Konstruktivistische Didaktik (Reich, 2012) und bilden in den Kursen die Grundlage zur Wissensvermittlung. Die Auswahl wurde anhand des Methodenpool<sup>6</sup> (Reich, 2007), der auch online zur Verfügung steht, getroffen.

#### **3.3.1 Frontalunterricht und Präsentationen**

Eine der größten Herausforderungen ist es, einen Frontalunterricht im Sinne des Konstruktivismus zu halten, da auf den ersten Blick und bei falscher Anwendung diese Methode eher zu einer Demotivation, anstatt einer Motivation zu der aktiven Beteiligung führt. Trotzdem wird diese Methode in kurzen, aber häufig wiederkehrenden Fällen angewendet, um Wissen auf eine sehr direkte Art zu vermitteln. Diese Wissensvermittlung muss aber sehr gut geplant sein, damit auch Raum für offene Phasen vorhanden bleibt. Im Methodenpool empfiehlt (Reich, 2007), dass

- Frontalunterricht zeitlich stark begrenzt – gerade bei abstrakten Themen – ist und
- offen für Kommunikation bleibt und das persönliche Interesse geweckt wird, z.B. Fragen stellen und zulassen
- Diese Phasen mit anderen Methoden abgestimmt sind, wie z.B. Einzel- oder Gruppenarbeiten

#### **3.3.2 Fragend-entwickelnde Methode**

Ähnlich dem Frontalunterricht ist der fragend-entwickelnde Unterricht sehr Lehrer/innen fokussiert. Deshalb wird auch hier den Vorgaben des Frontalunterrichts gefolgt, und die Phasen werden eher kurz gehalten. Im Laufe der Kurse wird diese Me-

---

<sup>6</sup> Reich, K. (Hg.): Methodenpool. In: url: <http://methodenpool.uni-koeln.de>

thode immer wieder mit dem Frontalunterricht abgewechselt und gerade für Wiederholungen des Unterrichtsstoffes wird diese Methode sehr gerne eingesetzt, da diese den Lehrenden ein Feedback über den allgemeinen Wissensstand der Teilnehmer/innen auf sehr schnelle Weise liefern kann.

### **3.3.3 Einzelarbeit**

Diese dient in den Kursen als Zeit des Übens. Hierbei haben die Teilnehmer/innen die Möglichkeit, Übungsbeispiele auf ihre eigene Weise zu lösen und hierfür den Unterrichtsstoff zu reflektieren. Die Lehrenden haben wiederum die Möglichkeit Teilnehmer/innen die besondere Aufmerksamkeit verlangen, Raum dafür zu geben, sofern sie es für nötig erachten.

### **3.3.4 Gruppenarbeit**

Die Gruppenarbeit verlangt nach (Reich, 2007) vier Phasen, die geplant werden müssen:

1. *„Vorbereitungsphase*
2. *Durchführungsphase*
3. *Präsentation/Auswertung*
4. *Feedback/Evaluation“*

Diese Methode wird meist verstärkt erst ab dem dritten Modul für die Teilnehmer/innen angeboten, da die Rahmenbedingungen vom Bildungsinstitut diese nicht immer zulassen. Einerseits stehen manchmal keine Räumlichkeiten für eine erfolgreiche Gruppenarbeit zur Verfügung und andererseits gibt es manchmal so große soziale Spannungen innerhalb einer Klasse, dass Gruppenarbeiten weniger der Wissensvermittlung, sondern mehr der Klärung sozialer Probleme dienen.

Deswegen wird diese Methode meist nur in höheren Modulen verwendet und häufig nur für optionale Themengebiete, die nicht zwingend für den Lehrabschluss sind. Ein Beispiel hierfür wäre die Auseinandersetzung mit verschiedenen Setup-Routinen oder verschiedenen Dokumentationstools. Trotzdem versuchen die Lehrenden immer wieder Gruppenarbeiten zu forcieren, da diese besonders in der Wirtschaft häufig vorkommen und soziale Kompetenzen durch Training gestärkt werden können.

### **3.3.5 Projektarbeit**

Die Projektarbeit in der Form, wie es der Methodenpool (Reich, 2007) folgend beschreibt: *„Projektarbeit ist das selbstständige Bearbeiten einer Aufgabe oder eines Problems durch eine Gruppe von der Planung über die Durchführung bis zur Präsentation des Ergebnisses.“* ist die treffende Beschreibung für das AKT-Projekt, welches in den Modulen 4 und 5 stattfindet. Hierbei wird großer Wert auf *„handelnd-lernend“*

gelegt, damit die Teilnehmer/innen die Möglichkeit haben, auch selbst ihre Grenzen entdecken und sprengen zu können.

Hierbei haben die Lehrenden eine moderierend/beratende Funktion und unterstützen in dieser Funktion die Klasse bei der Realisierung des AKT-Projekts.

In ersten Ansätzen und gerade bei sehr motivierten Klassen entwickelt sich die Projektarbeit in Richtung einer Juniorfirma laut (Reich, 2007), da hierbei auch unternehmerisches Denken und Planen wie z.B. Personalwesen, Produktion,... fächerübergreifend behandelt wird.

### **3.3.6 Blitzlicht**

Die Blitzlicht-Methode wird im Laufe des Kurses in einer ähnlichen Form als Daily Scrum Meeting im Rahmen des AKT abgehalten. Das Daily Scrum Meeting laut (Schwaber und Beedle, 2008) und dem Scrum Guide (Schwaber und Sutherland, 2013) beinhaltet drei Fragen:

1. „Was habe ich gestern erreicht, das dem Entwicklungsteam hilft, das Sprint-Ziel zu erreichen?“
2. Was werde ich heute erledigen, um dem Entwicklungsteam bei der Erreichung des Sprint-Ziels zu helfen?
3. Sehe ich irgendwelche Hindernisse, die mich oder das Entwicklungsteam vom Erreichen des Ziels abhalten?“

Diese Fragen liefern wie die Blitzlicht-Methode „... *Feedback, die schnell die Stimmung, Meinung, den Stand bezüglich der Inhalte und Beziehungen in einer Gruppe ermitteln kann.*“ (Reich, 2007)

Rein formell sind beide Methoden nicht deckungsgleich, doch am Bildungsinstitut wird das Daily Scrum Meeting den äußeren Umständen entsprechend angepasst und es dürfen auch Gefühle, wie es auch in der Blitzlichtmethode erlaubt ist, vermittelt werden.

Außerdem wird die Blitzlichtmethode sofort eingesetzt, sobald die Trainer/innen das Gefühl haben, dass die Klasse nicht ganz auf Kurs ist. Damit soll so bald wie möglich auf etwaige negative Gruppendynamiken reagiert werden und Probleme in Bezug auf die Planung der Lehreinheiten entdeckt werden.

Diese Methode wird auch mit der Fragend-entwickelnden Methode gemeinsam in den Lehreinheiten verwendet um den Wissensstand zu prüfen und findet somit zumindest einmal pro Modul statt.



### **3.3.7 Brainstorming**

Wird im Kurs hauptsächlich in der Vorbereitungsphase zu Gruppenarbeiten angewendet, sowie in den Planungsphasen für das AKT- und das LAP-Projekt.

Brainstorming ist eine Methode, um einen schnellen und sehr aktiven Einstieg in ein neues Thema zu bekommen. Hierbei können die Teilnehmer/innen ihr Vorwissen, ihre Ideen und Vorstellungen zu Tage bringen.

### **3.3.8 Feedback**

*„Feedback ist eine offene Rückmeldung an eine Person oder an eine Gruppe, wie ihr Verhalten von anderen wahrgenommen und gedeutet wird.“* (Reich, 2007). Diese Methode wird im Bildungsinstitut oft in Kombination mit Gruppenarbeiten und Referaten angewendet. Hierbei können die Teilnehmer/innen neue Seiten aus der Sicht einer anderen Person an sich wahrnehmen. Meist helfen Feedbackrunden innerhalb einer Gruppe auch das Selbstbild zu definieren.

### **3.3.9 Partnerarbeit**

Die Partnerarbeit dient dem Bearbeiten einer Aufgabe in gleichberechtigten Rollen. Hierbei wird die Partnerarbeit als Mittel zum Üben – ähnlich der Einzelarbeit – bei Teilnehmern/innen untereinander eingesetzt.

In höheren Modulen und bei besonderen Gelegenheiten wie dem LAP-Projekt wird für bestimmte Teilprobleme die Partnerarbeit auch zwischen Teilnehmer/innen und Trainer/innen angewandt, wobei acht zu geben ist, dass die Trainer/innen nicht in ihrer lehrenden Rolle verharren, sondern gleichberechtigt bleiben. Dies wird nur in Sonderfällen angewendet, sofern es für die Teilnehmer/innen nötig ist.

### **3.3.10 Referat**

Referate werden am Bildungsinstitut als einzelne Person oder in Gruppen gehalten und enden in der Regel in einer mündlichen Präsentation. Um mit Präsentationstechniken vertraut zu werden, werden in der Einführungsphase Seminare zu diesem Thema gehalten. Im Unterrichtsfach Programmieren werden Referate ab dem 3. Modul für Spezialisierungen in bestimmten Themengebieten eingesetzt.

Referate sind zwar sehr zeitintensiv, doch sie fördern die Selbstständigkeit der Teilnehmer/innen und werden auch für die Ausbildung zwingend benötigt – z.B. für die Endpräsentation des AKT-Projekts.

## 4 Unterrichtskonzept

### 4.1 Rahmenbedingungen

Ziel dieser Arbeit ist es, einen Vergleich zwischen einem prozedural vs. objektorientierten Programmieransatz aufstellen zu können.

Die Kurse, die hierfür durchgeführt werden, sind für Erwachsene mit Reha-Hintergrund. Jeder Kurs dauert über die Module 1 und 2, was 20 Wochen entspricht, wobei pro Woche 9 Einheiten à 45 Minuten Programmieren unterrichtet wird.

Ziel dieser Kurse ist, dass die Teilnehmer/innen nach Absolvierung dieser 20 Wochen die Grundlagen der objektorientierten Programmierung verstehen und anwenden können, sowie auch einfache Algorithmen in Hinblick auf die Lehrabschlussprüfung umsetzen können.

Es ist nicht angedacht, dass neben den Kurszeiten noch Übungen vergeben werden, somit findet die gesamte Wissensvermittlung und Übung während dieser Zeit statt. Es ist aber von Vorteil, wenn sich die Kursteilnehmer/innen auch außerhalb der Kurszeiten mit dem vermittelten Wissen auseinandersetzen.

Die Kurse finden in Schulungsräumen mit max. 20 Arbeitsplätzen statt. Allen Kursteilnehmer/innen wird ein Computer mit einer virtueller Maschine samt allen benötigten Programmen zur Verfügung gestellt. In der virtuellen Umgebung haben Teilnehmer/innen die Rechte eines Administrators und können gegebenenfalls zusätzlich benötigte Software installieren.

Die Größe der Kursgruppen kann stark variieren und beginnt bei zwei Teilnehmer/innen bis hin zu 20 Teilnehmer/innen.

Die Teilnehmer/innen des Kurses haben schon die Einführungsphase und das Modul 0 am Bildungsinstitut absolviert und somit schon erste Schritte in Richtung Programmierung getätigt. Hierbei wird mittels eines prozeduralen Lehransatzes die Programmiersprache VB.NET unterrichtet. Somit haben alle Kursteilnehmer/innen schon erste prozedurale Schritte getätigt. Der Programmierkurs in Modul 0 beträgt 10 Wochen à vier Einheiten und es werden Grundlagen wie

- Variablen,
- Datentypen,
- Einfache Ein-/Ausgabeoperationen,
- Rechenoperatoren,
- Verzweigungen und
- Schleifen

durchgenommen.

### 4.1.1 Hardware

Die Hardware pro Teilnehmer/in besteht aus einem Komplettsystem von Hewlett-Packard mit Intel Core i3-3240 und 8 GB RAM Arbeitsspeicher als Grundspezifikation. Diese Spezifikation ist nicht besonders unterschiedlich zu üblichen Office-PCs. Kursteilnehmer/innen mit besonderen Bedürfnissen, bekommen zusätzlich spezielle Eingabegeräte, z.B. ergonomisch geformte Tastaturen.

Die Hardware an sich ist nicht besonders wichtig, da im Grunde nur relativ einfache Programme lauffähig sein müssen, somit liegen die größten hardwaretechnischen Anforderungen an dem Betrieb der virtuellen Maschine, der Programmierumgebung und dem Management Studio für SQL. Im folgenden Kapitel werden die wichtigsten Softwarepakete für den Programmierunterricht aufgelistet.

### 4.1.2 Software

Folgende Software wird für den Programmierunterricht verwendet:

- VMware Player<sup>7</sup>  
Programm zum Ausführen der virtuellen Maschine
- Windows 7  
Als Betriebssystem
- Microsoft Visual Studio 2012  
Als Entwicklungsumgebung (IDE) für die Programmiersprache C#
- .NET Framework 4.0  
Als Basis für die Programmierung
- Adobe Acrobat Reader  
Für diverse Unterlagen als PDF
- Diverse Browser (Chrome, IE, Firefox)
- VLC<sup>8</sup>  
Für Lehrvideos
- Microsoft Office 2010  
Als Standardsoftwarepaket für Dokumentation und Präsentation

Bei Bedarf kann dieses Softwarepaket aber noch an den Unterricht angepasst erweitert werden.

---

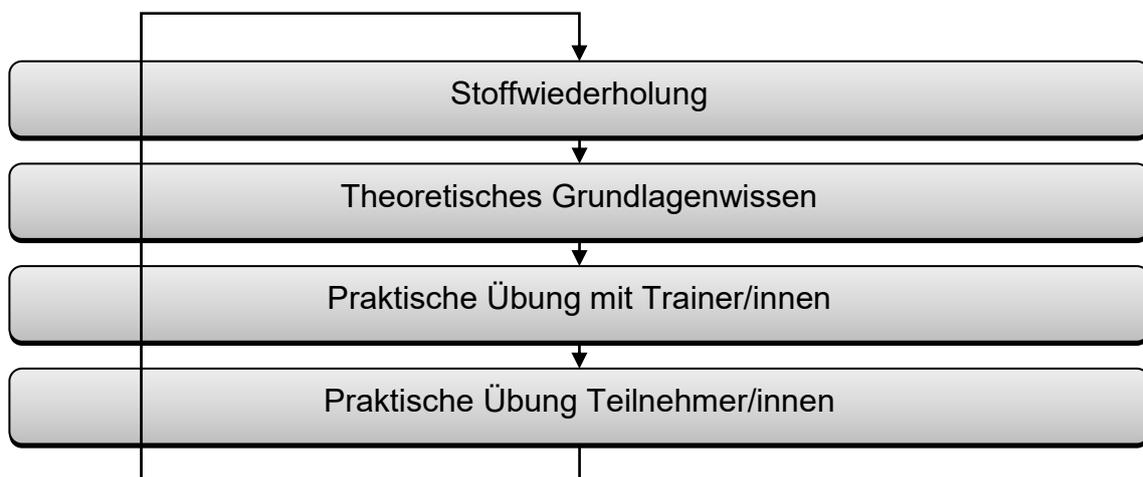
<sup>7</sup> <https://www.vmware.com/products/player>

<sup>8</sup> <http://www.videolan.org/>

## 4.2 Kursaufbau

### 4.2.1 Allgemein

Die Kurse sind dermaßen aufgebaut, dass theoretische Grundlagen nur kurz vermittelt werden, um so bald wie möglich zu praktischen Beispielen fortschreiten zu können. Die praktischen Beispiele sollen helfen die aktive Beteiligung laut (Hubwieser, 2007) der Teilnehmer/innen zu fördern. Diese praktischen Beispiele werden am Anfang noch sehr stark vom Lehrenden geführt, doch im Laufe der Zeit bekommen die Kursteilnehmer/innen die Möglichkeit diese immer selbstständiger zu lösen. Außerdem gibt es bei den Übungen meist Erweiterungen, damit die fortgeschrittenen Teilnehmer/innen auch gefordert werden. Die Kurse sind des Weiteren so gestaltet, dass die Teilnehmer/innen die praktischen Beispiele, die selbstständig zu lösen sind, dies auch innerhalb der Kurszeit bewerkstelligen können. Bevor ein Einheitenblock beginnt und bevor ein neues Kapitel besprochen wird, wird der gelernte Stoff wiederholt. Diese Wiederholungen sind ein wichtiger Aspekt der Kursplanung, damit den Kursteilnehmer/innen hierbei noch Raum für Klärungen etwaiger Unklarheiten zur Verfügung gestellt wird. Somit ergibt sich die Reihenfolge, dass zuerst der zuletzt erlernte Unterrichtsstoff wiederholt wird, danach neues theoretisches Wissen unterrichtet wird, darauf folgend ein oder mehrere praktische Beispiele miteinander durchgenommen werden und am Ende weitere Beispiele für selbständiges Lösen zur Verfügung gestellt werden. Bei all diesen Punkten ist eine aktive Kommunikation zwischen den Teilnehmern/innen untereinander und den Trainer/innen erwünscht.



**Abbildung 3: Unterrichtszyklus Einheitenblock**

Ein Einheitenblock entspricht im Unterrichtsfach Programmieren in der Regel drei bis fünf Einheiten. Damit ist gewährleistet, dass genügend Zeit zur Verfügung steht um in die Materie eintauchen zu können.

Die zeitliche Einteilung eines Einheitenblocks über drei Einheiten kann folgend aussehen:

- Max. 20 Minuten Stoffwiederholung
- Max. 25 Minuten theoretisches Wissen
- Min. 45 Minuten Praxis mit Trainer/innen
- Min. 45 Minuten Praxis Teilnehmer/innen

Bei komplexen Themenbereichen, kann sich dieser Unterrichtszyklus mit kürzeren Dauern öfters wiederholen, damit die Blöcke mit theoretischen Wissen nicht überhand nehmen und Teilnehmer/innen überfordert werden. Gerade bei einem Einheitenblock über fünf Einheiten, sollte dieser Zyklus zumindest zwei Mal stattfinden.

### **4.2.2 Prozedurale Gruppe**

Diese Zielgruppe lernt zuerst die Grundlagen der prozeduralen Programmierung kennen. Erst nachdem diese Grundlagen vermittelt und in ausreichenden Übungen verfestigt wurden, lernt die Gruppe noch die objektorientierten Erweiterungen.

Daraus ergibt sich, dass zuerst grundlegende Programmierkonzepte wie Variablen, Schleifen, Verzweigungen, etc. theoretisch wie auch praktisch behandelt werden und erst danach komplexere objektorientierte Strukturen kennen gelernt werden.

Der geplante Vorteil hierbei liegt einerseits daran, dass die Teilnehmer/innen schon im prozeduralen Denkschema des Modul 0 sind und dieses nur mehr auf die Programmiersprache C# ummünzen müssen.

Außerdem soll das Interesse an der Programmierung sehr früh mit kleinen leicht reproduzierbaren Programmen geweckt werden.

### **4.2.3 Objektorientierte Gruppe**

Die objektorientierte Gruppe beginnt gleich mit objektorientierten Grundlagen wie Klasse, Objekt und Instanz. Während diese Konzepte vermittelt werden, werden prozedurale Grundlagen stückweise eingestreut. So lernen zum Beispiel die Kursteilnehmer/innen mit den Eigenschaften (Properties) einer Klasse auch gleich die Verzweigung kennen oder mit den Methoden einer Klasse die verschiedenen Schleifenstrukturen.

Ziel dieser Vorgehensweise ist es die Teilnehmer/innen so schnell wie möglich in die objektorientierte Denkweise zu leiten, damit diese Denkweise in späteren Modulen nicht mehr schwer fällt.

## 4.3 Lernergebnisse

### 4.3.1 Allgemein

Im folgenden Kapitel setzen wir uns mit den Lernergebnissen auseinander. Diese werden für den Kurs in drei Klassifikationen unterteilt:

1. Kenntnisse
2. Fertigkeiten
3. Kompetenzen

Diese Aufteilung wurde dem Nationalen Qualifikationsrahmen (NQR) entnommen und dem Kursaufbau angepasst. Jede dieser Klassifikationen ist auf die vorhergehende aufbauend und wurde für den Kurs laut (Grün, Tritscher-Archan, und Silvia, 2009) folgend definiert.

*„**Kenntnisse** sind das Ergebnis der Verarbeitung von Information durch Lernen. Kenntnisse bezeichnen die Gesamtheit der Fakten, Grundsätze, Theorien und Praxis in einem Arbeits- oder Lernbereich. Im Europäischen Qualifikationsrahmen werden Kenntnisse als **Theorie und/oder Faktenwissen** beschrieben.“*

*„**Fertigkeiten** sind die Fähigkeit, **Kenntnisse anzuwenden und Know-how einzusetzen, um Aufgaben auszuführen und Probleme zu lösen**. Im Europäischen Qualifikationsrahmen werden Fertigkeiten als kognitive Fertigkeiten und praktische Fertigkeiten beschrieben.“*

*„**Kompetenz** ist die nachgewiesene Fähigkeit, Kenntnisse, Fertigkeiten sowie persönliche, soziale und methodische Fähigkeiten in Arbeits- oder Lernsituationen und für die berufliche und/oder persönliche Entwicklung zu nutzen. Im Europäischen Qualifikationsrahmen wird Kompetenz im Sinne der **Übernahme von Verantwortung und Selbständigkeit** beschrieben.“*

### 4.3.2 Lernergebnisse für den Kurs

Die Unterteilung in die drei Klassifikationen – Kenntnisse, Fertigkeiten und Kompetenzen – bildet die Grundlage für die Beschreibung der Lernergebnisse der Teilnehmer/innen, die das Unterrichtsfach Programmieren im Laufe der Ausbildung am Bildungsinstitut absolvieren.

Die definierten Lernergebnisse beziehen sich einerseits auf das Kapitel 2.1 Ausbildungsweg Informatik laut bmfwf mit der besonderen Anlehnung auf das Berufsprofil und das Berufsbild, welche die benötigten Qualifikationen von Informatiker/innen näher beschreiben. Andererseits beziehen sie sich auch auf die Rahmenbedingungen, die das Bildungsinstitut durch den vorgeschriebenen Ausbildungsweg laut Kapitel 2.2 Ausbildungsweg Informatik laut Bildungsinstitut, vorschreibt.

Hieraus entstehen folgende Beschreibungen für Lernergebnisse bezogen auf den kompletten Kurszeitraum in Bezug auf das Unterrichtsfach Programmieren. Da beide Gruppen (prozedurale vs. objektorientierte) am Ende der Ausbildung anhand des Wissens gleichgestellt sein sollen, ist diese Auflistung unabhängig der Lehrmethode:

Unit 1	Programmieren in Bezug auf Analytik		
	Er/Sie ist in der Lage, einfache Kontrollstrukturen anzuwenden um einfache Algorithmen umzusetzen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt Datentypen und ihre Größenzuordnungen	Er/Sie kann diverse Suchalgorithmen umsetzen	Er/Sie ist verantwortlich für die Auswahl der Algorithmen
	Er/Sie kennt Variablen und Konstanten und ihre Einsatzmöglichkeiten	Er/Sie kann diverse Sortieralgorithmen umsetzen	Er/Sie entscheidet die Datenstrukturen
	Er/Sie kann einfache Datenstrukturen erstellen	Er/Sie kann einfache Datenstrukturen erstellen	Er/Sie entscheidet über technische Hilfsmittel wie z.B. einzusetzende Klassenbibliotheken
	Er/Sie kennt einfache Datencontainer (Array, einfache Liste)	Er/Sie kann Datencontainer erstellen und verwenden	
	Er/Sie kennt Operatoren (arithmetische Operatoren, Zuweisungsoperatoren, relationale Operatoren, logische Operatoren) die für die Umsetzung von Algorithmen wichtig sind	Er/Sie kann Textaufgaben in Algorithmen umsetzen	
	Er/Sie kennt Strukturen für Verzweigungen		
	Er/Sie kennt Strukturen für Schleifen		
	Er/Sie kennt Vor- und Nachteile diverser		

	<p>Suchalgorithmen</p> <p>Er/Sie kennt Vor- und Nachteile diverser Sortieralgorithmen</p> <p>Er/Sie kennt verschiedene Darstellungsarten für Algorithmen</p>		
Unit 2	Programmieren in Bezug auf Datenorganisation		
	Er/Sie ist in der Lage, Daten in geordneter Form zu verwalten um diese bei Bedarf anlegen, finden, bearbeiten und entfernen zu können.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt CSV Dateien</p> <p>Er/Sie kennt XML Dateien</p> <p>Er/Sie kennt relationale Datenbanksysteme</p> <p>Er/Sie kennt die Grundlagen von SQL</p> <p>Er/Sie kennt programmiertechnisch alternative Abfragesprachen zu SQL, z.B. LINQ</p> <p>Er/Sie kennt Möglichkeiten des Zugriffs auf Daten, z.B. Streamreader, Entity Framework</p>	<p>Er/Sie definiert logische Datenstrukturen zu einzelnen Daten, z.B. Aufbau einer XML-Datei</p> <p>Er/Sie greift auf Daten unterschiedlicher Datenquellen zu und kann diese manipulieren</p> <p>Er/Sie kann Daten aus einer Quelle einer anderen Datenquelle zuordnen, z.B. Datenmapping mittels EF</p>	<p>Er/Sie ist verantwortlich für die Auswahl der Software für die Datenverwaltung</p> <p>Er/Sie koordiniert die Kommunikation zwischen Programmier- und Datenorganisationsteam, z.B. Schnittstellendefinition</p> <p>Er/Sie entscheidet über Technologien für den Datenzugriff</p>
Unit 3	Programmieren mit objektorientierten Bezug		
	Er/Sie ist in der Lage, komplexe Datenstrukturen anzuwenden um Kundenwünsche umzusetzen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Möglichkeiten komplexe	Er/Sie kann bestehende Datenstrukturen an	Er/Sie ist verantwortlich für die Planung der



	<p>Datenstrukturen zu erstellen</p> <p>Er/Sie kennt die Möglichkeiten Datenstrukturen wieder zu verwenden und anzupassen, z.B. Vererbung</p> <p>Er/Sie kennt die Vor- und Nachteile von Generika</p> <p>Er/Sie kennt die Eckpfeiler der objektorientierten Programmierung</p>	<p>Kundenwünsche anpassen</p> <p>Er/Sie kann neue Datenstrukturen für Kunden erstellen</p> <p>Er/Sie kann alleine einfache Kundenprojekte umsetzen</p> <p>Er/Sie kann im Team einfache Kundenprojekte umsetzen</p> <p>Er/Sie kann objektorientierte Paradigmen anwenden, z.B.: Polymorphie</p>	<p>Softwarearchitektur</p> <p>Er/Sie koordiniert das Programmiererteam</p> <p>Er/Sie entscheidet über technische Hilfsmittel</p>
Unit 4	Programmieren in Bezug auf Projektmanagement		
	Er/Sie ist in der Lage, Projektmanagementtechniken anzuwenden um Kundenwünsche umzusetzen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt verschiedene Lizenzmodelle</p> <p>Er/Sie kennt verschiedene Vorgehen im Projektmanagement, z.B. Scrum</p> <p>Er/Sie kennt den Unterschied zwischen Lasten- und Pflichtenheft</p> <p>Er/Sie kennt Möglichkeiten die Softwarequalität zu testen</p> <p>Er/Sie kennt Richtlinien für die Dokumentation von Softwareprojekten</p>	<p>Er/Sie kann ein Pflichtenheft erstellen</p> <p>Er/Sie kann eine Projektdokumentation erstellen</p> <p>Er/Sie kann Arbeitspakete erstellen und abarbeiten</p> <p>Er/Sie kann eine Installationsroutine erstellen</p> <p>Er/Sie kann den Aufwand für Softwareentwicklung grob abschätzen</p> <p>Er/Sie kann Software-</p>	<p>Er/Sie ist verantwortlich für die Aufgabenverteilung im Projekt</p> <p>Er/Sie entscheidet das Vorgehensmodell für das Projekt</p> <p>Er/Sie ist verantwortlich für die Qualitätskontrolle der Dokumentation</p> <p>Er/Sie entscheidet über das Versionierungstool</p> <p>Er/Sie entscheidet über die Testmethoden</p>

	Er/Sie versteht die Bedeutung von Versionsmanagement in einem Softwareprojekt	tests ausführen, z.B. Unit-Test  Er/Sie kann ein Versionierungstool in der Praxis einsetzen, z.B. SVN	
Unit 5	Programmieren in Bezug auf Oberflächen		
	Er/Sie ist in der Lage, eine grafische Oberfläche für ein Programm umzusetzen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt Möglichkeiten Oberflächen zu erzeugen  Er/Sie kennt die Grundlagen von Userinterfacedesign  Er/Sie kennt die Vor- und Nachteile der Trennung der Oberfläche vom Code-Behind	Er/Sie kann eine XML-basierende Oberfläche erzeugen, z.B. XAML, HTML  Er/Sie kann eine Oberfläche anhand von Designregeln gestalten  Er/Sie kann Schnittstellen zwischen Oberfläche und Code-Behind definieren und erzeugen  Er/Sie kann mehrsprachige Oberflächen gestalten  Er/Sie kann angepasste Oberflächen für Mobile- und Desktop-Applikationen gestalten	Er/Sie entscheidet über die anzuwendende Technologie für die Oberfläche  Er/Sie ist verantwortlich für die Auswahl des Tools zur Erzeugung der Oberfläche  Er/Sie kontrolliert die Regeln für das Userinterfacedesign  Er/Sie bestimmt die Schnittstellen zwischen Oberfläche und Code-Behind

**Abbildung 4: Lernergebnisse Kurs Allgemein**

Wie die einzelnen Units für die unterschiedlichen Kurse umgesetzt und aufgeteilt werden, wird im Kapitel 4.7 Lehreinheiten näher beschrieben.

### 4.3.3 Erhebung des Erfolgs

Die festgelegten Lernergebnisse, die in den einzelnen Units im Kapitel 4.3.2 Lernergebnisse für den Kurs festgehalten wurden und im Kapitel 4.7 Lehreinheiten näher

für die einzelnen Kurse beschrieben werden, werden auf mehrere unterschiedliche Wege erhoben. Die Erhebung selbst hängt vom Modul ab und ist wie folgt aufgeteilt:

- In den Modulen 1, 2 und 3 erfolgt die Erhebung mittels schriftlichen Prüfungen und Mitschriften der Trainer/innen.
- In den Modulen 4 und 5 erfolgt die Erhebung mittels der Skillcard (siehe Kapitel 4.3.3.3 Skillcard) und den Mitschriften der Trainer/innen
- In den Modulen 7 und 8 mittels des LAP-Projekts und der Mitschrift der Trainer/innen, sowie am Ende des 8. Moduls durch einen Fragebogen, den die Teilnehmer/innen ausfüllen.

#### **4.3.3.1 Schriftliche Prüfung**

Die schriftliche Prüfung dauert im Unterrichtsgegenstand Programmieren in der Regel zwei bis drei Einheiten zu je 45 Minuten. Hierbei besteht eine Prüfungsangabe aus theoretischen Fragen, die Kenntnisse abprüfen sollen, und praktischen Angaben, die Fertigkeiten und teilweise Kompetenzen überprüfen sollen.

Die schriftliche Prüfung wird einerseits als Leistungsfeststellung verwendet, soll aber andererseits auch als Informationsquelle dienen, welche Themen im Unterricht klarer verstanden wurden als andere.

Die Leistungsfeststellung wird laut § 2 Absatz 1 der Leistungsbeurteilungsverordnung<sup>9</sup> nur anhand der festgelegten Lernergebnisse aus dem Kapitel 4.3.2 Lernergebnisse für den Kurs festgelegt.

#### **4.3.3.2 Mitschriften der Trainer/innen**

Die Mitschrift der Trainer/innen soll Auffälligkeiten einer Klasse bezüglich der Lehreinheiten und somit bezüglich der unterrichteten Themen festhalten. Hiermit soll festgestellt werden, bei welchen Themen sich eine Kursgruppe schwerer oder leichter getan hat als eine andere Gruppe.

Außerdem sind in der Mitschrift der Trainer/innen die Ergebnisse der aktiven Beobachtung zu finden, sowie die Resultate der Leistungen, die die Teilnehmer/innen erbracht haben. Zum Beispiel werden besondere Beobachtungen in Hinblick auf die Antworten bei der fragend-entwickelnden Methode, Referaten, Gruppenarbeiten, Blitzlicht, Feedback und Projektarbeiten (wie in Kapitel 3.3 Verwendete Methoden dokumentiert) festgehalten.

#### **4.3.3.3 Skillcard**

Die Skillcard ist in Anlehnung an den NQR-Standard am Bildungsinstitut entwickelt worden, um das AKT-Projekt zu begleiten. Wie in Kapitel 2.2 Ausbildungsweg Informatik laut Bildungsinstitut erwähnt, findet das AKT-Projekt in den Modulen 4 und 5 statt. AKT steht hierbei für Anwendungs-Kompetenz-Training und bildet den Schwerpunkt dieser Module. AKT ist vergleichbar mit einer Übungsfirma, wo die Teilneh-

---

<sup>9</sup> <https://www.ris.bka.gv.at/Dokumente/Bundesnormen/NOR12119629/NOR12119629.html>

mer/innen von fiktiven Kund/innen ein Lastenheft bekommen und aus diesem ein Projekt von Anfang bis Ende durchführen.

In diesen Modulen bekommen die Teilnehmer/innen keine schriftlichen Prüfungen für die Leistungsfeststellung, sondern müssen eine gewisse Anzahl an Skills selbstständig absolvieren. Aus dieser Anzahl an Skills wird dann ihre Note berechnet. Einzelne Skills haben eine Gewichtung, die sich aus dem Aufwand und der Komplexität berechnet.

Kurz zusammengefasst entscheiden die Teilnehmer/innen selbst, welche Skills sie erfüllen möchten und entscheiden dadurch selbst, welche Kenntnisse, Fertigkeiten und/oder Kompetenzen sie abgenommen bekommen.

#### **4.3.3.4 LAP-Projekt**

In den Modulen 7 und 8 findet im Unterrichtsfach Programmieren nur mehr die Realisierung des Lehrabschluss-Projekts statt. Hierbei bekommen die Teilnehmer/innen am Anfang des Moduls 7 ein Pflichtenheft und müssen dieses bis Ende des Moduls 8 erfüllen. Die Teilnehmer/innen erfüllen hierbei ihr Projekt eigenverantwortlich, werden aber von den Trainer/innen bei technischen Problemen, Fragen oder Unklarheiten unterstützt. Dieses Projekt bildet die Grundlage für die praktische Lehrabschlussprüfung wie im Kapitel 2.3.3.3 Prüfungsphase – Tag 2: praktische Prüfung Projektarbeit beschrieben.

#### **4.3.3.5 Fragebogen**

Der Fragebogen ist die einzige Erhebung, die von den Teilnehmer/innen ausgefüllt wird. Hierbei soll festgestellt werden, in wieweit der Kurs aus Sicht der Teilnehmer/innen erfolgreicher war, ihnen Wissen und Motivation zur Programmierung zu vermitteln. Außerdem gibt diese Erhebung Aufschluss darüber, wie stark die Selbsteinschätzung der programmiertechnischen Fähigkeiten der Teilnehmer/innen mit den Aufzeichnungen und Tests des Bildungsinstituts übereinstimmen.

## **4.4 Methoden**

Die für die einzelnen Kurse verwendeten Methoden sind im Kapitel 3.3 Verwendete Methoden näher beschrieben. Welche Methode zu welcher Lehreinheit und welchem Kurs verwendet wird, wird im Kapitel 4.7 Lehreinheiten aufgelistet. Im Allgemeinen sind diese Methoden festgelegt, doch bei besonderem Bedarf der Anpassung der Lehreinheiten an die Klasse können sich diese auch ändern.

## **4.5 Vorkenntnisse**

Als Vorbereitung für beide Kurse werden am Bildungsinstitut die Einführungsphase und das Modul 0 – siehe Kapitel 2.2.3 Übersicht Modulaufteilung- vorausgesetzt. In dieser Phase werden gruppendedynamische Übungen in Richtung Teambuilding bestritten, sowie auch Computergrundlagen besprochen. Im Modul 0 werden auch die ersten Schritte in Richtung Programmierer/in mit der Programmiersprache Visual Ba-

sic.NET unternommen. Details zu dieser Phase findet man im Kapitel 2.2 Ausbildungsweg Informatik laut Bildungsinstitut, wo diese Phasen von Beginn an beschrieben sind.

Daraus ergibt sich, dass sich die Gruppe schon untereinander kennen gelernt hat und man im Kurs diese Phase überspringen kann. Außerdem kennt die Gruppe schon die Gepflogenheiten am Bildungsinstitut und hat sich mit der Arbeitsumgebung anfreunden können. Zusätzlich können die Trainer/innen ein Grundniveau in Form eines ECDL-Standard von den Teilnehmer/innen annehmen, sowie erste Erfahrungen in einer Programmiersprache – d.h. Grundlegende Vorstellungen in den Bereichen

- Grundstruktur eines Programmes
- Variablen und Datentypen
- Zuweisungs- und Rechenoperatoren
- Logische Operatoren (und, oder)
- Verzweigungen
- Schleifen

in der Programmiersprache VB.NET als Konsolenapplikation mit prozeduralem Lehransatz sind vorhanden. Dies ermöglicht einen raschen Einstieg in den Kurs, und man erspart sich einen großen Teil der typischen Kurseinführungsphase.

Falls im Modul 0 und im Kurs, der sich über die Module 1 und 2 erstreckt der/die gleiche Trainer/in unterrichtet, so ist auch schon ein persönliches Verhältnis zwischen Trainer/in und Teilnehmer/innen aufgebaut. Dies ist in der Regel auch der Fall, kann aber nicht immer vom Bildungsinstitut gewährleistet werden.

## 4.6 Kursvorbereitungen

Zur Kursvorbereitung müssen die Trainer/innen

- Sicherstellen, dass alle Teilnehmer/innen einen Computer mit einer virtuellen Maschine haben. Auf dieser ist jede benötigte Software wie z.B. das Visual Studio 2012 – siehe Kapitel 4.1.1 Hardware und 4.1.2 Software installiert.
- Übungsangaben, sowie Beispielprogramme auf dem Klassenserver bereitstellen, sowie beim Klassenvorstand prüfen ob alle Teilnehmer/innen auch schon die Programmierbücher erhalten haben.
- Lehreinheiten – siehe Kapitel 4.7 Lehreinheiten – an mögliche Ausfälle wie z.B. Feiertage oder Seminare anpassen.
- Angaben für die schriftlichen Prüfung in der letzten Woche jedes Moduls (außer 4, 5 und 7) vorbereiten und ausdrucken.
- Kontrollieren, ob der Klassenvorstand die Skillcard am Beginn des Moduls 4 ausgedruckt und an die Teilnehmer/innen verteilt hat.
- Fragebögen am Ende vom Modul 2 und 8 ausdrucken und verteilen.

## 4.7 Lehreinheiten

Im Kapitel 4.3.2 Lernergebnisse für den Kurs wurden anhand von Klassifizierungen – Kenntnisse, Fertigkeiten und Kompetenzen – einzelne Units beschrieben. Diese sollen das Können widerspiegeln, welches die Teilnehmer/innen des Bildungsinstituts nach der Ausbildung im Fach Programmieren nach den Modulen 1 und 2 aufzeigen können sollen. Kurz zusammengefasst lässt es sich so beschreiben:

Teilnehmer/innen sollen

- Selbstständig einfache Algorithmen umsetzen (Bubblesort, lineare Suche, etc.) können
- Autonom aus gelernten Programmierschritten neue Lösungen entwickeln können
- Komplexere Aufgabenstellungen mit Hilfe von externen Ressourcen (Internet, Buch, Programmbeispielen, etc.) lösen können
- Einzelschritte eines Programms analysieren und auswerten können
- Einfache Datenstrukturen selbstständig neu planen und umsetzen können
- Objektorientierte Grundlagen verstehen können und anhand von Aufgabenstellungen einsetzen können
- Externe Daten verwalten können
- Kundenwünsche effizient alleine oder in einem Team umsetzen können

Die Beschreibungen der Lehreinheiten beziehen sich nur auf das Unterrichtsfach Programmieren, können sich aber abschnittsweise mit anderen Unterrichtsgegenständen überschneiden. Ein Beispiel hierfür wären die Such- und Sortieralgorithmen, die in dem Unterrichtsgegenstand Analytik theoretisch besprochen werden, aber in Programmieren praktisch umgesetzt werden.

Die Definition der einzelnen Lehreinheiten hat immer in Anlehnung an die Fachdidaktik die Grundfrage:

*„Welche Kompetenz soll wann und wie erworben werden?“* (Schubert und Schwill, 2011)

In diesem Kapitel werden die programmiertechnischen Grundlagen, die für diese Diplomarbeit die wichtigsten sind, anhand eines prozeduralen und im Vergleich dazu eines objektorientierten Programmieransatzes aufbereitet. Diese verschiedenen Ansätze werden im Kapitel 4.7.1 Grundlagen – Module 1 und 2 durchgenommen und bilden die Grundlage für die spätere Evaluierung. Die Kapitel 4.7.2 Benutzeroberfläche – Modul 3, 4.7.3 Aktives Kompetenz Training – Module 4 und 5 und 4.7.4 Lehrabschluss – Module 7 und 8 werden nicht unterschieden und unterliegen einem objektorientierten Programmieransatz. Daraus ergibt sich, dass diese nicht direkt für den Vergleich des prozeduralen und objektorientierten Ansatz herangezogen werden. Doch diese Kapitel dienen der Feststellung, ob die Teilnehmer/innen das objektorientierte Konzept wirklich verstanden haben und sind somit die Grundlage für

die erweiterte Evaluierung, wo festgestellt wird, welcher Lehransatz der effektivere ist. Aus diesem Grund werden auch diese Lehreinheiten in diesem Kapitel vorgestellt, um eine allgemeine Übersicht über die Rahmenbedingungen der Lehreinheiten über den gesamten Zeitraum zu geben.

## 4.7.1 Grundlagen – Module 1 und 2

### 4.7.1.1 Übersicht Lehreinheiten prozedural vs. objektorientiert

In Anlehnung der Planung der Lehreinheiten für den Kurs laut (Suh, 2007) ist folgende Übersicht entstanden:

Themengebiet	Wochen		Anmerkung
	Prozedural	Objektorientiert	
Plattform, IDE kennenlernen	1	1	.NET Framework, Visual Studio, Code snippet
Datentypen, I/O	1	1,5	Wertdatentypen, Konsolen I/O
Kontrollstrukturen	3	2,5	Schleifen (while, for, foreach), Verzweigungen (if, switch)
Array/List	2,5	2,5	Eindimensionaler Schwerpunkt
Funktionen/Methoden	2	1,5	
Strukturen	2	0,5	Als Datentyp
Klassen	3	4	Felder, Eigenschaften
Vererbung, Polymorphie und Klassen-Design	2	3,5	
Suchalgorithmen	1,5	1	Lineare Suche, binäre Suche
Sortieralgorithmen	1	1	Bubblesort, Quicksort
Test	1	1	Mit Vor- und Nachbereitung
<b>Summe</b>	<b>20</b>	<b>20</b>	

Tabelle 7: Übersicht Lehreinheiten

### 4.7.1.2 Aufteilung prozedurale Gruppe

Die prozedurale Gruppe verwendet den Leitsatz „... use an OOP language such as Java and teach it in stages. The first stage would focus only on the procedural aspects of the language, and the second stage would introduce object-oriented methods only after a procedural foundation was firmly laid.“ (Burton und Bruhn, 2003).

Die Programmiersprache ist zwar nicht Java, sondern C#.NET, doch durch die sehr starke Ähnlichkeit der Programmiersprachen kann der Ansatz 1:1 übernommen werden. Jede Lehreinheit wird folgend mit Unit beschriftet und beinhaltet angelehnt an den NQR Standard eine Beschreibung, was die Teilnehmer/innen nach Absolvierung dieser Unit können sollten. Zusätzlich zum Standard werden empfohlene Unterrichtswochen für eine Reihenfolge vorgeschlagen, sowie die Methoden, die für den Unterricht verwendet werden sollten. Aus diesen Rahmenbedingungen ist folgende Aufteilung der Lehreinheiten entstanden.



Unit 1	.NET und IDE		
	Er/Sie ist in der Lage die Entwicklungsumgebung (IDE) einzurichten und zu bedienen, um einfache Programme schreiben zu können, und versteht die Grundlagen des .NET Frameworks.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt die Grundlagen des .NET Frameworks – Aufbau, Versionsunterschiede, Vor- und Nachteile.</p> <p>Er/Sie kennt die Befehle ein Programm auch außerhalb einer IDE zu kompilieren.</p> <p>Er/Sie kennt die Möglichkeiten ein Projekt in einer IDE anzulegen.</p> <p>Er/Sie kennt die Möglichkeiten benötigte Fenster in der IDE auszurichten, sowie zu erstellen.</p> <p>Er/Sie kennt den grundlegenden Ablauf vom Quellcode zur ausführbaren Datei.</p> <p>Er/Sie kennt den Unterschied zwischen verschiedenen Projektvorlagen.</p>	<p>Er/Sie verwaltet Projektmappen und Projekte.</p> <p>Er/Sie wählt das entsprechende .NET Framework für ein Projekt aus.</p> <p>Er/Sie kann ein Programm für eine spezielle Framework Version in der IDE kompilieren.</p> <p>Er/Sie kann mit Hilfe der Fehlerausgabe Ansätze finden wo diese zu finden ist.</p> <p>Er/Sie findet sich in der Ordnerstruktur eines Projektes zurecht.</p> <p>Er/Sie kann neue Quelldateien anlegen.</p>	<p>Er/Sie ist verantwortlich für die Konfiguration der IDE.</p> <p>Er/Sie ist verantwortlich für die Dateiverwaltung der eigenen Projekte.</p> <p>Er/Sie entscheidet über die anzuwendende Projektvorlage.</p>
Unterrichtswoche	Modul 1 Woche 1		
Methoden	Frontalunterricht, Fragend-entwickelnd, Einzelarbeit		
Anmerkung	Spielerisches Kennenlernen der IDE. Möglichkeit eines ersten <i>Hello World</i> Programms.		
Unit 2	Datentypen und I/O		
	Er/Sie ist in der Lage zwischen Datentypen zu unterscheiden und kann einfache Ein- und Ausgaben veranlassen.		

	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt alle grundlegenden Datentypen – Wertdatentypen - sowie den Datentyp String.</p> <p>Er/Sie kennt die Bedeutung von Variablen und Konstanten.</p> <p>Er/Sie kennt die Bedeutung von Zuweisungsoperatoren.</p> <p>Er/Sie kennt arithmetische Operatoren.</p> <p>Er/Sie kennt die allgemeine Abarbeitungsreihenfolge von Befehlen.</p> <p>Er/Sie kennt die Möglichkeiten Text auf dem Bildschirm auszugeben.</p> <p>Er/Sie kennt die Möglichkeiten Text einlesen zu lassen.</p> <p>Er/Sie kennt den Vorrang der wichtigsten Operatoren</p>	<p>Er/Sie kann Variablen erzeugen.</p> <p>Er/Sie kann Werte auf Variablen speichern.</p> <p>Er/Sie kann einfach formatierten Text am Bildschirm ausgeben.</p> <p>Er/Sie kann Werte von Variablen am Bildschirm ausgeben lassen.</p> <p>Er/Sie ist in der Lage einfache Typumwandlungen vorzunehmen.</p>	<p>Er/Sie entscheidet über die richtige Wahl des Datentyps.</p> <p>Er/Sie überprüft die Abarbeitungsreihenfolge von Programmbefehlen.</p>
Unterrichtswoche	Modul 1 Woche 2		
Methoden	Frontalunterricht, Einzelarbeit, Partnerarbeit		
Anmerkung	Strukturen als Werttypen werden extra behandelt. Erstellen eines einfachen Programms wie ein <i>Papagei</i> , der alle Eingaben wiedergibt, um I/O zu behandeln.		
Unit 3	Kontrollstrukturen		
	Er/Sie ist in der Lage den Ablauf eines Programms mit Hilfe von Kontrollstrukturen zu steuern.		

	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt den Aufbau und Bedeutung einer Schleife.	Er/Sie kann eine kopfgesteuerte Schleife anwenden.	Er/Sie entscheidet über die Wahl der Schleife.
	Er/Sie kennt den Aufbau und Bedeutung einer Verzweigung.	Er/Sie kann eine fußgesteuerte Schleife anwenden.	Er/Sie entscheidet über die Wahl der Verzweigung.
	Er/Sie kennt relationale Operatoren.	Er/Sie kann eine Zählschleife anwenden.	Er/Sie legt die Reihenfolge der Abarbeitung fest.
	Er/Sie kennt logische Operatoren.	Er/Sie kann eine einfache Verzweigung anwenden.  Er/Sie kann eine Mehrfachverzweigung anwenden.	
Unterrichtswoche		Modul 1 Woche 3, 4, 5	
Methoden		Frontalunterricht, Fragend-entwickelnd, Einzelarbeit, Blitzlicht	
Anmerkung		In der fünften Modulwoche soll der/die Trainer/in eine Erhebung bezüglich des Wissenstands machen, diese Erhebung findet zumeist mit der fragend-entwickelnden Methode statt.	
Unit 4	Arrays und Listen		
	Er/Sie ist in der Lage mehrere Daten eines Datentyps in einer Struktur zu verwalten.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Definition eines Arrays.	Er/Sie kann ein Array implementieren und Daten ablegen.	Er/Sie trifft die Entscheidung ob ein Array oder eine dynamische Liste für ein Problem passender ist.
	Er/Sie kennt die Definition einer dynamischen Liste.	Er/Sie kann ein Array auf verschiedene Arten durchlaufen.	
	Er/Sie kennt die Differenzen zwischen eines Arrays und einer dynamischen Liste.	Er/Sie kann Daten in einem Array verwalten.  Er/Sie kann eine Liste	

		implementieren und Daten ablegen.  Er/Sie kann eine Liste auf verschiedene Arten durchlaufen.  Er/Sie kann Daten in einer Liste verwalten.	
Unterrichtswoche		Modul 1 Woche 6, 7, 8 (teilweise)	
Methoden		Frontalunterricht, Fragend-entwickelnd, Einzelarbeit, Partnerarbeit	
Anmerkung		Es werden nur eindimensionale Arrays und Listen behandelt, sofern nicht genügend Zeit für mehrdimensionale oder verschachtelte Datenstrukturen vorhanden ist.	
Unit 5	Methoden (Prozeduren und Funktionen)		
	Er/Sie ist in der Lage Programme sinnvoll in Methoden zu unterteilen und diese zu steuern.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Bedeutung von Methoden.  Er/Sie kennt den Aufbau von Methoden.  Er/Sie kennt die Bedeutung des Methoden-Stack.  Er/Sie kennt den Unterschied zwischen Prozedur und Funktion.	Er/Sie kann eine Methode ohne Parameter erstellen.  Er/Sie kann eine Methode mit Wertparameter erstellen.  Er/Sie kann eine Methode mit Rückgabe erstellen.	Er/Sie ist verantwortlich für die Gestaltung der Methodensignatur.  Er/Sie überwacht die Aufteilung einer Methode in mehrere sinnvolle Methoden.
Unterrichtswoche		Modul 1 Woche 8, 9, 10 (teilweise)	
Methoden		Frontalunterricht, Fragend-entwickelnd, Einzelarbeit, Feedback	
Anmerkung		In der Modulwoche 10 findet der schriftliche Test statt. Am Ende des Moduls soll eine Feedback-Runde stattfinden.	
Unit 6	Suchalgorithmen		
	Er/Sie ist in der Lage die lineare Suche und die binäre Suche zu implementieren.		
	Kenntnisse	Fertigkeiten	Kompetenz

	<p>Er/Sie kennt die Funktionsweise der linearen Suche.</p> <p>Er/Sie kennt die Funktionsweise der binären Suche.</p> <p>Er/Sie kennt die Vor- und Nachteile beider Suchalgorithmen.</p>	<p>Er/Sie kann die lineare Suche implementieren.</p> <p>Er/Sie kann die binäre Suche implementieren.</p> <p>Er/Sie kann vorhandene Implementierungen des .NET Frameworks für die Suche verwenden.</p>	<p>Er/Sie trägt die Entscheidung welcher Suchalgorithmus für das Problem der passendste ist.</p>
Unterrichtswoche		Modul 2 Woche 1, 2 (teilweise)	
Methoden		Frontalunterricht, fragend-entwickelnd, Gruppenarbeit, Einzelarbeit	
Anmerkung		Die Algorithmen sollen zuerst in einer Gruppenarbeit ohne Computer behandelt werden. Die Implementation soll als Einzelarbeit mittels Computer stattfinden.	
Unit 7	Strukturen		
	Er/Sie ist in der Lage einen zusammengesetzten Typ sinnvoll zu erstellen und einsetzen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt die Bedeutung einer Struktur.</p> <p>Er/Sie kennt die Vor- und Nachteile einer Struktur.</p>	<p>Er/Sie kann eine Struktur erstellen.</p> <p>Er/Sie kann eine Struktur verwenden.</p>	<p>Er/Sie ist verantwortlich für die Planung einer Struktur.</p>
Unterrichtswoche		Modul 2 Woche 2, 3, 4 (teilweise)	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Partnerarbeit	
Anmerkung		Diese Unit soll die Vorbereitung für abstraktes Denken und Klassendesign sein.	
Unit 8	Klassen		
	Er/Sie ist in der Lage eine Klassendefinition zu planen, diese aufzubauen und sinnvoll einzusetzen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt die Bedeutung einer Klasse.</p> <p>Er/Sie kennt den Auf-</p>	<p>Er/Sie kann eine Klasse erstellen.</p> <p>Er/Sie kann Instanzen</p>	<p>Er/Sie ist verantwortlich für die Planung einer Klasse.</p>

	<p>bau einer Klasse und ihre Member.</p> <p>Er/Sie kennt Sichtbarkeiten einer Klasse.</p> <p>Er/Sie kennt die Bedeutung von Klassenbibliotheken.</p> <p>Er/Sie kennt die Begriffe: Klasse, Objekt, Instanz.</p> <p>Er/Sie kennt die Bedeutung eines Klassendiagramms.</p>	<p>einer Klasse erzeugen.</p> <p>Er/Sie kann mit Member einer Klasse arbeiten.</p> <p>Er/Sie kann eine Klassenbibliothek erstellen.</p> <p>Er/Sie kann eine fremde Klassenbibliothek verwenden.</p> <p>Er/Sie kann aus vorhandenen Klassen ein Klassendiagramm erstellen.</p>	<p>Er/Sie ist verantwortlich für die Planung komplexer Datentypen und ihre Beziehungen zueinander, z.B. ist eine Klasse ein Datentyp in einer anderen Klasse.</p>
Unterrichtswoche		Modul 2 Woche 4, 5, 6, 7 (teilweise)	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Partnerarbeit, Blitzlicht	
Anmerkung		In der fünften Modulwoche soll der/die Trainer/in eine Erhebung bezüglich des Wissenstands machen.	
Unit 9	Vererbung, Polymorphie und Klassen-Design		
	Er/Sie ist in der Lage fortgeschrittene Techniken für das Klassendesign anzuwenden.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt die Definition von Vererbung.</p> <p>Er/Sie kennt die Vorteile von Vererbung.</p> <p>Er/Sie kennt die Vorteile von Datenkapselung.</p> <p>Er/Sie kennt eine allgemeine Definition von Polymorphie.</p> <p>Er/Sie kennt Designpattern und ihre allgemeine Bedeutung.</p>	<p>Er/Sie kann eine Vererbungshierarchie aufbauen.</p> <p>Er/Sie kann Polymorphie in einem praktischen Beispiel anwenden.</p> <p>Er/Sie kann Zugriffsebenen im Sinne der Datenkapselung erstellen.</p>	<p>Er/Sie plant eine Klassenstruktur mit Vererbung.</p> <p>Er/Sie überprüft die richtige Anwendung von Designpattern.</p>

Unterrichtswoche	Modul 2 Woche 7, 8, 9 (teilweise)		
Methoden	Frontalunterricht, fragend-entwickelnd, Gruppenarbeit, Einzelarbeit		
Anmerkung	Spezielle Designpattern werden nur rudimentär behandelt, z.B. Singleton und Multiton. Falls Zeit bleibt können auch Schnittstellen und abstrakte Klassen besprochen werden.		
Unit 10	Sortieralgorithmen		
	Er/Sie ist in der Lage diverse Sortieralgorithmen zu erklären und implementieren.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Funktionsweise des Bubblesort.	Er/Sie kann den Bubblesort implementieren.	Er/Sie entscheidet über den einzusetzenden Sortieralgorithmus.
	Er/Sie kennt die Funktionsweise des Quicksort.	Er/Sie kann den Quicksort implementieren.	
	Er/Sie kennt Methoden des .NET Frameworks zum Sortieren.	Er/Sie kann Sortiermethoden des .NET Frameworks einsetzen.	
Unterrichtswoche	Modul 2 Woche 9, 10 (teilweise)		
Methoden	Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Gruppenarbeit		
Anmerkung	Die Algorithmen sollen zuerst in einer Gruppenarbeit ohne Computer behandelt werden. Die Implementation soll als Einzelarbeit mittels Computer stattfinden. In der Modulwoche 10 findet der schriftliche Test statt. Am Ende des Moduls soll eine Feedback-Runde stattfinden.		

**Tabelle 8: Lehreinheiten prozedurale Gruppe Module 1 und 2**

#### **4.7.1.3 Aufteilung objektorientierte Gruppe**

Die objektorientierte Gruppe arbeitet laut dem Leitsatz von (Burton und Bruhn, 2003): „... *object-oriented thinking at the earliest possible opportunity*“. Hierbei wird also der Fokus darauf gelegt, ein objektorientiertes Denken so schnell wie möglich zu Erlangen. Prozedurale Konzepte werden zweitrangig behandelt. Die Herausforderung hierbei ist die objektorientierte Denkweise schnell verständlich zu vermitteln, damit die Teilnehmer/innen und die Trainer/innen auf einer Kommunikationsebene interagieren. Hieraus ergibt sich eine andere Reihenfolge der Units, sowie eine leichte Verschiebung der Zeiteinheiten, da die Priorität anders gelegt werden als bei dem

prozeduralem Ansatz. Diese Unterschiede anhand der Zeiteinheiten lassen sich sehr gut in der Übersicht Lehreinheiten prozedural vs. objektorientiert ablesen.



Unit 1	.NET und IDE		
	Er/Sie ist in der Lage die IDE zu einzurichten und zu bedienen um einfache Programme schreiben zu können und versteht die Grundlagen des .NET Frameworks.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt die Grundlagen des .NET Frameworks – Aufbau, Versionsunterschiede, Vor- und Nachteile.</p> <p>Er/Sie kennt die Befehle ein Programm auch außerhalb einer IDE zu kompilieren.</p> <p>Er/Sie kennt die Möglichkeiten ein Projekt in einer IDE anzulegen.</p> <p>Er/Sie kennt die Möglichkeiten benötigte Fenster in der IDE auszurichten, sowie zu erstellen.</p> <p>Er/Sie kennt den grundlegenden Ablauf vom Quellcode zur ausführbaren Datei.</p> <p>Er/Sie kennt den Unterschied zwischen verschiedenen Projektvorlagen.</p>	<p>Er/Sie erstellt Projektmappen und Projekte.</p> <p>Er/Sie wählt das entsprechende .NET Framework für ein Projekt aus.</p> <p>Er/Sie kann ein Programm für ein entsprechendes Framework in der IDE kompilieren.</p> <p>Er/Sie kann mit Hilfe der Fehlerausgabe Ansätze finden wo diese zu finden ist.</p> <p>Er/Sie findet sich in der Ordnerstruktur eines Projektes zurecht.</p> <p>Er/Sie kann neue Quelldateien anlegen.</p>	<p>Er/Sie ist verantwortlich für die Konfiguration der IDE.</p> <p>Er/Sie ist verantwortlich für die Dateiverwaltung der eigenen Projekte.</p>
Unterrichtswoche	Modul 1 Woche 1		
Methoden	Frontalunterricht, Fragend-entwickelnd, Einzelarbeit		
Anmerkung	Spielerisches Kennenlernen der IDE. Möglichkeit eines ersten <i>Hello World</i> Programms. Fokussiert auf objektorientiertes Denken, z.B. Projektvorlage Klassenbibliothek herausheben und die Klasse <i>Program</i> einer Konsolenanwendung gleich erläu-		

	tern.		
Unit 2	Datentypen und I/O		
	Er/Sie ist in der Lage zwischen Wert- und Verweisdatentypen zu unterscheiden und kann einfache Ein- und Ausgaben veranlassen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt alle grundlegenden Wertdatentypen.</p> <p>Er/Sie kennt den Unterschied zwischen Wert- und Verweisdatentypen.</p> <p>Er/Sie kennt die Bedeutung von Variablen.</p> <p>Er/Sie kennt die Bedeutung von Zuweisungsoperatoren.</p> <p>Er/Sie kennt arithmetische Operatoren.</p> <p>Er/Sie kennt die allgemeine Abarbeitungsreihenfolge von Befehlen.</p> <p>Er/Sie kennt die Möglichkeiten Text auf den Bildschirm auszugeben.</p> <p>Er/Sie kennt die Möglichkeiten Text einlesen zu lassen.</p> <p>Er/Sie kennt den Vorrang der wichtigsten Operatoren</p>	<p>Er/Sie kann Variablen erzeugen.</p> <p>Er/Sie kann Werte auf Variablen speichern.</p> <p>Er/Sie kann Verweise auf Variablen speichern.</p> <p>Er/Sie kann einfach formatierten Text am Bildschirm ausgeben.</p> <p>Er/Sie kann Werte von Variablen am Bildschirm ausgeben.</p> <p>Er/Sie ist in der Lage einfache Typumwandlungen vorzunehmen – Boxing.</p>	<p>Er/Sie entscheidet über die richtige Wahl des Datentyps.</p>
Unterrichtswoche	Modul 1 Woche 2, 3 (teilweise)		
Methoden	Frontalunterricht, fragend-entwickelnd, Partnerar-		

	beit, Einzelarbeit		
Anmerkung	Es werden trotz dem objektorientierten Ansatzes, zuerst Werttypen erklärt, da dieses Wissen schon vom Modul 0 vorhanden ist und somit nur auf C# angepasst werden muss. Der Fokus liegt aber bei Verweisdatentypen. Strukturen als Werttypen werden extra behandelt.		
Unit 3	Strukturen		
	Er/Sie ist in der Lage einen zusammengesetzten Typ sinnvoll zu erstellen und einsetzen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Bedeutung einer Struktur.	Er/Sie kann eine Struktur erstellen.	Er/Sie ist verantwortlich für die Planung einer Struktur.
	Er/Sie kennt die Vor- und Nachteile einer Struktur.	Er/Sie kann eine Struktur verwenden.	
Unterrichtswoche	Modul 1 Woche 3 (teilweise)		
Methoden	Frontalunterricht, fragend-entwickelnd, Einzelarbeit		
Anmerkung	Strukturen werden nur rudimentär behandelt, Aufbau sehr einfacher Strukturen wie z.B.: Punkt der aus zwei Integer besteht.		
Unit 4	Klassen Felder und Eigenschaften		
	Er/Sie ist in der Lage eine Klassendefinition zu planen, diese aufzubauen und sinnvoll einzusetzen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kann die Bedeutung einer Klasse.	Er/Sie kann eine Klasse erstellen.	Er/Sie ist verantwortlich für die Planung einer Klasse.
	Er/Sie kennt den Aufbau einer Klasse und ihre Member.	Er/Sie kann Felder einer Klasse definieren.	
	Er/Sie kennt Sichtbarkeiten einer Klasse.	Er/Sie kann Eigenschaften einer Klasse definieren.	Er/Sie ist verantwortlich für die Planung komplexer Datentypen und ihre Beziehungen zueinander.
	Er/Sie kennt die Bedeutung von Klassenbibliotheken.	Er/Sie kann Instanzen einer Klasse erzeugen.	
	Er/Sie kennt die Begrif-	Er/Sie kann mit Mem-	

	<p>fe: Klasse, Objekt, Instanz.</p> <p>Er/Sie kennt die Bedeutung eines Klassendiagramms.</p> <p>Er/Sie kennt den Unterschied einer Klasse und einer Struktur.</p>	<p>ber einer Klasse arbeiten.</p> <p>Er/Sie kann eine Klassenbibliothek erstellen.</p> <p>Er/Sie kann eine fremde Klassenbibliothek verwenden.</p> <p>Er/Sie kann aus vorhandenen Klassen ein Klassendiagramm erstellen und erklären.</p>	
Unterrichtswoche		Modul 1 Woche 4, 5, 6, 8	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Partnerarbeit, Blitzlicht	
Anmerkung		In der fünften Modulwoche soll der/die Trainer/in eine Erhebung des Wissenstands machen.	
Unit 5	Kontrollstrukturen - Verzweigung		
	Er/Sie ist in der Lage den Ablauf eines Programms mit Hilfe von Verzweigungen zu steuern.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt den Aufbau und Bedeutung einer Verzweigung.</p> <p>Er/Sie kennt relationale Operatoren.</p> <p>Er/Sie kennt logische Operatoren.</p>	<p>Er/Sie kann eine einfache Verzweigung anwenden.</p> <p>Er/Sie kann eine Mehrfachverzweigung anwenden.</p>	<p>Er/Sie entscheidet über die Wahl der Verzweigung.</p> <p>Er/Sie legt die Reihenfolge der Abarbeitung fest.</p>
Unterrichtswoche		Modul 1 Woche 7	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Partnerarbeit	
Anmerkung		Die Verzweigung wird in Kombination mit der set-Eigenschaft verwendet, damit auf bestimmte Werte überprüft wird.	
Unit 6	Methoden (Prozeduren und Funktionen)		
	Er/Sie ist in der Lage Programme sinnvoll in Methoden zu unterteilen und diese zu steuern.		
	Kenntnisse	Fertigkeiten	Kompetenz

	<p>Er/Sie kennt die Bedeutung von Methoden.</p> <p>Er/Sie kennt den Aufbau von Methoden.</p> <p>Er/Sie kennt den Unterschied zwischen Instanz- und Klassenmethoden.</p> <p>Er/Sie kennt die Bedeutung des Methoden-Stack.</p>	<p>Er/Sie kann eine Methode ohne Parameter erstellen.</p> <p>Er/Sie kann eine Methode mit Wertparameter erstellen und erklären.</p> <p>Er/Sie kann eine Methode mit Verweisparameter erstellen und erklären.</p> <p>Er/Sie kann eine Methode mit Rückgabe erstellen.</p>	<p>Er/Sie ist verantwortlich für die Gestaltung der Methodensignatur.</p> <p>Er/Sie überwacht die Aufteilung einer Methode in mehrere sinnvolle Methoden.</p>
Unterrichtswoche		Modul 1 Woche 9, 10 (teilweise)	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Feedback	
Anmerkung		In der Modulwoche 10 findet der schriftliche Test statt.	
Unit 7	Kontrollstrukturen - Schleifen		
	Er/Sie ist in der Lage den Ablauf eines Programms mit Hilfe von Schleifen zu steuern.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt den Aufbau und Bedeutung verschiedener Schleifentypen.</p> <p>Er/Sie kennt die Gefahren einer Endlosschleife.</p>	<p>Er/Sie kann eine kopfgesteuerte Schleife anwenden.</p> <p>Er/Sie kann eine fußgesteuerte Schleife anwenden.</p> <p>Er/Sie kann eine Zählschleife anwenden.</p>	<p>Er/Sie entscheidet über die Wahl der Schleife.</p> <p>Er/Sie legt die Reihenfolge der Abarbeitung fest.</p>
Unterrichtswoche		Modul 2 Woche 1, 2 (teilweise)	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit	
Anmerkung		Soll vermehrt in Kombination mit Methoden verwendet werden, damit diese im Gedächtnis blei-	

	ben.		
Unit 8	Arrays und Listen		
	Er/Sie ist in der Lage mehrere Daten eines Datentyps in einer Struktur zu verwalten.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Definition eines Arrays.  Er/Sie kennt die Definition einer dynamischen Liste  Er/Sie kennt die Differenzen zwischen eines Arrays und einer dynamischen Liste	Er/Sie kann ein Array implementieren und Daten ablegen.  Er/Sie kann ein Array auf verschiedene Arten durchlaufen.  Er/Sie kann Daten in einem Array verwalten.  Er/Sie kann eine Liste implementieren und Daten ablegen.  Er/Sie kann eine Liste auf verschiedene Arten durchlaufen.  Er/Sie kann Daten in einer Liste verwalten.	Er/Sie trifft die Entscheidung ob ein Array oder eine dynamische Liste für ein Problem passender ist.
Unterrichtswoche	Modul 2 Woche 2 (teilweise), 3, 4		
Methoden	Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Partnerarbeit		
Anmerkung	Es werden nur eindimensionale Arrays und Listen behandelt. Der Fokus liegt auf den Listen. Bei genügend Zeit soll auch der generische Teil von Listen erklärt werden (Generika).		
Unit 9	Vererbung, Polymorphie und Klassen-Design		
	Er/Sie ist in der Lage fortgeschrittene Techniken für das Klassendesign anzuwenden.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Definition von Vererbung.  Er/Sie kennt die Vorteile und Gefahren von	Er/Sie kann eine Vererbungshierarchie aufbauen.  Er/Sie kann Polymor-	Er/Sie plant eine Klassenstruktur.  Er/Sie überprüft die Richtige Anwendung

	<p>Vererbung.</p> <p>Er/Sie kennt die Vorteile von Datenkapselung.</p> <p>Er/Sie kennt eine allgemeine Definition von Polymorphie.</p> <p>Er/Sie kennt Designpattern.</p>	<p>phie in einem praktischen Beispiel anwenden.</p> <p>Er/Sie kann Zugriffsebenen im Sinne der Datenkapselung erstellen.</p>	<p>von Designpattern.</p>
Unterrichtswoche		Modul 2 Woche 5, 6, 7, 8 (teilweise)	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Partnerarbeit, Blitzlicht	
Anmerkung		<p>Klassendiagramme sollen wiederholt werden. Für das Beispiel für die Polymorphie sollen am besten Listen eingesetzt werden. Falls Zeit bleibt können auch Schnittstellen und abstrakte Klassen besprochen werden.</p> <p>In der fünften Modulwoche soll der/die Trainer/in eine Erhebung des Wissenstands machen.</p>	
Unit 10	Suchalgorithmen		
	Er/Sie ist in der Lage die lineare Suche und die binäre Suche zu implementieren.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt die Funktionsweise der linearen Suche.</p> <p>Er/Sie kennt die Funktionsweise der binären Suche.</p> <p>Er/Sie kennt die Vor- und Nachteile der beiden Suchalgorithmen.</p>	<p>Er/Sie kann die lineare Suche implementieren.</p> <p>Er/Sie kann die binäre Suche implementieren.</p> <p>Er/Sie kann vorhandene Implementierungen des .NET Frameworks für die Suchalgorithmen verwenden.</p>	<p>Er/Sie trägt die Entscheidung welcher Suchalgorithmus für das Problem der passendste ist.</p>
Unterrichtswoche		Modul 2 Woche 8 (teilweise), 9 (teilweise)	
Methoden		Frontalunterricht, fragend-entwickelnd, Gruppenarbeit, Einzelarbeit	
Anmerkung		Theoretisches Wissen in Gruppen in Anlehnung an Analytik durchführen. Unklarheiten im Frontalunterricht, fragend-entwickelnd auflösen. Prak-	

	tische Umsetzung aller Teilnehmer/innen selbstständig als Einzelarbeit.		
Unit 11	Sortieralgorithmen		
	Er/Sie ist in der Lage diverse Sortieralgorithmen zu erklären und implementieren.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Funktionsweise des Bubblesort.	Er/Sie kann den Bubblesort implementieren.	Er/Sie entscheidet über den einzusetzenden Sortieralgorithmus.
	Er/Sie kennt die Funktionsweise des Quicksort.	Er/Sie kann den Quicksort implementieren.	
	Er/Sie kennt Methoden des .NET Frameworks zum Sortieren.	Er/Sie kann Sortiermethoden des .NET Frameworks einsetzen.	
Unterrichtswoche	Modul 2 Woche 9 (teilweise), 10 (teilweise)		
Methoden	Frontalunterricht, fragend-entwickelnd, Gruppenarbeit, Einzelarbeit, Feedback		
Anmerkung	Theoretisches Wissen in Gruppen in Anlehnung an Analytik durchführen. Unklarheiten im Frontalunterricht, fragend-entwickelnd auflösen. Praktische Umsetzung aller Teilnehmer/innen selbstständig als Einzelarbeit. Schriftlicher Test am Ende des Moduls		

**Tabelle 9: Lehreinheiten objektorientierte Gruppe Module 1 und 2**

### 4.7.2 Benutzeroberfläche – Modul 3

Im dritten Modul sollen die erworbenen Kenntnisse, Fähigkeiten und Kompetenzen aus den Modulen 1 und 2 angewendet werden, indem mit Hilfe dieses Basiswissens kleine Applikationen mit Oberflächen programmiert werden. In diesem Modul werden das erste Mal Unterschiede bei den Ergebnissen der Lehrmethoden sichtbar und die Trainer/innen sind angehalten diese zu dokumentieren. Die Erkenntnisse der Beobachtungen dieses Moduls findet man in dem Kapitel 5.9 Überprüfung der Hypothese und dem Kapitel 6 Schlussfolgerungen.

Das objektorientierte Denken sollte in diesem Modul im Idealfall schon bei den Teilnehmer/innen vorhanden sein und somit liegt der Fokus auf dem Wiederholen dieser Grundlagen und Einsetzen dieses Wissens um neue Themengebiete leichter verstehen zu können.



Da die Module 1 und 2 sich damit beschäftigen, ein objektorientiertes Denkmuster zu entwickeln, um in eine neue Welt – die Welt des objektorientierten Programmierens – eintauchen zu können, bietet das Modul 3 die Möglichkeit dieses Wissen einwirken zu lassen. Neuer Unterrichtsstoff wird nur mehr objektorientiert vermittelt. Somit wird vermehrt auf bereits Vermitteltes auf eine andere Art und Weise zurückgegriffen und trainiert.

Alle nachfolgenden Lehreinheiten der Module 3 bis 8 werden objektorientiert unterrichtet und bauen auf die Module 1 und 2 auf. Diese Module sind für die Evaluierung wichtig, damit festgestellt werden kann, welcher Programmieransatz längerfristig der effektivere für eine erfolgreiche objektorientierte Wissensvermittlung ist.

### Themen

- Extensible Application Markup Language (XAML)  
Unterliegt den gleichen Regeln wie XML, welche die Teilnehmer/innen schon im Unterrichtsfach Webprogrammieren erlernt haben.
- Eventhandling  
Events sind ein neues Themengebiet, hierbei sollen aber nur die nötigsten Grundlagen verwendet werden, z.B. Klick auf einen Button
- Datenspeicherung  
Hier geht es darum objektorientiertes Wissen einzusetzen, indem man schon vorhandene Klassen konstruktiv verwendet.
- Eventuell Datenbindung  
Soll nur mit begabten Klassen und bei entsprechender Zeit in der Tiefe behandelt werden, da dies ein komplexes Thema ist.

Unit 1	Grafische Oberflächen im .NET Framework		
	Er/Sie ist in der Lage den Begriff GUI zu erklären und kann verschiedene Technologien des .NET Frameworks mit ihren Vor- und Nachteilen für die Erstellung einer GUI auflisten.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt den Begriff GUI und kann diesen erklären.</p> <p>Er/Sie kennt die wichtigsten Rahmenbedingungen die für ein GUI einzuhalten sind.</p> <p>Er/Sie kennt Technologien um ein GUI mit Hilfe des .NET Frameworks zu erstellen.</p> <p>Er/Sie kennt grundlegende Steuerelemente von Oberflächen.</p>	<p>Er/Sie kann eine einfache grafische Oberfläche mittels Drag und Drop und WPF erstellen.</p> <p>Er/Sie kann eine einfache grafische Oberfläche mittels Drag und Drop und Windows Forms erstellen.</p>	<p>Er/Sie berät über die passendste Lösung für eine Oberfläche.</p>
Unterrichtswoche	Modul 3 Woche 1		
Methoden	Frontalunterricht, fragend-entwickelnd, Einzelarbeit		
Anmerkung	Spielerischer Umgang mit Hilfe der IDE um eine einfache Oberfläche mittels Drag und Drop zu erzeugen. Bei fortgeschrittenen Klassen kann ein Fokus auf XAML forciert werden und somit die nächste Unit vorgezogen werden.		
Unit 2	XAML		
	Er/Sie ist in der Lage eine ansprechende Oberfläche mittels XAML zu erstellen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt den Begriff XAML und die Vor- bzw. Nachteile dieser Technologie.</p> <p>Er/Sie kennt den Unterschied zwischen Layout Elementen und Inhalts-</p>	<p>Er/Sie kann eine einfach strukturierte Oberfläche mittels XAML erstellen.</p> <p>Er/Sie kann verschiedene Layouts mittels XAML erstellen.</p>	<p>Er/Sie prüft das Design der Oberfläche.</p> <p>Er/Sie entscheidet die Wahl der eingesetzten Steuerelemente.</p>

	<p>elementen,</p> <p>Er/Sie kennt die wichtigsten Klassen von WPF.</p> <p>Er/Sie kann den Begriff Datenbindung erklären.</p>	<p>Er/Sie kann einfache Steuerelemente erstellen.</p> <p>Er/Sie kann Containerelemente mittels XAML erstellen und befüllen.</p> <p>Er/Sie kann einfache Formatierungen für ein Element festlegen.</p> <p>Er/Sie kann einfache Formatierungen für ein ganzes Projekt festlegen.</p> <p>Er/Sie kann eine Datenbindung mittels XAML erstellen.</p> <p>Er/Sie kann ein Datentemplate erstellen.</p>	
Unterrichtswoche		Modul 3 Woche 2, 3, 4	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit	
Anmerkung		Spielerischer Zugang mit sehr visuellen Vorgaben. Datentemplates können bei Bedarf sehr reduziert oder ausgebaut werden.	
Unit 3	Zusammenspiel Design und Code-Behind		
	Er/Sie ist in der Lage auf Eingaben der Oberfläche zu reagieren und Daten für die Oberfläche aufzubereiten.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt den Ablauf der Ereignisbehandlung.</p> <p>Er/Sie kennt die Bedeutung von Schnittstellen zwischen Oberfläche und Code-Behind.</p>	<p>Er/Sie kann Klick-Ereignisse behandeln.</p> <p>Er/Sie kann weitere Ereignisse, wie z.B. Drag und Drop implementieren.</p>	<p>Er/Sie definiert die Schnittstellen für die Kommunikation.</p>

		<p>Er/Sie kann mehrere Ereignisse, einer Methode zuordnen.</p> <p>Er/Sie kann Schnittstellen definieren und anwenden um Daten zu übergeben.</p> <p>Er/Sie kann Schnittstellen definieren und anwenden um Daten anzuzeigen.</p> <p>Er/Sie kann Daten aus einer Oberfläche an eine andere Oberfläche Übergeben.</p> <p>Er/Sie kann eine Datenbindung im Code-Behind erstellen.</p>	
Unterrichtswoche		Modul 3 Woche 5, 6	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Blitzlicht	
Anmerkung		<p>Weitere Ereignisse sind dem/der Trainer/in freigestellt und hängen vom fortschrittsgrad der Gruppe sowie zukünftigen Projekten ab.</p> <p>In der fünften Modulwoche soll der/die Trainer/in eine Erhebung des Wissenstands machen</p>	
Unit 4	Validierung		
	Er/Sie ist in der Lage Dateneingaben auf Korrektheit zu validieren.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt die Bedeutung einer Validierung.</p> <p>Er/Sie kennt verschiedene Möglichkeiten Validierung durchzuführen und kann ihre Vor- und Nachteile aufzählen.</p>	<p>Er/Sie kann Validierung in Eigenschaften implementieren.</p> <p>Er/Sie kann Validierungsklassen im Code-Behind erstellen und anwenden.</p> <p>Er/Sie kann Validie-</p>	<p>Er/Sie entscheidet über den Aufbau einer Validierungsklasse.</p> <p>Er/Sie überprüft verwendete Validierungsregeln.</p>

		rungsklassen in XAML zuweisen.  Er/Sie kann eine Validierungsbibliothek erstellen und anwenden.	
Unterrichtswoche	Modul 3 Woche 7, 8		
Methoden	Frontalunterricht, fragend-entwickelnd, Partnerarbeit, Einzelarbeit		
Anmerkung	Validierung an verschiedenen Orten vorzeigen und die Vor- und Nachteile aufzeigen, z.B. Validierung nur in der Oberfläche oder nur in der Klasse.		
Unit 5	Speicherung		
	Er/Sie ist in der Lage einfache und komplexe Datentypen in Dateien zu speichern und auszulesen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die wichtigsten Klassen um mit Datenströmen zu arbeiten.  Er/Sie kennt die Vor- und Nachteile Daten binär oder textuell zu speichern.	Er/Sie kann einfache Daten als Text speichern und laden.  Er/Sie kann komplexe Datentypen speichern und laden.  Er/Sie kann XML-Dateien durchlaufen.	Er/Sie entscheidet über den Aufbau eines Datentyps.
Unterrichtswoche	Modul 3 Woche 9, 10		
Methoden	Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Feedback		
Anmerkung	Schriftlicher Test am Ende des Moduls		

**Tabelle 10: Lehreinheiten UI Modul 3**

### 4.7.3 Aktives Kompetenz Training – Module 4 und 5

Die Module 4 und 5 bringen für die Teilnehmer/innen, eine stärkere Verschiebung vom Frontalunterricht in Richtung Einzel-, Partner-, Gruppen- und Projektarbeit. Die Trainer/innen ziehen sich mehr aus dem unterrichtenden Bereich zurück und übernehmen immer mehr die Rolle von Berater/innen.

Die Lehreinheiten, die für AKT eingeteilt sind, entsprechen dem Gedanken der Projektarbeit. Hierbei erstellt die Gruppe ein Produkt für Kund/innen, von der Planung bis zur Verkaufspräsentation.

Die übrigen PR-Einheiten beschäftigen sich hauptsächlich mit Themen, die für die Umsetzung des AKT-Projekts von Nöten sind.

Diese Themen werden teilweise in Form von Referaten von den Teilnehmer/innen behandelt. Diese Referate werden entweder als Einzelreferate oder Gruppenreferate abgehalten.

Diese Module bieten den Trainer/innen aber auch die erste Möglichkeit, Differenzen zwischen dem prozeduralen vs. objektorientierten Lehransatz an einem selbst umgesetzten praktischen Beispiel zu entdecken.

Die hier vorgestellten Lehreinheiten beziehen sich auf das Fach PR und nicht AKT/PR, da nur im Fach PR unterrichtet wird und AKT/PR eine eigenständige Projektarbeit der ganzen Klasse ist.

Unit 1	Technische Dokumentation		
	Er/Sie ist in der Lage eine technische Dokumentation zu erstellen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Bedeutung einer technischen Dokumentation.  Er/Sie kennt Tools zur automatischen Erstellung einer technischen Dokumentation.	Er/Sie kann mit Hilfe eines Tools eine technische Dokumentation erstellen.	Er/Sie entscheidet über das Tool zur Erstellung einer technischen Dokumentation.
Unterrichtswoche	Modul 4 Woche 1		
Methoden	Gruppenarbeit, Referat, Feedback		
Anmerkung	Unterteilung der Klasse in mehrere Gruppen, wobei eine Gruppe die Bedeutung behandeln soll und jede weitere Gruppe ein Tool für die techn. Dokumentation vorstellen soll.		
Unit 2	Datenspeicherung ADO.NET und EF		
	Er/Sie ist in der Lage mittels ADO.NET Daten in einer Datenbank zu verwalten.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt den Begriff ADO.NET und kann diesen erklären.  Er/Sie kennt den Begriff Entity Framework und kann diesen erklä-	Er/Sie kann mittels ADO.NET Daten abfragen, anzeigen und manipulieren.  Er/Sie kann mittels ADO.NET EF Daten	Er/Sie entscheidet über die Technologie für den Datenbankzugriff.

	<p>ren.</p> <p>Er/Sie kennt den Begriff Entität.</p> <p>Er/Sie kennt den Begriff Model.</p> <p>Er/Sie kennt den Unterschied zwischen Code-first vs. Database-first vs. Model-first.</p>	<p>abfragen, anzeigen und manipulieren.</p> <p>Er/Sie kann aus einer Datenbank ein Entity-Data-Model generieren lassen.</p> <p>Er/Sie kann aus Klassen eine Datenbank mittels EF generieren lassen.</p>	
Unterrichtswoche		Modul 4 Wochen 2 - 10	
Methoden		Gruppenarbeit, Referat, Feedback, Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Blitzlicht	
Anmerkung		<p>Die Themen ADO.NET und EF werden auf Gruppen aufgeteilt, die eine Einführung in die Themen präsentieren sollen. Tieferen Einblick sollen die Trainer/innen liefern und zusätzliche Einzelübungen verteilen.</p> <p>In der fünften und letzten Modulwoche sollen die Trainer/innen eine Erhebung des Wissensstands machen.</p> <p>In der letzten Modulwoche gibt es keinen schriftlichen Test, da die Teilnehmer/innen ihr Wissen anhand der Skillcard präsentieren.</p>	
Unit 3	Einführung MVC		
	Er/Sie ist in der Lage eine einfache Website für eine Onlinepräsenz mittels ASP.NET MVC zu erstellen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt den Unterschied zwischen einer statischen und dynamischen Webseite.</p> <p>Er/Sie kennt alternative Technologien um dynamische Webseiten zu erstellen.</p> <p>Er/Sie kennt die Be-</p>	<p>Er/Sie kann ein MVC Projekt erstellen und dessen Struktur erklären.</p> <p>Er/Sie kann einen Controller erstellen und dessen Bedeutung erklären.</p> <p>Er/Sie kann eine View erstellen und dessen</p>	<p>Er/Sie entscheidet über die eingesetzte Webtechnologie.</p> <p>Er/Sie legt die Datenmodelle fest.</p>

	deutung von MVC und dessen Vor- und Nachteile.	Bedeutung erklären.  Er/Sie kann ein Model erstellen und dessen Bedeutung erklären.	
Unterrichtswoche		Modul 5 Woche 1-8	
Methoden		Frontalunterricht, fragend-entwickelnd, Einzelarbeit, Partnerarbeit, Blitzlicht	
Anmerkung		Ausschnitte von MVC, wie z.B. die Validierung sollen von Teilnehmer/innen in einer Partnerarbeit erarbeitet werden.  In der fünften Modulwoche sollen die Trainer/innen eine Erhebung des Wissenstands vornehmen.	
Unit 4	Lizensierung		
	Er/Sie ist in der gängige Lizenzformen in der Softwareentwicklung einordnen zu können.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt verschiedene Lizenzformen mit ihren Vor- und Nachteilen.	Er/Sie kann eine Lizenzform für ein Softwareprojekt bestimmen.	Er/Sie entscheidet über die einzusetzende Lizenzform.
Unterrichtswoche		Modul 4 Woche 8	
Methoden		Gruppenarbeit, Referat, Feedback	
Anmerkung		Teilnehmer/innen sollen in Gruppen verschiedene Lizenzmodelle kurz präsentieren.	
Unit 5	Setuproutinen		
	Er/Sie ist in der Lage ein Setupprojekt zu erstellen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt die Bedeutung von Setupprojekten.  Er/Sie kennt verschiedene Tools um Setupprojekte zu erstellen.	Er/Sie kann ein Setupprojekt erstellen.  Er/Sie kann ein Setupprojekt für Updates erstellen.	Er/Sie entscheidet über das Tool zur Erstellung von Setupprojekten.
Unterrichtswoche		Modul 4 Woche 9	
Methoden		Partnerarbeit	
Anmerkung		Teilnehmer/innen sollen für ein vorhandenes Softwareprojekt – kann auch das AKT-Projekt sein – eine Setuproutine in Partnerarbeit erarbei-	



	<p>ten.</p> <p>In der letzten Modulwoche gibt es keine schriftliche Prüfung, da diese Woche als Vorbereitung für die Präsentation des AKT-Projekts und die Fachkraftprüfung verwendet wird. Die Fachkraftprüfung soll eine theoretische Prüfung in Form eines Fachgespräches simulieren.</p>
--	--

**Tabelle 11: Lehreinheiten AKT Module 4 und 5**

#### 4.7.4 Lehrabschluss – Module 7 und 8

In diesen Modulen werden nur mehr Themen in Bezug auf die praktische Lehrabschlussprüfung behandelt. Die Teilnehmer/innen müssen hierfür eine Webanwendung programmieren. Hierbei greifen diese auf das erlangte Wissen der Fächer Webprogrammieren und Programmieren zurück.

Die Programmierereinheiten werden gerade im Modul 7 dafür verwendet, unzureichend behandelte Themen mit den Teilnehmern/innen zu behandeln und noch offene Fragen bezüglich des Lehrabschlussprojekts zu klären.

Das Modul 8 beinhaltet zumeist nur mehr wenige Fragen bezüglich des Lehrabschlussprojektes, sondern beschäftigt sich eher mit einer Wiederholung der Fragen aus den ersten Modulen, den Grundlagenmodulen 1 und 2, da das dort erlangte theoretische und praktische Wissen die Grundlage für die Lehrabschlussprüfung ist.

Außerdem haben auch hier die Trainer/innen wieder eine Möglichkeit an einem praktischen Beispiel Differenzen in Folge des prozeduralen und objektorientierten Lehransatzes zu erkennen. Auch das Praktikum kann hier eine Rolle spielen, wo die Teilnehmer/innen ihr programmiertechnisches Knowhow womöglich verbessert haben. Im Detail wird diese Beobachtung bei der Evaluierung beschrieben.

Unit 1	Vertiefung MVC		
	Er/Sie ist in der Lage eine komplexe dynamische Website mit Sicherheitsmerkmalen zu erstellen.		
	Kenntnisse	Fertigkeiten	Kompetenz
	Er/Sie kennt Vor- und Nachteile eines Onlineshops.	Er/Sie kann eine Authentifizierung erstellen.	Er/Sie entscheidet über den Authentifizierungsmodus.
	Er/Sie kennt die gängigsten Angriffstypen auf eine Webseite.	Er/Sie kann eine Validierung anhand des Models einbauen.	Er/Sie prüft die Einstellungen für einen Webserver in Bezug auf Sicherheit.
	Er/Sie kennt Möglichkeiten eine Webseite	Er/Sie kann asynchrone Anfragen mittels	

	<p>responsive zu gestalten.</p> <p>Er/Sie kennt Möglichkeiten Daten zu validieren.</p> <p>Er/Sie kennt Möglichkeiten Daten asynchron zu verschicken.</p>	<p>ASP.NET MVC stellen und verarbeiten.</p> <p>Er/Sie kann ein responsive Layout anwenden.</p> <p>Er/Sie kann eine Datenbank als Quelle für dynamische Inhalte verwenden.</p>	
Unterrichtswoche		Modul 7 Wochen 1 - 5	
Methoden		Frontalunterricht, fragend-entwickelnd, Partnerarbeit, Projektarbeit	
Anmerkung		Die Trainer/innen stellen nur mehr die wichtigsten Teile im Frontalunterricht vor, alle Erweiterungen zum Themengebiet sind in Partnerarbeit zu behandeln. Sobald die Grundlagen klar sind, wird das Lehrabschlussprojekt begonnen.	
Unit 2	JavaScript Advanced		
	Er/Sie ist in der Lage mittels JavaScript eine ASP.NET MVC Anwendung dynamischer zu steuern.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt Vorteile fertiger JavaScript-Bibliotheken.</p> <p>Er/Sie kennt JavaScript-Bibliotheken für dynamische Steuerelemente.</p> <p>Er/Sie kennt JavaScript-Bibliotheken zur Unterstützung der Kommunikation mit ASP.NET MVC.</p>	<p>Er/Sie kann fremde JavaScript-Bibliotheken in ein MVC Projekt einbinden.</p> <p>Er/Sie kann Steuerelemente mit Hilfe von JavaScript benutzerfreundlicher arbeiten lassen.</p> <p>Er/Sie kann komplexe Validierungen mittels JavaScript und ASP.NET MVC steuern.</p> <p>Er/Sie kann dynamische Formulare mittels JavaScript erstellen</p>	<p>Er/Sie entscheidet über die Wahl der JavaScript-Bibliotheken und ihre Notwendigkeit.</p>

		und mittels MVC auswerten lassen.	
Unterrichtswoche		Modul 7 Wochen 6 – 10	
Methoden		Frontalunterricht, fragend-entwickelnd, Projektarbeit	
Anmerkung		Die Trainer/innen stellen nur mehr die wichtigsten Teile im Frontalunterricht vor, ansonsten wird am Lehrabschlussprojekt selbstständig weitergearbeitet.	
Unit 3	Gestaltung		
	Er/Sie ist in der Lage mittels CSS eine ASP.NET MVC Anwendung dynamischer zu gestalten.		
	Kenntnisse	Fertigkeiten	Kompetenz
	<p>Er/Sie kennt Vor- und Nachteile von CSS-Templates.</p> <p>Er/Sie kennt die Ordnerstruktur für ein CSS-Template.</p> <p>Er/Sie kennt CSS-Vorschriften für ein ASP.NET MVC Projekt.</p>	<p>Er/Sie kann ein fremdes CSS-Template in ein MVC Projekt einbinden.</p> <p>Er/Sie kann Steuerelemente mit Hilfe eines Templates gestalten.</p> <p>Er/Sie kann komplexe Webseiten mit Hilfe eines Templates gestalten.</p> <p>Er/Sie kann ein komplexes CSS-Template an die eigenen Bedürfnisse anpassen.</p>	<p>Er/Sie entscheidet über die Wahl des CSS-Templates und dessen Gestaltung.</p>
Unterrichtswoche		Modul 8 Wochen 1 – 3	
Methoden		Frontalunterricht, fragend-entwickelnd, Projektarbeit	
Anmerkung		Die Trainer/innen stellen nur mehr die wichtigsten Teile im Frontalunterricht vor, ansonsten wird am Lehrabschlussprojekt selbstständig weitergearbeitet.	
Unit 4	Projektabschluss		
	Er/Sie ist in der Lage ein ASP.NET MVC Webprojekt zu einem Abschluss zu bringen.		
	Kenntnisse	Fertigkeiten	Kompetenz

	<p>Er/Sie kennt die Wichtigkeit eines Projektabschlusses.</p> <p>Er/Sie die Vorgehensweise bei einem Projektabschluss.</p>	<p>Er/Sie kann alle nötigen Schritte für einen Projektabschluss durchführen.</p>	<p>Er/Sie entscheidet über die Vorgehensweise bei einem Projektabschluss.</p>
Unterrichtswoche		Modul 8 Wochen 4 – 5	
Methoden		Frontalunterricht, fragend-entwickelnd, Projektarbeit	
Anmerkung		<p>Die Trainer/innen stellen nur mehr die wichtigsten Teile im Frontalunterricht vor, ansonsten wird am Lehrabschlussprojekt selbstständig weitergearbeitet.</p> <p>Die restlichen Wochen können individuell gestaltet werden, da das Ende der Ausbildung nicht für jede Klasse gleich ist, müssen diese Lehreinheiten dynamisch angepasst werden, z.B. können vorhergehende Units bei länger geplanten Kurszeiten weiter ausgeschmückt werden oder Themen für die Lehrabschlussprüfung besprochen werden.</p>	

**Tabelle 12: Lehreinheiten LAP Module 7 und 8**

## 5 Evaluierung

### 5.1 Definitionen

Das Ziel dieser Diplomarbeit ist eine Gegenüberstellung des prozeduralen vs. objekt-orientierten Lehransatzes im Rahmen der Ausbildung Informatiker/innen mit Lehrabschluss am zweiten Bildungsweg für Erwachsene mit Reha-Hintergrund. Die primären Ziele für die Ausbildung sind, die Teilnehmer/innen durch die Lehrabschlussprüfung Informatik zu führen, sowie eine Basis für einen beruflichen Wiedereinstieg zu schaffen. Die Herausforderung hierbei ist, dass den Teilnehmer/innen nur ein begrenzter Zeitraum für die Ausbildung zur Verfügung steht und dadurch die Lehrmethode Auswirkungen auf das Verhältnis zwischen Überforderung und benötigtem Wissen beeinflusst.

Die verschiedenen Lehrmethoden, die für den Kurs angewendet werden, und ihre Auswirkungen in den restlichen Modulen zeigen, entscheiden maßgeblich die Arbeitsweise der Teilnehmer/innen. Werden für die Zielgruppe und die Themengebiete passende Lehrmethoden verwendet, so können entlastende Effekte auftreten, indem Interesse an der Programmierung positiv beeinflusst wird.

Damit hierbei die bessere Methode festgestellt werden kann, wird zuerst das Unterrichtskonzept aus Kapitel 4 mit den geplanten Lehreinheiten aus Kapitel 4.7.1 Grundlagen – Module 1 und 2 an der Zielgruppe angewandt und auf Unterschiede, sowie auf Gemeinsamkeiten überprüft. Hierbei werden besondere Beobachtungen herausgehoben und im Kapitel 5.5 Beobachtungsergebnisse Kursverlauf festgehalten.

Diese Überprüfung findet im Rahmen einer Evaluation statt. Die Evaluierung hat laut Duden<sup>10</sup> folgende Synonyme: Beurteilung, Bewertung, Wertung. Daraus ergibt sich, dass in diesem Kapitel die Beurteilung der verschiedenen Lehransätze im Fokus auf ihre Effizienz steht. Diese Beurteilung wird laut (Wulf, 1982) folgend beschrieben: *„Evaluation erfolgte in einer fortwährenden Überprüfung der Realisierung von Intentionen bzw. normativen Kriterien und als Entwicklung von Alternativen auf Grund unbefriedigender Ergebnisse, wobei die Modifikationen sich häufig auf die Mittel der Realisierung, oft genug aber auch auf die Intentionen selbst bezogen.“*

Hieraus ergibt sich für diese Diplomarbeit der Ansatz, dass die Evaluation ständig erfolgt und nicht erst am Ende des Kurses. Sowie auch eine kritische Hinterfragung der eigentlichen Ziele, weshalb überhaupt eine Evaluation der verschiedenen Lehransätze von Nöten ist.

---

<sup>10</sup> <http://www.duden.de/rechtschreibung/Evaluierung>

## **5.2 Die zehn Empfehlungen**

Aufgrund der Definition von Evaluation kommt man schnell zu dem Entschluss, dass Evaluation geplant sein soll. Für diese Planung wird das Kapitel *Tipps für den Erfolg – Auf gutes Gelingen: zehn Empfehlungen* aus dem Buch *Praxishandbuch - Evaluation in der Schule* (Burkard und Eikenbusch, 2000) herangezogen. Hierbei handelt es sich zwar um ein Buch für den schulischen Bereich, doch in Hinsicht auf die Evaluation, und die hierbei wichtigen Module 1 und 2, trifft dieses sehr gut zu. Folgend werden die zehn Empfehlungen auf das Bildungsinstitut und den Kurs angepasst vorgestellt, um häufige Fehler in der Durchführung der Evaluation zu vermeiden.

### **5.2.1 Nicht unbedingt mit allen, aber auch nicht gegen viele beginnen**

Im Buch von (Burkard und Eikenbusch, 2000) wird von einem Gegeneinander gesprochen. Dies ist im Bildungsinstitut nicht der Fall. Da die Programmiertrainer/innen schon seit längerer Zeit über einen Wechsel eines prozeduralen zu einem objektorientierten Lehransatzes nachdenken, wurde die Idee der Evaluierung dieser beiden Lehransätze sehr wohlwollend angenommen und die Trainer/innen, die die Evaluierung durchführten, unterstützt.

### **5.2.2 Normen und Spielregeln für die Durchführung klären**

Bei der Planung der Evaluierung werden alle Programmiertrainer/innen im vorab aufgeklärt wie diese erfolgen wird. Außerdem haben alle Programmiertrainer/innen das Recht ihre eigenen Ideen einzubringen. Bei unterschiedlichen Meinungen wird demokratisch abgestimmt. Daten und Ergebnisse der Evaluierung werden außerdem allen Programmiertrainer/innen mitgeteilt und beeinflussen die Planung des nächsten Lehrplanes. Bei Supplierungen übernehmen die supplierenden Trainier/innen alle Pflichten der evaluierenden Trainer/innen.

### **5.2.3 Ziele der Evaluation klären**

Auf die Frage *Was bringt uns das?* (Burkard und Eikenbusch, 2000) wird folgend geantwortet: Die Ergebnisse der Evaluierung beeinflussen die nächste Planung des Lehrplans für das Unterrichtsfach Programmieren im Rahmen der Informatikausbildung des Bildungsinstituts. Somit beeinflusst diese Beurteilung direkt die Ausbildung zukünftiger Klassen.

### **5.2.4 Schritt für Schritt vorgehen**

Ganz in dem Sinn, dass die Evaluation keine *eierlegende Wollmilchsau* sein soll, ist die Evaluierung auf folgende Fragestellungen konzentriert:

- Welcher Lehransatz ist effektiver in der Wissensvermittlung für die Lehrabschlussprüfung?
- Welcher Lehransatz ist effektiver in der Wissensvermittlung für die Praxis?
- Welcher Lehransatz weckt stärker das Interesse an der Informatik?

### **5.2.5 Wichtige und relevante Themen auswählen**

Warum für die Arbeit ein Vergleich zwischen einem prozeduralen und objektorientierten Lehransatz gewählt wurde ergibt sich daraus, dass sich die Programmiertrainer/innen schon seit längerer Zeit überlegt haben, dass Aufgrund der zeitlichen Beschränkung der Kursdauer und der beruflichen Anforderungen am Arbeitsmarkt eine verstärkt objektorientierte Lehrweise zielführender für die Ausbildung wäre. Außerdem könnte dadurch womöglich Druck von den Teilnehmer/innen genommen werden, indem gleich von Anfang an objektorientierte Konzepte vermittelt werden, und nicht nur in einem Modul zusammengefasst werden. Hierbei dürfen aber die prozeduralen Grundlagen, die für die Lehrabschlussprüfung genauso nötig sind, nicht zu sehr vernachlässigt werden.

### **5.2.6 Engen Bezug zur Arbeitsplanung herstellen**

Die Planung und Durchführung der Evaluation wird hauptsächlich im Rahmen dieser Diplomarbeit gestaltet. Alle Entscheidungen werden aber allen Programmiertrainer/innen vorgelegt und diese haben eine gleichberechtigte Stimme um Einwände, Ergänzungen oder Veränderungen zu beeinflussen. Hierfür gibt es Teamsitzungen, wo die Planung, der Fortschritt in der Durchführung und die Ergebnisse besprochen werden.

### **5.2.7 Evaluationsinstrumente maßschneidern, aber verfügbare Beispiele nutzen**

Die Evaluationsinstrumente werden im Kapitel 5.3 Untersuchungsmethoden der Evaluation näher beschrieben.

### **5.2.8 Unterschiedliche Sichtweisen einbeziehen**

Da wichtige Grundsätze, wie z.B. die Pläne für die Durchführung, einerseits dem Programmiererteam vorgestellt werden, sowie diese ein Mitspracherecht haben, ist die Gefahr der einseitigen Sichtweise eher nicht gegeben.

### **5.2.9 Ergebnisse, Bewertungen und Beschlüsse dokumentieren**

Die Dokumentation erfolgt im Zuge dieser Arbeit und ist somit für alle offen zur Einsicht. Außerdem haben die Programmiertrainer/innen im Unternehmen jederzeit Zugriff auf die aktuellen Ergebnisse der erhobenen Daten und Bewertungen.

### **5.2.10 Erfahrungen mit der Durchführung von Evaluation gemeinsam auswerten**

Die Evaluation an sich wird im Rahmen von Teamsitzungen der Programmiertrainer/innen besprochen, damit zukünftige Evaluationen von diesen Erfahrungen profitieren können. Im Rahmen dieser Arbeit wird die Evaluierung und deren Sinnhaftigkeit im Kapitel 6 Schlussfolgerungen kritisch behandelt.

## **5.3 Untersuchungsmethoden der Evaluation**

Damit die Evaluierung nachvollziehbar und reproduzierbar ist, werden im folgenden Kapitel die verwendeten Methoden für die Informationsgewinnung vorgestellt.

### **5.3.1 Fragebogen**

Eine Aufgabe des Fragebogens ist es, die Gruppenzusammenstellung zu ermitteln. Es werden im Fragebogen Fragen bezüglich schon vorhandener Computerkenntnisse und Programmierkenntnisse gestellt.

Ein weiteres Ziel des Fragebogens ist zu ermitteln, welche Medien, Unterrichtsstile und Themen bei den Teilnehmer/innen angewendet wurden und welchen Eindruck diese auf die Teilnehmer/innen machten. Diese Daten werden auch im Besonderen mit den Daten der Trainer/innen, die die aktive Beobachtung durchführten, verglichen, sowie auch mit dem Unterrichtskonzept.

Am Ende des Fragebogens geben die Teilnehmer/innen eine Selbstabschätzung in Bezug auf ihre weitere Informatiklaufbahn ab, z.B. ob man sich selbst in der Rolle von Programmierer/innen sieht und wie man seine Jobaussichten einschätzt. Hierbei kann man auch die Einstellung ablesen, ob jemand im IT-Sektor beschäftigt werden



möchte, und hiervon wiederum ein entsprechendes Interesse für die Informatik ableiten.

### **5.3.2 Analyse Kursunterlagen inkl. Verwaltungsunterlagen**

Bei der Analyse der Kursunterlagen werden die festgehaltenen Leistungsnachweise während der Ausbildung herangezogen. Gerade zum Kursende hin soll man dadurch feststellen können, ob beide Lehrmethoden zu unterschiedlichen Ergebnissen führten oder ob das Endresultat trotz unterschiedlicher Wege doch das Gleiche ist. Als Leistungsnachweise gelten Wiederholungen und Tests.

Die Verwaltungsunterlagen werden hauptsächlich zur Analyse der Gruppendynamik und sonstigen Einflussfaktoren verwendet. Da es vorkommen kann, dass Teilnehmer/innen während der Ausbildung kurz aussetzen und wiedereinsteigen, kann es Änderungen in der Gruppe geben. Diese Änderungen in der Gruppe während der Ausbildung beeinflussen auch die Qualität dieser und müssen deswegen auch schriftlich festgehalten werden. Außerdem kann die Gruppenzusammensetzung, sowie die Gruppengröße auch das Leistungsniveau beeinflussen.

### **5.3.3 Aktive Beobachtung**

Hierbei wird begleitend und nach dem Unterricht kurz festgehalten, wie die Gruppe auf verschiedene Unterrichtsstile und Unterrichtsmethoden reagiert – z.B. wie sie auf englische Fachliteratur oder Video-Tutorials reagiert, was für Unterrichtsstoff positiv oder negativ aufgenommen wurde oder wo die meisten Hürden beim Bewältigen einer Aufgabenstellung waren. Diese Daten sind wichtig, damit die Gruppen miteinander verglichen werden können, und auch eine Aussage getroffen werden kann, ob eine Gruppe effektiver (bezüglich Geschwindigkeit und Verständnis) den Lehrinhalt aufgenommen hat als die andere Gruppe. Dazu wird eine Art Tagebuch in Form von Unterrichtsnotizen – siehe Kapitel 5.5.1 – angelegt, um den doch relativ langen Ausbildungszeitraum ausführlich zu dokumentieren. Die Beobachtung findet durch die Trainer/innen selbst statt. Hierbei wurden die Klasse, die Ausbildungswoche (Modul/Modulwoche), Unterrichtsthemen und Besonderheiten im Unterricht festgehalten. Diese Daten benötigt man, um die Praxiserprobung mit dem theoretischen Unterrichtsplan in Verbindung zu bringen.

### **5.3.4 Eigene Unterrichtserfahrung**

Hier wird festgehalten, wie sich anhand der eigenen Unterrichtserfahrung die Gruppe oder einzelne Personen im Vergleich zu anderen verhalten. Es wird außerdem festgehalten, ob es besondere Auffälligkeiten innerhalb der Gruppe gibt, die für die Eva-

luierung beachtet werden sollten. Zusätzlich wird festgehalten, ob sich das Unterrichtskonzept im Unterricht bewährt und ob es auffällige Vor- oder Nachteile im Vergleich zum bisherigen Unterrichtsstil gegeben hat.

## **5.4 Praktische Umsetzung**

Die praktische Umsetzung ist für jeweils zwei Klassen pro Lehransatz geplant. Das ergibt sich daraus, da mit zwei Klassen pro Lehransatz die gruppenspezifischen Auffälligkeiten abgeschwächer auftreten können als bei einer Klasse. Außerdem sind die Klassengrößen sehr variabel, wodurch es bei nur einer Klasse pro Lehransatz zu einer zu kleinen Gruppengröße kommen könnte, was bei einer Evaluierung mittels der eingesetzten Methoden die Ergebnisse zu sehr beeinflussen würde. Durch diese Festlegung von zwei Klassen pro Lehransatz ist gewährleistet, dass zumindest 15 Teilnehmer/innen pro Lehransatz erfasst werden können. Die Klassen können theoretisch 5 bis 20 Teilnehmer/innen fassen, doch sind in der Regel 8 Teilnehmer pro Klasse vertreten. Bei der Auswertung der Daten wird nicht nach Klassen, sondern nach Lehransatz unterschieden.

Alle Teilnehmer/innen sind erwachsene Personen im Alter von 18 bis 55 Jahre. Sie befinden sich am zweiten Bildungsweg, wobei die Vorkenntnisse bezüglich allgemeines Verständnis von Computern, der Informatik und der Programmierung im Speziellen von geringen bis sehr guten Kenntnissen reichen kann. Genauere Informationen bezüglich technischer Vorkenntnisse finden sich in Kapitel 4.5 Vorkenntnisse.

Der Fragebogen wird am Ende des Moduls 2, d.h. direkt nach dem Kurs, und am Ende des Moduls 8, d.h. am Ende der Ausbildung, ausgeteilt und für die Auswertung eingesammelt.

Alle relevanten Kursunterlagen werden während der Kurszeit gesammelt und anonymisiert festgehalten, sowie nach dem Kurs für die Evaluation ausgewertet. Die aktive Beobachtung findet hierbei während des Unterrichts statt und alle Unterrichtsnotizen werden zu der entsprechenden Gruppe gesammelt verwahrt. Die eigene Unterrichtserfahrung wird ebenfalls in Form von Unterrichtsnotizen bei den Kursunterlagen für die Auswertung verwahrt und im Kapitel 5.5 Beobachtungsergebnisse Kursverlauf ausgewertet.

## **5.5 Beobachtungsergebnisse Kursverlauf**

In diesem Kapitel wird kurz der Ablauf der einzelnen Kurse in verbaler Form zusammengefasst. Hierbei wird auf Besonderheiten in Bezug auf Anpassungen des Lehrstoffes oder gruppenspezifisches Verhalten Stellung bezogen. Die Darstellung dieser Ereignisse geschieht in einer Art Tagebuch, wie es auch die Trainer/innen für die entsprechenden Kurswochen geführt haben. Hier werden in kurzer Form die Be-

obachtungen und Ergebnisse des Unterrichts festgehalten, die in Kapitel 4.7 Lehreinheiten festgelegt wurden. Die Auswertung der Kursunterlagen, wie z.B. Tests, findet im Kapitel 5.7 Analyse Kursunterlagen und Tests statt und die Auswertung des Fragebogens im Kapitel 5.8 Auswertung des Fragebogen. Somit findet man in den folgenden Kapiteln eine Zusammenfassung der Beobachtungen und Mitschriften während des Unterrichts ergänzt mit einem Bezug auf die eigene Unterrichtserfahrung. Da der Vergleich der Unterrichtsstile nur in den Modulen 1 und 2 stattfindet, werden nur diese Wochen aktiv von Trainer/innen dokumentiert. Darauf folgende Module werden anhand der aktiven Beobachtung bewertet, da es in diesen Modulen zumeist keinen Test gibt und hierbei nicht zwischen einem prozeduralen und objektorientierten Lehransatz unterschieden wird. Ein Beispiel dieser Dokumentation kann man im Kapitel 5.5.1 Unterrichtsnotiz näher betrachten.

Wie man schon in Tabelle 5: Lehrplan IT-Informatiker/in erkennen kann, liegt der Schwerpunkt der Ausbildung in zwei Unterrichtsgegenständen, in der Datenorganisation (DO) und der Programmierung (PR).

Auf das Unterrichtsfach Datenorganisation wird im Verlauf der Arbeit nicht im Detail eingegangen, da sich dieses Unterrichtsfach fast ausschließlich mit relationalen Datenbankmanagementsystemen beschäftigt und sich frühestens im dritten Modul direkt mit dem Unterrichtsfach Programmieren überschneidet. Andere Unterrichtsfächer wie Analytik, die sich schon früher mit Programmieren überschneiden oder gegenseitig aufeinander aufbauen, werden nur am Rande erwähnt, sofern dies von Nöten ist.

Gerade in den folgenden Modulen, ändert sich das Unterrichtsfach Programmieren sehr stark in ein praxisorientiertes Unterrichtsfach: Diese Änderung sieht man ganz besonders in den Modulen 4 und 5 mit dem aktiven Kompetenztraining (AKT), im Modul 6, wo die Teilnehmer/innen ein Praktikum absolvieren müssen, und im Modul 8, beim Lehrabschlussprojekt.

### **5.5.1 Unterrichtsnotiz**

Folgend wird ein Beispiel aufgezeigt wie eine Unterrichtsnotiz auszusehen hat und was für Felder auszufüllen sind.

Klasse	Datum	Modul	Woche	Einheiten
<b>Anmerkungen</b>				
Anmerkungen können sich z.B. auf folgendes beziehen:				
<ul style="list-style-type: none"> <li>• Gruppendynamik/soziale Auffälligkeiten</li> <li>• Stoffänderungen/besprochene Stoffgebiete</li> <li>• Übungsangaben/Übungsdurchführungen</li> <li>• Änderungen bei Unterrichtsmethoden</li> <li>• Auffälligkeiten in Bezug auf die eigene Unterrichtserfahrungen</li> </ul>				

**Abbildung 5: Unterrichtsnotiz**

Für jeden Unterricht ist eine solche Unterrichtsnotiz auszufüllen. Die daraufhin anonymisierten Daten sind zusammengefasst worden und werden in den folgenden Kapiteln als Ergebnisse der Unterrichtsbeobachtung präsentiert.

### 5.5.2 Ergebnisse prozeduraler Ansatz – Klasse 1

Klassengröße: 7

Gruppendynamische Besonderheiten anhand der Unterrichtsbeobachtung: Die Klasse hat sich in zwei Gruppen geteilt, die sich nur sehr ungerne gegenseitig unterstützen und zusammenarbeiten. Außerdem hat ein Teilnehmer Englisch als Muttersprache, wodurch deutsche Fachbegriffe des Öfteren ein Problem sind und Prüfungsangaben auch auf Englisch verfasst wurden.

Zeitraster in Bezug auf Kapitel 4.7 Lehreinheiten:

Modul	Woche	Anmerkung
1	1	Keine besonderen Vorkommnisse. Alle Teilnehmer/innen finden sich in der IDE sehr schnell gut zurecht und sind schon sehr gespannt tiefer in die Programmierwelt einzudringen. Alle Teilnehmer/innen haben ausreichende Kenntnisse bezüglich des Dateisystems und finden sich in der Ordnerstruktur zurecht.
1	2	Wertdatentypen und String wurde unterrichtet. Variablen wurden zumeist verstanden. Die Partnerarbeit mit Zahlen und Datentyp Zuordnung ist gut angekommen. Befehle zum Einlesen von Text und Zahlen, sowie zur Ausgabe wurden behandelt. Typumwandlung wurde mittels der Klasse Convert, ohne auf diese näher einzugehen durchgeführt.
1	3	If-Verzweigung wurde besprochen. Mehrfachverzweigung mit Else-If wurde ebenfalls behandelt, verschachtelte Verzweigungen wurden ausgelassen. Erste Probleme mit der Klammern-Setzung bei ein paar Teilnehmer/innen. Die wichtigsten relationalen Operatoren <, >, <=, >=, ==, != wurden besprochen, sowie die logischen Operatoren && (und) und    (oder).
1	4	While-Schleife als kopfgesteuerte und fußgesteuerte Variante wurde unterrichtet und Unterschiede in Übungen festgehalten. Interaktionen mit Benutzereingaben wurden geübt. Am Ende der Woche

		wurden auch Kombinationen mit If-Verzweigungen durchgenommen, z.B. Erstellung eines einfachen Menüs welches solange sichtbar ist, bis man Ende eingibt und auf andere Eingaben entsprechend reagiert. Die Übungen haben gut funktioniert, doch Teilnehmer/innen bleiben des Öfteren dem Unterricht fern und lernen den versäumten Unterrichtsstoff nicht selbstständig nach.
1	5	For-Schleife behandelt. Einfache Zählschleifen in positive wie auch negative Richtung geübt. Die Operatoren ++, --, +=, -= und *= mittels Übungen erarbeitet. Im Rahmen der Blitzlichtmethode sind folgende Informationen gesammelt worden: <ul style="list-style-type: none"> <li>Die Teilnehmer/innen wünschen sich mehr Wiederholungen, weil unterschiedliche Teilnehmer/innen schon Fehlzeiten haben und deswegen Erklärungen versäumt haben.</li> </ul> Als Reaktion darauf hat der/die Trainer/in den Teilnehmer/innen versprochen den Stoff die kommende Woche nochmals von Anfang an zu wiederholen.
1	6	Wiederholung des Stoffs der Vorwochen mit Übungsbeispielen, die diesen Stoff kombinieren. Außerdem wurden bei den Datentypen auch Verweisdatentypen näher beschrieben und am Ende der Woche eine Einführung in Arrays gegeben. Es wird hauptsächlich fragend-entwickelnd unterrichtet und die Teilnehmer/innen bekommen Einzelarbeiten, wobei der/die Trainer/in alle Teilnehmer/innen gerade in Bezug auf den versäumten Unterrichtsstoff unterstützt. Ein/e Teilnehmer/in nützte diese Übungszeit nicht entsprechend und spielt mit dem Handy, woraufhin diese/r ein klärendes Gespräch mit dem/r Trainer/in hatte.
1	7	Eindimensionale Arrays werden besprochen. Erste kleine Algorithmen werden mit dem/r Trainer/in oder in Partnerarbeit am Papier entwickelt und selbstständig programmiertechnisch umgesetzt. Ein Beispiel hierfür wäre ein Array in umgekehrter Reihenfolge in ein anderes Array zu speichern.
1	8	Beenden der Übungen zu Array, sowie theoretischer Blick auf mehrdimensionale Arrays – diese werden aber nicht besprochen, wegen dem Zeitverlust durch die Wiederholung des vorhergehenden Stoffs. Einführung in die Klasse List<T>, mit Vor- und Nachteilen, sowie praktischen Beispielen dazu. Methoden verschieben sich in die folgende Woche. Die Fehlzeiten sind weiterhin im Vergleich zu anderen Klassen überdurchschnittlich und die Teilnehmer/innen unterstützen sich nicht den versäumten Stoff aufzuholen.
1	9	Theoretische Einführung in Methoden ohne Parameter anhand des Methoden-Stacks. Praktisches Beispiel zu Methoden mit einer kurzen Übung dazu. Weitere Erklärung zu Methoden mit Rückgabewert und dazugehörigen Übungen.
1	10	Methoden mit Übergabeparameter wurden erklärt, kommen aber nicht mehr zum Test. Die restliche Woche bis zum Test wurden Übungsbeispiele über den gesamten Unterrichtsstoff ausgeführt und besprochen. Im Rahmen des Feedbackgesprächs kommt ein ähnliches Ergebnis wie im Rahmen der Blitzlicht-Methode. Die Teilnehmer/innen

		fühlen sich aufgrund der erhöhten Fehlzeiten teils massiv überfordert. Die Tests sind bei diesen Teilnehmer/innen dementsprechend durchschnittlich bis schlecht. Die Trainer/innen haben sich untereinander beraten und Stoffgebiete für die Klasse gekürzt. Somit werden tiefergehende Details von weniger relevanten Themengebieten nicht mehr besprochen.
2	1	Wiederholung des Stoffs des ersten Moduls als Reaktion auf das Feedback, mit zusätzlichen Übungen in Bezug auf Methoden mit Übergabeparameter und Rückgabewert. Außerdem wird die lineare Suche als Übung aufgegeben.
2	2	Theoretisches Wissen zur binären Suche als Gruppenarbeit angeeignet. In Einzelarbeit wurde der Algorithmus implementiert. In Bezug darauf wurde die Rekursion von/m Trainer/in den Teilnehmer/innen näher gebracht. Die Teilnehmer/innen hatten bei der Gruppenarbeit eine schlechte Arbeitsaufteilung, doch durch die Intervention und Kontrolle des/r Trainer/in hat die Gruppenarbeit noch produktive Ergebnisse abgeliefert und die Gruppe hat trotz anfänglicher Schwierigkeiten positiv zusammengearbeitet.
2	3	Strukturen als komplexe Datentypen werden erklärt. Diese abstrakt, objektorientierte Denkweise bereitet ca. der Hälfte der Klasse Probleme, obwohl die Beispiele sehr einfach gewählt wurden. Die Fehlzeiten der Teilnehmer/innen sind wieder gestiegen, es sind ca. 50% pro Einheit anwesend, wobei sich die Teilnehmer/innen sich in der Anwesenheit abwechseln.
2	4	Wegen der hohen Abwesenheit wurden die einfachen Strukturen der Vorwoche nochmals besprochen. Weitere Übungen, die gemeinsam mit dem/r Trainer/in besprochen wurden, wurden daraufhin in Einzelarbeiten ausprogrammiert.
2	5	Partnerarbeiten zu Strukturen wurden durchgeführt, wobei jeweils zwei Teilnehmer/innen ein Übungsbeispiel von der Planung bis zur Durchführung umsetzten. Bei ungeraden Teilnehmer/innenzahlen haben einerseits die Trainer/innen ausgeholfen, andererseits wurden für das nächste Beispiel neue Partner/innen gewählt. Der Kontrollaufwand der Trainer/innen war auch hierbei wieder ein höherer, da nicht alle Teilnehmer/innen selbstständig mitarbeiten wollten. In den letzten Einheiten der Woche wurden die Unterschiede zu Klassen theoretisch erklärt. Die Blitzlicht-Methode hat die Unzufriedenheit der Klasse mit den häufig fehlenden Teilnehmer/innen noch einmal herauskristallisiert. Andererseits finden die Teilnehmer/innen die häufige Wiederholung von Unterrichtsthemen gut und wollen dieses ruhige Arbeitstempo beibehalten.
2	6	Der Aufbau von Klassen mit Feldern und Methoden wurde vermittelt. Konstruktoren wurden nur grob theoretisch erklärt. Zum Ende der Woche hin wurden überladene Methoden vorgestellt und in Übungen ausprogrammiert. Jede Übung zu Klassen und Strukturen hatte eine Planungsphase, wo auch der Aufbau der Struktur und/oder Klasse entstand. Danach folgte immer auch die Anwendungsphase, wo diese Struktur und/oder Klasse verwendet wurde.
2	7	In Partnerarbeit wurden Klassen geplant und danach haben die Trainer/innen auf die Problematik von öffentlich sichtbaren Feldern verwiesen. Daraufhin wurden Eigenschaften von Klassen erklärt

		<p>und gemeinsam implementiert. Weitere Beispiele bezüglich Klassen mit Felder, Eigenschaften, Konstruktoren und Methoden wurden in dieser Woche durchgeführt.</p> <p>Durch die verlängerte Zeit, die für die Planung von Strukturen verwendet wurde, empfand die Klasse die Planung von Klassen eher einfach – der Unterschied zu Strukturen ist der Klasse aber nicht komplett zu vermitteln, da die Unterschiede zwischen Wert- und Verweistypen anscheinend noch nicht zu 100% verstanden wurden.</p>
2	8	Die Vererbung von Klassen wurde zur Verdeutlichung der Unterschiede zu Strukturen näher gebracht. Einfache Vererbungsbeispiele wurden als Einzelarbeit geübt.
2	9	Da Gruppenarbeiten bis jetzt nur unter Mehraufwand möglich waren, werden wieder Einzelarbeiten verstärkt eingesetzt und Vererbung anhand von Polymorphie geübt. Außerdem wurden nochmals die Sichtbarkeiten von Membern einer Klasse an einem praktischen Beispiel besprochen, da diese nicht verstanden wurden.
2	10	Designpattern, abstrakte Klassen und Schnittstellen wurden für die Klasse komplett gestrichen. In der letzten Modulwoche wurde zum ersten Mal ein komplexeres Beispiel gemeinsam erarbeitet und dabei auch der Stoff der beiden Module des Kurses wiederholt. Außerdem hat der Abschlusstest des Moduls stattgefunden. Die Ergebnisse waren anhand der Fehlzeiten wenig überraschend, wobei die anwesenden Personen auch nicht überdurchschnittlich gut aufgefallen sind. Im Allgemeinen hat die Klasse ein eher nüchternes Feedback abgegeben, wobei die Teilnehmer/innen sich mehr untereinander kritisiert haben, als die Trainer/innen und den Unterrichtsaufbau sowie die Methodik an sich.

### 5.5.3 Ergebnisse prozeduraler Ansatz – Klasse 2

Klassengröße: 9

Gruppendynamische Besonderheiten anhand der Unterrichtsbeobachtung: Die Klasse arbeitet gut miteinander, doch muss dies von den Trainer/innen zumeist koordiniert werden. Einzelne Teilnehmer/innen haben Probleme mit der deutschen und englischen Sprache.

Zeitraster in Bezug auf Kapitel 4.7 Lehreinheiten:

Modul	Woche	Anmerkung
1	1	Kennenlernen der IDE erfolgte äußerst positiv, es gibt vereinzelt sehr motivierte Teilnehmer/innen, die auch in der Freizeit Unterrichtsstoff durchnehmen. Das Quiz bezüglich Fragen der IDE kam bei diesen Teilnehmer/innen ausgesprochen gut an. Andererseits gibt es nun schon eine/n Teilnehmer/in, welche/r nur einmalig anwesend war und ein/e Teilnehmer/in, die/der sich mit der deutschen Sprache sehr schwer tut.
1	2	Die/Der fehlende Teilnehmer/in ist wieder zum Unterricht erschie-

		nen und hat Erfahrung in der Programmierung bewiesen, indem er/sie sich schon jetzt sehr gut mit Datentypen auskennt und in der Partnerarbeit unterstützend und zuvorkommend ist. Ein/e sehr motivierte/r Teilnehmer/in arbeitet sehr brav mit, kann sich aber anscheinend Informationen nicht sehr lange merken. Diese/r Teilnehmer/in ist aber sehr motiviert und bemüht ein/e gute/r Programmierer/in zu werden. Zuerst wurden in dieser Woche Datentypen, danach Variablen besprochen. Zuweisungen und einfache Ein-, sowie Ausgaben wurden von den Teilnehmer/innen sehr gut aufgenommen.
1	3	Die If-Verzweigung wurde besprochen und die Übungsbeispiele mit Interaktion mittels Konsole wurden positiv aufgenommen. Die Typumwandlungen mittels der Klasse Convert wurden von dem/r Teilnehmer/in mit den Sprachschwierigkeiten nur mühsam erledigt, außerdem dürfte diese/r auch den Stoff der Vorwoche noch nicht verinnerlicht haben. Andere Teilnehmer/innen haben Probleme mit den Mehrfachverzweigungen mittels Else-If. Diese Verzweigung muss nächste Woche wiederholt werden.
1	4	Wiederholung von Else-If war ernüchternd, aber nach mehreren Einzelbeispielen und der Unterstützung der Trainer/innen, sowie der besseren Teilnehmer/innen dürfte dieses Themengebiet abgeschlossen sein. Die Einführung zur While-Schleife ist sehr gut angekommen, da manche Teilnehmer/innen schon komplexere Programme mittels Userinteraktion machen wollten. Erste Strukturen in der Klasse zeichnen sich heraus, sodass ca. die Hälfte der Klasse gut dem Unterricht folgen kann und eigenes Interesse entwickelt hat. Die restliche Klasse hat entweder Sprachprobleme, ist desinteressiert oder arbeitet lieber für sich – es gibt zwei sehr introvertierte Teilnehmer/innen, die auch nicht gerne mit anderen Teilnehmer/innen zusammenarbeiten.
1	5	Die Vertiefung der While-Schleife funktionierte sehr gut – auch bei den eher schwächeren Teilnehmer/innen. Die größten Probleme bereiteten noch die logischen Operatoren, deswegen wurden diese nochmals außerhalb von Analytik genauer besprochen und mittels Wahrheitstabelle erklärt. Hierbei lag der Fokus auf den Operatoren <i>und</i> , <i>oder</i> und <i>nicht</i> . Die Zählschleife wurde auch besprochen und die Übungsbeispiele sehr gut abgearbeitet, die meisten Syntaxfehler sind wie in fast jeder Klasse die Klammern oder Rechtschreibfehler, die besonders die Teilnehmer/innen mit Sprachproblemen betreffen. Die mittels Blitzlicht festgestellte Stimmung in der Klasse ist durchwegs positiv. Die fleißigen Teilnehmer/innen wünschen sich von den schwächeren Teilnehmer/innen, dass diese ein bisschen schneller mitarbeiten, aber im Allgemeinen waren alle Teilnehmer/innen zufrieden. Da die introvertierten Teilnehmer/innen sich aber so gut wie nie gemeldet hatten, wurden diese auch nach dem Unterricht kurz nach ihrer Meinung gefragt, doch auch hier gaben sie kein konstruktives Feedback, außer dass eh „alles OK“ wäre.
1	6	Die Einführung in Arrays und Listen wurde ganz gut aufgenommen. Die Unterschiede, sowie Vor- und Nachteile sind den Teilnehmer/innen zumeist klar. Die praktische Umsetzung von Arrays



		bereitet sehr vielen Teilnehmer/innen aber Probleme. Gerade Schlampigkeitsfehler in Bezug auf die Klammernsetzung und die Syntax beim Erstellen von Arrays.
1	7	Arrays wurden in Übungen, besonders in Partnerarbeiten, nochmals besprochen. Auf mehrdimensionale Arrays wurde verzichtet, da hierbei die Klammernsetzung und Syntax noch komplizierter ist. Die dynamische Liste bereitet den Teilnehmer/innen hingegen weniger Probleme und die verlorene Zeit von den Arrays wurde bisher gut aufgeholt.
1	8	Die Übungen zu den Listen kamen bei den Teilnehmer/innen viel besser an und wurden sichtbar verstanden. Einfache Methoden wurden erläutert und geübt. Die theoretische Einführung zu Rückgabewert und Übergabeparameter wurde auch durchgeführt.
1	9	Die Theorie zu Methoden wurde wiederholt und mehrere Einzelübungen durchgeführt. Die Bedeutung unterschiedlicher Methodensignaturen wurde aber noch nicht erkannt und die Teilnehmer/innen hatten große Probleme mit Methoden mit mehreren Übergabeparametern.
1	10	Weitere Beispiele zu Methoden wurden geübt, sowie die Bedeutung von Wert- und Verweistypen nochmals verinnerlicht. Im Fortschritt mit dem Lehrstoff ist die Klasse im Plan, nur vertiefende Themen wurden auf Kosten von einfachen Übungen ausgelassen. Diese Übungen sind aber sehr gut angekommen und die meisten Teilnehmer/innen empfinden Programmieren als ein sehr interessantes und lehrreiches Unterrichtsfach, fühlen sich aber auch sehr gefordert. Der Test zeigt gut, dass die Teilnehmer/innen ein gutes logisches Verständnis für grundsätzliche Programmabläufe entwickelt haben, aber durch die vielen Syntaxfehler, die aus ungenauem Arbeiten entstehen, ist die Klasse langsam in der Durchführung und verbringt viel Zeit mit der Fehlersuche. Nicht jede/r Teilnehmer/in hat den Test in der vorgegebenen Zeit absolviert.
2	1	Die lineare Suche als Gruppenarbeit und Wiederholung des Unterrichtsstoffs des ersten Moduls ist sehr gut angekommen. Die Teilnehmer/innen freuten sich über den Synergieeffekt aus Programmieren und Analytik. Die Teilnehmer/innen haben auch mittels Gruppenarbeit die binäre Suche prozedural implementieren können.
2	2	Wegen des schnellen Fortschritts wurde mit der Klasse die binäre Suche auch rekursiv implementiert und die Rekursion an sich besprochen. Man hat im Unterricht gut gesehen, dass das Prinzip der Rekursion für die meisten Teilnehmer/innen noch zu kompliziert ist. Am Ende der Woche wurden Strukturen besprochen und die Teilnehmer/innen haben in Partnerarbeit ihre erste Struktur selbst entwickelt.
2	3	Motiviert von den äußerst positiven letzten Wochen hat die Klasse fast selbstständig in Partnerarbeit die meisten Strukturbeispiele gemeistert. Die Trainer/innen haben sich hierbei auf die schwächeren Teilnehmer/innen konzentrieren können. Ohne die Motivation der besseren Teilnehmer/innen wäre eine derartige Unterstützung der schwächeren Teilnehmer/innen aber nicht möglich gewesen. Es haben sich verschiedene Gruppen innerhalb der Klasse gebil-

		det. Die besseren Teilnehmer/innen arbeiten miteinander und die schlechteren Teilnehmer/innen unterstützen sich nicht sehr stark untereinander, sondern warten passiv auf die Hilfe der Trainer/innen.
2	4	Der Versuch mittels Partnerarbeit schlechtere Teilnehmer/innen mit besseren Teilnehmer/innen zusammenarbeiten zu lassen hat nicht besonders gut funktioniert. Zumeist haben die Teilnehmer/innen trotzdem alleine gearbeitet und nur die Ergebnisse verglichen, sofern die Trainer/innen die Teilnehmer/innen nicht zur Kommunikation aufgefordert haben. Die Einführung zu Klassen wurde ganz gut aufgenommen, es wurde aber nur die Theorie besprochen und ein einfaches Übungsbeispiel gemeinsam erarbeitet.
2	5	Der tiefere Einblick in die Klassen hat ganz gut funktioniert. Ohne die häufige Unterstützung von Codesnippets der IDE hätten die Syntaxfehler aber überhandgenommen. Zuerst wurden Felder, und danach die dazugehörigen Eigenschaften, besprochen. Außerdem wurde ein kleiner Exkurs in das Exception-Handling unternommen, da die Klasse zeitlich gut im Plan liegt. Anhand der Blitzlicht-Methode wurde festgestellt, dass die Klasse ganz zufrieden mit den Übungen ist, aber diese noch immer Probleme mit Syntaxregeln hat.
2	6	Methoden und der Konstruktor, sowie der Lebenszyklus eines Objekts wurden besprochen. Gerade die schwächeren Teilnehmer/innen und der/die Teilnehmer/in der/die des Öfteren dem Unterricht fern bleibt, haben aber Probleme im Verständnis und der Unterscheidung von Klassen und Instanzen. Selbst die besseren Teilnehmer/innen verwechseln immer wieder die verschiedenen Begriffe und versuchen fehlerhafte Befehle umzusetzen.
2	7	Die Wiederholung der Syntax zu einfachen Klassen wurde wohlwollend aufgenommen. Vererbung von Klassen und ihren Members, sowie die dazugehörige Übung, ist von den Teilnehmer/innen mit Interesse abgearbeitet worden.
2	8	Die komplette Woche wurde Vererbung und Polymorphie unterrichtet, sowie in diversen Einzel- und Gruppenarbeiten untersucht. Hauptsächlich haben sich bei der Gruppenarbeit drei Gruppen herausgebildet, wobei nur eine dieser Gruppen wirklich als Gruppe zusammengearbeitet hat. Die beiden anderen Gruppen mussten sehr stark durch die Trainer/innen in der Kommunikation und Organisation geführt werden. Trotzdem hat jede Gruppe ihr Ziel erreicht, wobei es aber große qualitative Unterschiede gab. Eine Gruppe im speziellen hat auch viel Freizeit in diese Arbeit investiert.
2	9	Da der Fortschritt in der vorherigen Woche sehr gut war, wurden noch kurz abstrakte Klassen, anhand eines gemeinsam erarbeiteten Beispiels, angeschnitten. Die Implementation von Schnittstellen wurden auch in Einzelarbeiten geübt und in Kombination mit Polymorphie vorgeführt. Die Vorteile dieser Kombination von Programmier-techniken haben nur wenige Teilnehmer/innen der Klasse verstanden.
2	10	Klassen wurden wiederholt und Übungsbeispiele als Testvorberei-

		<p>tung durchgenommen. Die Teilnehmer/innen hatten sehr unterschiedlich abgeschlossen, von sehr guten, bis sehr schlechten Ergebnissen war alles vertreten. Trotz sehr viel Fleißaufgaben und motiviertem Einsatz im Unterricht, haben aber nicht alle fleißigen Teilnehmer/innen auch gute Noten erhalten, da diese unter Druck weiterhin die gleichen Syntaxfehler wie schon zu Beginn des Kurses gemacht haben. Trotzdem wurde der Kurs sehr positiv von den Teilnehmer/innen aufgenommen, selbst von denen, die sich nicht sehr aktiv daran beteiligt hatten. Die häufigste Kritik war, dass der Stoff zu schnell durchgenommen wurde und man nicht genug Zeit zum Üben hatte.</p>
--	--	--

### 5.5.4 Ergebnisse objektorientierter Ansatz – Klasse 1

Klassengröße: 8

Gruppendynamische Besonderheiten anhand der Unterrichtsbeobachtung: Gute Zusammenarbeit innerhalb der Gruppe.

Zeitraster in Bezug auf Kapitel 4.7 Lehreinheiten:

Modul	Woche	Anmerkung
1	1	Die Einführung in die IDE und das .NET Framework wurde sehr gut angenommen. Die Teilnehmer/innen haben bis auf zwei Teilnehmer/innen einen sehr soliden Wissensstand bezüglich Dateimanagement.
1	2	Die Übung zur Kategorisierung der Wertdatentypen wurde sehr gut angenommen. Die Erklärung der Verweisdatentypen anhand von DateTime und Random wurde sehr gut aufgenommen. Die erste Partnerarbeit hierzu hat sehr gut funktioniert. Zwei Teilnehmer/innen fallen positiv auf, indem diese andere Teilnehmer/innen tatkräftig unterstützen.
1	3	Die vertiefende Übung zu Wert- und Verweisdatentypen, wie auch Operatoren wurde sehr gut aufgenommen. Die Klasse scheut auch nicht davor zurück in Einzel- oder Partnerarbeit sehr selbstständig Recherche zu betreiben. Die Klassengemeinschaft scheint gut zu sein, was die weniger interessierten im Unterrichtsfach Programmieren wohl auch motiviert bei Partnerarbeiten aktiv mitzumachen. Obwohl in der Unit 3 im Kapitel 4.7.1.3 Aufteilung objektorientierte Gruppe nicht festgelegt, wurde zu den Strukturen auch eine Partnerarbeit vergeben, da diese Unterrichtsform in dieser Gruppe sehr gut funktioniert.
1	4	Der Fokus dieser Woche lag auf dem Unterschied zwischen Wert- und einfachen Verweistypen, die man selbst erstellt hat. Die dazugehörigen Übungen sind gut angekommen, da die Teilnehmer/innen selbst Datentypen planen durften und diese in kleinen Beispielen gleich einsetzen durften. Selbst die Teilnehmer/innen, die in den vorherigen Wochen Probleme mit dem Unterschied zwischen Wert- und Verweistyp hatten, dürften diesen diese Woche verstanden haben. Am Ende der Woche wurden noch Sichtbarkei-

		ten und die möglichen Problematiken dazu angesprochen.
1	5	Die Theorie und die Übungen zu Eigenschaften sind gut angekommen. Probleme herrschen zumeist noch bei manchen Teilnehmer/innen bei der Klammersetzung. Außerdem wurde diese Woche eine Einführung zu Methoden gezeigt, bei dieser hat man den get- und set-Teil von Eigenschaften nachprogrammiert. Teilnehmer/innen machen öfters den Fehler, dass in den Eigenschaften wieder auf die Eigenschaften verwiesen wird und dadurch ein Zyklus entsteht. Am Ende der Woche wurde deswegen das Codesnippet propfull vorgezeigt, was die Teilnehmer/innen bei der korrekten Syntax unterstützt. Im Unterricht ist jetzt vermehrt auf korrekte Syntax zu achten, da es nicht für alle Probleme Codesnippets gibt. Außerdem müssen die Unterschiede zwischen einem Feld und einer Eigenschaft nochmals genauer besprochen werden, damit die Teilnehmer/innen die Befehle aus dem Codesnippet auch verstehen.
1	6	Vertiefende Übung zu Eigenschaften mit selbst geplanten Klassen in einer Klassenbibliothek. Die Übungen zu den Datentypen wurden gut aufgenommen. Das Prinzip von Wert- und Verweistypen wurde in den vorherigen Wochen mit Hilfe der wiederholten Übungen verstanden. Am Ende der Woche wurde noch kurz gezeigt, wie man selbst Codesnippets erstellen kann. Hier fand zuvor eine Absprache mit den Webprogrammiertrainer/innen statt, ob XML schon besprochen wurde.
1	7	Die Erklärung und Implementierung von Verzweigungen in Kombination mit Eigenschaften und einfachen Methoden wurde gut angenommen, gerade die fragend-entwickelnde Methode bei den gesuchten Bedingungen wurde positiv aufgenommen. Die Teilnehmer/innen arbeiten sehr aktiv mit den Trainer/innen und untereinander zusammen. Relationale Operatoren mittels Wahrheitstabellen wurden durch extra Übungen erlernt. Die Klasse ist im Allgemeinen sehr schnell, gerade im Maschineschreiben, wodurch der Unterrichtsverlauf sehr zügig voranschreitet. Methoden sind deswegen schon zeitlich vorgezogen worden, obwohl diese erst in der Unit 6 zu tragen kommen würden.
1	8	Teilnehmer/innen zeigen erhöhtes Interesse und entwickeln eigene Vorschläge für Übungsbeispiele. Auch bei Einzelarbeiten stehen einzelne Teilnehmer/innen immer wieder den schwächeren Teilnehmer/innen zur Seite. Das Prinzip der Klassenbibliothek ist gut angenommen worden und wird sehr selbstständig von der Klasse wiederverwendet.
1	9	Komplexere Methoden mit Rückgabewert und Übergabeparametern wurden besprochen. Kleine Algorithmen in Bezug auf Verzweigungen wurden ebenfalls geübt. Am Ende der Woche hat sich gezeigt, dass dieser Umfang ein bisschen zu viel auf einmal für manche Teilnehmer/innen aus der Klasse war, und die Methoden wieder vereinfacht werden sollten.
1	10	Beispiele zu der Syntax von Methoden mit mehreren Parametern wurden durchgenommen. Einzelne Teilnehmer/innen hatten hierbei noch Probleme. Konstruktoren und Überladungen von Methoden wurden besprochen und geübt. Der Test ist im Allgemeinen

		ganz gut ausgefallen, nur zwei Teilnehmer/innen hatten gröbere Probleme, wobei eine/r davon starke Prüfungsangst hat. Die Teilnehmer/innen sind im Allgemeinen sehr angetan von der Unterrichtsform. Sie finden trotz den vielen Übungen alles ein bisschen theoretisch und können sich noch nicht wirklich vorstellen, wofür man Klassen wirklich benötigt. Hier wäre Nachholbedarf von praxisorientierteren Beispielen.
2	1	Wiederholung der Methoden ist gut bei den Teilnehmer/innen angekommen. Die Übung, wo anhand von Ablaufdiagrammen kleine Methoden implementiert wurden, ist besonders beliebt gewesen. Dies war auch gleichzeitig eine Übung für das Unterrichtsfach Analytik und somit wurde dieser Stoff fächerübergreifend wiederholt. Es wurden hauptsächlich Übungen zu der While-Schleife getätigt. Die For-Schleife wurde nur in einem kurzen Beispiel als Vergleich zu der While-Schleife vorgeführt.
2	2	Die For-Schleife wurde in Übungen behandelt, die Teilnehmer/innen haben hierbei auch die Operatoren +=, -= kennen gelernt. Die Operatoren ++ und -- haben die Teilnehmer selbstständig anhand des Programmierbuchs (Louis, Kansy, und Strasser, 2013) erarbeitet. Die ersten Schritte mit Arrays in Kombination mit der For-Schleife sind gut angekommen.
2	3	Die theoretische Gegenüberstellung der Arrays und Listen ist nicht so gut von statten gegangen, anscheinend wurde dieser Stoff in Analytik nicht entsprechend behandelt. Die praktischen Programmierübungen mit beiden Datenstrukturen sind hingegen ganz gut angekommen, wobei den Teilnehmer/innen die Syntax von Listen lieber war.
2	4	Die genauere Untersuchung der Member von Arrays und Listen ist gut bei den Teilnehmer/innen angekommen, wobei schon zuvor von manchen Teilnehmer/innen spezielle Methoden dieser Datenstrukturen, z.B. zum Sortieren, selbstständig erforscht wurden.
2	5	Die Theorie in Bezug auf die Vererbung war für ein paar Teilnehmer/innen verwirrend, hier wären mehr praktische Beispiele sinnvoller gewesen. Die Syntax an sich und einfache Übungen sind gut angekommen. In Partnerarbeit sind die Basisklassen entstanden, die wir in späteren Übungen durch Vererbung an spezielle Bedürfnisse angepasst haben. Wie fast immer in dieser Klasse sind gerade die Partnerübungen sehr gut abgelaufen. Die Gefahr ist, dass sich die schwächeren Teilnehmer/innen auf die stärkeren Teilnehmer/innen in diesen Übungen zu sehr verlassen. Im Allgemeinen kommt die Klasse aber überdurchschnittlich gut mit dem Lehrstoff voran.
2	6	Die Vererbung mit Verdecken und Überschreiben wurde behandelt. Die Teilnehmer/innen haben nun den Sinn von Vererbung in Hinsicht auf Arbeitserleichterung wahrgenommen. Außerdem wurde die Polymorphie in Bezug auf die ToString-Methode vorgestellt und von den Teilnehmer/innen positiv wahrgenommen.
2	7	Die Übung in Bezug auf Klassendiagramme ist so gut angekommen, dass mit der Klasse auch eine erweiterte Übung gemacht wurde. Die Klasse arbeitet sich sehr schnell in die Planung von Klassenhierarchien ein.

2	8	<p>Das abschließende Beispiel zu Klassen ist nur teilweise positiv angekommen, die meisten Teilnehmer/innen haben die Klassen aus einem Klassendiagramm generieren lassen und nur mehr die Methoden implementiert. Bei der Implementierung dieser Methoden wurden Schwächen in Bezug auf Schleifen und Verzweigungen aufgedeckt.</p> <p>Der Wechsel zu den Suchalgorithmen hat sich leider als nicht sehr flüssig dargestellt. Die lineare Suche, die eigentlich sehr leicht zu implementieren wäre, hat manchen Teilnehmer/innen Probleme bereitet. Hauptsächlich haben sich diese nicht mehr an die Syntax von Arrays erinnern können.</p>
2	9	<p>Die binäre Suche ist mit den Teilnehmer/innen gemeinsam erarbeitet worden und hierbei ist auch gleich der Stoff zu Arrays, Schleifen und Verzweigungen wiederholt worden. Der Bubblesort Algorithmus ist daraufhin in Gruppenarbeit zu erarbeiten gewesen und hierbei hat es deutlich weniger Probleme als noch die Woche zuvor gegeben. Es hat sich aber deutlich herauskristallisiert, welche Teilnehmer/innen sich mehr in der Freizeit mit Programmieren beschäftigen haben als andere. Diese konnten einfach davon profitieren, dass die Syntax der selteneren Befehle noch im Gedächtnis verblieben ist. Hier sollte man überlegen, ob man in den Übungsbeispielen vermehrt schon gehörtes Wissen überprüfen sollte.</p>
2	10	<p>Der Quicksort wurde gemeinsam besprochen und eine einfache Variante davon implementiert. Die restlichen Modulwochenstunden wurden zur Wiederholung des Gesamtstoffs und für den Test verwendet. Der Test hat im Allgemeinen genau die Schwächen gezeigt, die zu erwarten waren, dass die Planung von Klassen und objektorientiertes Verständnis ganz gut war, aber die Implementation von einfachen Schleifen mit Verzweigungen und Arrays oder Listen, doch immer noch manchen Teilnehmer/innen Probleme bereitet.</p>

### 5.5.5 Ergebnisse objektorientierter Ansatz – Klasse 2

Klassengröße: 13

Gruppendynamische Besonderheiten anhand der Unterrichtsbeobachtung: keine Besonderheiten trotz der größeren Gruppengröße.

Zeitraster in Bezug auf Kapitel 4.7 Lehreinheiten:

Modul	Woche	Anmerkung
1	1	<p>Die Einführungsstunden zu der Entwicklungsumgebung sind gut von statten gegangen. Die Teilnehmer/innen fanden sich in der IDE zurecht und konnten eigene Projektmappen mit Projekten erstellen. Die praktische Arbeit hierzu war trotz der Gruppengröße organisierbar. Der Wissenstand bezüglich Dateimanagement ist sehr gestreut, doch die besseren Teilnehmer/innen halfen schon jetzt den schwächeren Teilnehmern/innen aus, wodurch der/die Trainer/in entlastet wurde. Details zum .NET Framework wurden</p>

		nur kurz behandelt, da manche Teilnehmer/innen aufgrund von medikamentöser Behandlung eine kurze Aufmerksamkeitsspanne haben. Deswegen werden auch in Zukunft die theoretischen Themen, sofern möglich, in kurzen Blöcken abgehalten. Die genaueren Unterschiede werden im Laufe der folgenden Wochen besprochen und soweit wie möglich praktisch und sehr interaktiv mit den Teilnehmer/innen durchgenommen.
1	2	Die Partnerarbeiten und das fragend-entwickelnde Quiz bezüglich der Datentypen wurden sehr gut von den Teilnehmer/innen angenommen. Durch die Gruppengröße ist es leider gegeben, dass der/die Trainer/in nicht immer zu allen Teilnehmer/innen mit Problemen persönlich kommen kann. Die häufigsten Probleme oder Programmierfallen werden meist über den Beamer als Frontalunterricht geklärt, sowie es auch in dieser Modulwoche der Fall war. Nur in speziellen Fällen kommt der/die Trainer/in zu den Teilnehmer/innen, da dieses System vorerst als das praktikabelste erscheint. Die Teilnehmer/innen werden also schon von Anfang an darauf hingewiesen, sehr selbstständig Fehlermeldungen zu lesen und diese entsprechend zu interpretieren. Außerdem wird versucht die Partnerarbeit zu forcieren, da diese sehr gut angekommen ist und die Klassendynamik es zulässt. Dies hat hoffentlich auch den Effekt, dass vier Augen mehr sehen als zwei und somit die Fehlerquote ein bisschen nach unten sinkt.
1	3	In dieser Modulwoche hat sich gerade bei den Übungen zu den Verweistypen und Typumwandlungen die Partnerarbeit als Form des Unterrichts bewährt. Sie hält den Verwaltungsaufwand geringer und ist besser als die Übungen zur Einzelarbeit der Vorwoche angekommen. In gleicher Form wurde deswegen die Unit 3 dementsprechend abgeändert, dass die ersten Übungen in Partnerarbeit erarbeitet wurden und nur die letzten Beispiele in Einzelarbeit. Strukturen wurden als Datentypen behandelt.
1	4	Die Einführung zu Klassen wurde anhand der Vorwoche, die sich mit der Einführung von Strukturen beschäftigt hat, angepasst. Dieser Teil wurde auch gut von den Teilnehmer/innen angenommen. Die unterschiedlichen Einsatzmöglichkeiten von Klassen und Strukturen wurde von den Teilnehmer/innen noch nicht ganz verstanden und muss in der folgenden Woche nochmals forciert werden.
1	5	In Einzelarbeit wurde der Unterschied zwischen Strukturen und Klassen nochmals durchgenommen, wobei der Aufwand für den/die Trainer/in um einiges gestiegen ist. Jede Unklarheit der Teilnehmer/innen individuell zu lösen bedarf eines erhöhten Zeitaufwandes, welcher nur begrenzt zur Verfügung steht. Typische Fehler wie die Klammersetzung ließen den/die Trainer/in viel zwischen den Teilnehmer/innen hin und her wechseln. Hier hätte eine Installation eines Unterstützungstools für die Klammersetzung einiges an Aufwand reduziert. Der Mehraufwand an Übungen hat einigen Teilnehmer/innen gut geholfen, doch noch ist nicht der gesamten Klasse der Unterschied zwischen Strukturen und Klassen komplett klar. Diese Woche wurde außerdem die Klassenbibliothek besprochen, wobei sich hier wieder die teils guten Dateimanage-

		<p>ment-Kenntnisse der Klasse positiv bemerkbar gemacht haben. Außerdem hat den Teilnehmer/innen die Übung wieder Selbstvertrauen gebracht, welches nach den Verweistypen teilweise sichtbar geschwächt war. Das Feedback der Klasse war bis jetzt äußerst positiv, einzelne Teilnehmer/innen – gerade die erfahreneren Programmierer – wünschen sich mehr Beispiele für Einzelarbeiten. Die schwächeren Teilnehmer/innen würden sich ein langsames Tempo wünschen. Anhand der Beobachtung der Trainer/innen liegt es bei diesen Teilnehmer/innen nicht so sehr am Programmierverständnis. Es liegt eher daran, dass diese Teilnehmer/innen nicht den ganzen Unterricht mitbekommen, da diese meist mit dem Tippen auf der Tastatur zu langsam sind und deswegen nicht alles mitschreiben. Als Folge dessen werden die Trainer/innen das Tempo beim Tippen reduzieren und öfters Feedback, bezüglich des Tempos, einholen. Die Sichtbarkeiten der Klasse wurden noch nicht im Detail besprochen, dieses Themengebiet wurde auf nächste Woche verschoben.</p>
1	6	<p>In dieser Woche wurden die Übungen zu Eigenschaften, sowie die Sichtbarkeiten private, internal und public besprochen. Beide Themengebiete sind gut angekommen. Codesnippets zu wichtigen Befehlen wurden besprochen. Diese Codesnippets sollen die langsameren Teilnehmer/innen beim Maschineschreiben unterstützen.</p>
1	7	<p>Verzweigungen in Kombination mit Eigenschaften wurden besprochen. Die einfache Verzweigung ist sehr gut angekommen, die Mehrfachverzweigung war manchen Teilnehmer/innen zu schnell besprochen und gehört wiederholt. Die Operatoren für logische Zusammenhänge wurden ganz gut angenommen und durften schon in Analytik genauer besprochen worden sein. Bei Einzelarbeiten ist aufgefallen, dass manche Datentypen leider nicht mehr so gut im Gedächtnis sind und mit Datentypen von SQL verwechselt werden.</p>
1	8	<p>Es wurden Klassendiagramme aus Klassen und umgekehrt erstellt. Diese Übung ist bei den Teilnehmer/innen gut angekommen. Es ist nun damit zu rechnen, dass die langsameren Teilnehmer/innen mehr Klassen über das Klassendiagramm erstellen lassen. Die Wiederholung zu Feldern, Eigenschaften und Verzweigungen ist sehr gut angekommen und selbst die schwächeren Teilnehmer/innen können diese erstellen. In dieser Woche wurden hauptsächlich Einzelarbeiten getätigt, damit einzelne Teilnehmer/innen sich nicht von der Mitarbeit entziehen konnten und die Trainer/innen den individuellen Fortschritt besser einschätzen können. Im Allgemeinen bleibt aber die Partnerarbeit im Fokus, da diese sich bis jetzt doch ausgezeichnet hat.</p>
1	9	<p>Als Einführung in die Methoden wurde der Konstruktor verwendet. Diese Übung wurde nochmals verwendet um den Unterschied zwischen Wert- und Verweistypen deutlich zu machen. Hierbei wurden Konstruktoren ohne Parameter und mit Parameter als Frontalunterricht, Partnerarbeit und Einzelarbeit durchgenommen. Die zweite Hälfte der Woche hat sich in ähnlicher Form mit einfachen Methoden ohne Rückgabewert beschäftigt.</p>
1	10	<p>Methoden mit Rückgabewert wurden besprochen und geübt. Bei</p>



		mehreren Übergabeparametern und der Methodensignatur herrschen noch Unklarheiten. Methoden im Allgemeinen sind zum Test in sehr einfacher Form vorgekommen, hauptsächlich war der Schwerpunkt bei den Unterschieden von Datentypen und dem Erstellen, sowie Verwenden von Klassen. Der Test wurde im Allgemeinen positiv beendet, auch die Teilnehmer/innen haben den bisherigen Verlauf als positiv bewertet. Sie finden nur, dass viele Themen zu theoretisch und abstrakt sind und sie sich noch nicht wirklich etwas vorstellen können. Ein Wechsel von der Konsole in die grafische Programmierung wäre hier ein Wunsch der Teilnehmer/innen – dieser Wechsel ist aber für den Kurs zu diesem Zeitpunkt nicht geplant. Die grafische Benutzeroberfläche ist ein Themengebiet, das nach dem Kurs, im Modul 3 besprochen wird.
2	1	Die Nachbesprechung zum Test und die Wiederholung über den Stoff des Moduls 1 ist gut bei den Teilnehmer/innen angekommen. Im Rahmen der Methodenübung wurde die For-Schleife vorgestellt. Die weiteren Übungen in Einzelarbeit sind für den/die Trainer/in sehr anstrengend gewesen, da viele Teilnehmer/innen noch Probleme mit der Syntax von Befehlen haben. Sobald keine Codesnippets für Eigenschaften von Klassen verwendet werden, stimmt die Syntax nicht mehr. Die kopfgesteuerte While-Schleife wurde am Ende der Woche erklärt und Codesnippets sind für diesen Zeitraum vom Trainer verboten worden, damit die Teilnehmer/innen mehr auf die Syntax achten.
2	2	Ohne Verwendung der Codesnippets ist das Arbeitstempo der Klasse in dieser Woche erheblich gesunken. Es wurden nur Schleifen besprochen und entsprechende Übungen gemacht. Es war für die Teilnehmer/innen des Öfteren frustrierend und Kräfte raubend.
2	3	Durch das langsamere Vorankommen sind Arrays in dieser Woche nur kurz besprochen worden um die Grundlagen zu erläutern. Anschließend wurden Arrays mit Listen verglichen. Praktisch wurde nur mit Listen gearbeitet und die Klasse hat die dazugehörigen Übungen gut angenommen.
2	4	Erweiterte Übungen zu Listen wurden besprochen und zumeist in Einzelarbeit erledigt. Komplexere Thematiken wie Generika und verschachtelte Listen, sowie Arrays wurden nicht besprochen und müssen auf spätere Module mit mehr zeitlichen Puffern, wie z.B. die AKT-Module, verschoben werden.
2	5	Vererbung wurde erklärt und praktisch geübt. Teilnehmer/innen fühlten sich wieder sicherer und zeigten mehr Begeisterung als bei den Schleifen und Methoden. Man sieht jetzt schon, dass die Teilnehmer/innen mit den prozeduralen Themengebieten wie Schleifen und Verzweigungen mehr Probleme hatten als mit der Definition von Klassen. Wobei nach Hinterfragen des/der Trainer/in manche Teilnehmer/innen noch Probleme bei der theoretischen Klassendefinitionen haben. Im praktischen Einsatz haben aber nur die wenigsten Teilnehmer/innen Probleme mit Klassen.
2	6	Anhand der Aufzeichnungen der vorherigen Woche wurde in dieser Woche wieder vermehrt ein Fokus auf Methoden gelegt und diese gleich mit Vererbung im Sinne des Überladens und Verde-

		cken besprochen. Im Zuge der Vererbung wurden die theoretischen Grundlagen von Klassen noch einmal im Detail besprochen.
2	7	Das Themengebiet der Polymorphie mit Vererbung und Überschreiben von Methoden ist diese Woche ungewöhnlich gut von statten gegangen. So wie in den meisten Klassen, haben die Teilnehmer/innen noch nicht komplett den Sinn hinter Polymorphie verstanden, doch die Syntax und Voraussetzungen wurden von den Teilnehmer/innen gut aufgenommen.
2	8	Überschreiben von Methoden anhand von ToString wurde geübt, sowie ein kleines Projekt (Tierpension) in Partnerarbeit ausprogrammiert. Das Projekt hat mehr Zeit in Anspruch genommen als geplant, ist aber anhand der Reaktionen der Teilnehmer/innen ganz gut angekommen. Hierbei wurde fast der komplette bisherige Stoff in einem Projekt wiederholt und der/die Trainer/in hatte Zeit auf individuelle Fragen einzugehen.
2	9	In Gruppenarbeiten wurde die Arbeitsweise von der linearen Suche und der binären Suche aufgearbeitet und teilweise als Rollenspiele präsentiert. Die Algorithmen selbst sind in iterativer Einzelarbeit implementiert worden.
2	10	Sortieralgorithmen wurden vorgestellt und der Bubblesort auch von den Teilnehmer/innen in Einzelarbeit implementiert. Der Quicksort wurde nur im Frontalunterricht vorgeführt, wobei die Teilnehmer/innen hierbei interaktiv, fragend-entwickelnd, zu der Implementierung befragt wurden. Diese Interaktion ist gut bei den Teilnehmer/innen angekommen. Bei der Fragenstellung wurden besonders die schlechteren Teilnehmer/innen herangezogen. Die praktische Umsetzung des Quicksort wurde aus zeitlichen Gründen nicht mehr von den Teilnehmer/innen selbst implementiert. Der Abschlusstest ist äußerst positiv abgelaufen.

## 5.6 Beobachtung darauffolgender Module

Da sich der Kurs nur über die Module 1 und 2 erstreckt, sind die restlichen Module bis zur Lehrabschlussprüfung nicht im Detail beschrieben. Eine grobe Zusammenfassung über die Beobachtungen und den weiteren Verlauf der Gruppen findet man in dem Kapitel 5.9 Überprüfung der Hypothese und dem Kapitel 6 Schlussfolgerungen.

## 5.7 Analyse Kursunterlagen und Tests

In diesem Kapitel werden die Kursunterlagen von den Teilnehmer/innen der unterschiedlichen Unterrichtsstile gegenübergestellt. Diese Gegenüberstellung findet in Form eines Vergleichs der Mitarbeits- und Testnoten statt.

Die Mitarbeitsnote ergibt sich aus folgenden verschiedenen Faktoren:

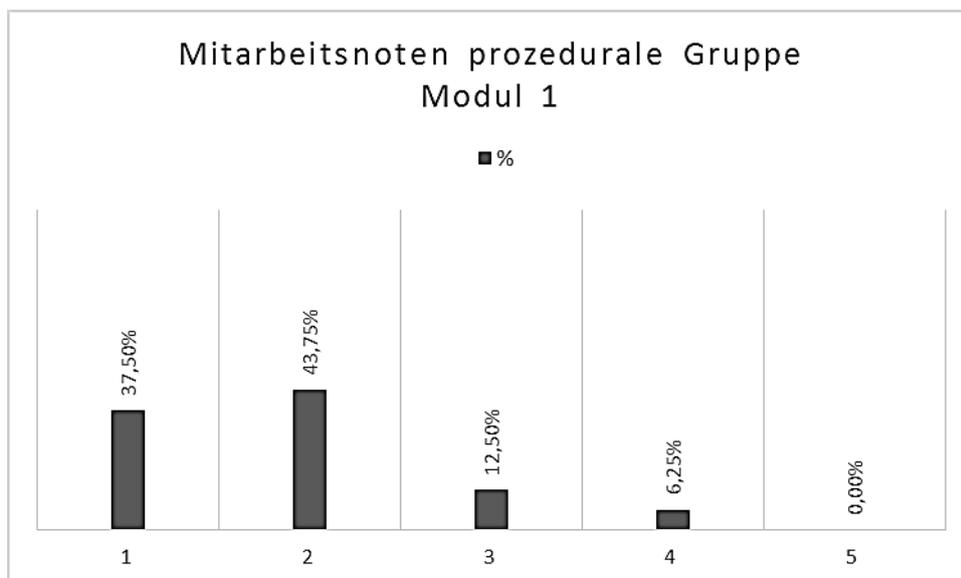
- Aktive Beteiligung am Unterricht
- Durchführung der aufgegebenen Übungen
- Aktive Beteiligung an Partner-, Gruppenarbeiten und Referaten

- Ergebnisse aus den Stundenwiederholungen und den Erhebungen des Wissensstandes aus der Modulwoche 5 des jeweiligen Moduls

Die Testnoten, die sich aus den praxisorientierten Tests am Ende jedes Moduls errechnen lassen, sind Fakten die gut miteinander vergleichbar sind. Am Ende des Kurses – d.h. am Ende des Moduls 2 – sollten beide Gruppen einen fast deckungsgleichen Wissensstand bezüglich prozeduraler und objektorientierter Programmierung haben. Ein Vergleich dieser Testergebnisse kann gerade hierbei sehr aussagekräftig sein, welche Lehrmethode die objektorientierte Programmierung erfolgreicher vermittelt hat.

### 5.7.1 Mitarbeitsnoten prozedurale Gruppen

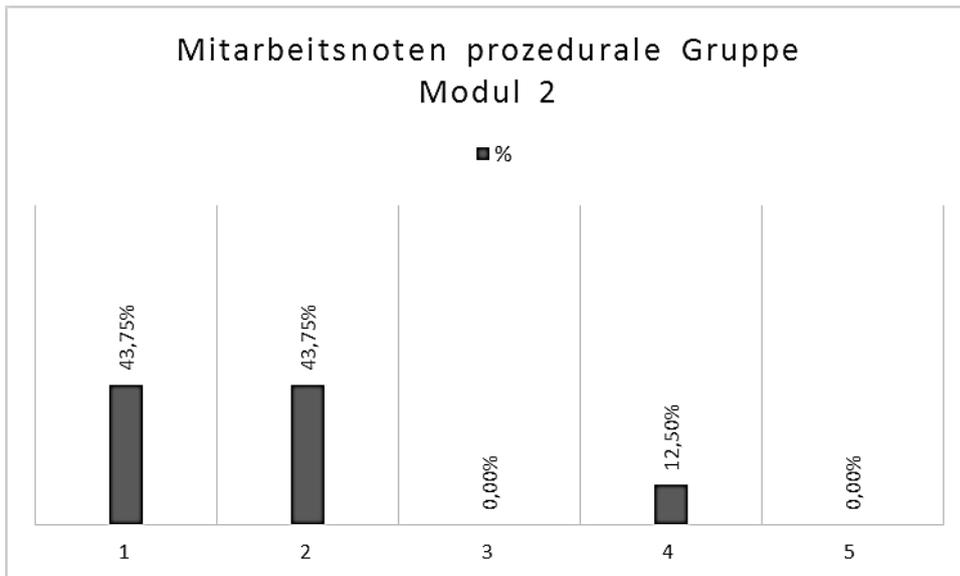
Die Mitarbeitsnoten der beiden prozeduralen Gruppen wurden zusammengerechnet und ergeben folgende Notenverteilung für das Modul 1:



**Abbildung 6: Verteilung Mitarbeitsnoten prozedurale Gruppe Modul 1**

Hieraus ergibt sich eine durchschnittliche Mitarbeitsnote von 1,875 für Teilnehmer/innen der prozeduralen Gruppe. Von den insgesamt 16 Teilnehmer/innen haben 13 Teilnehmer/innen entweder die Note 1 oder 2 als Mitarbeitsnote erhalten. Daraus ergibt sich, dass bis auf drei Teilnehmer/innen, alle anderen Teilnehmer/innen sehr aktiv am Unterricht beteiligt waren.

Im folgenden Modul hat sich nicht sehr viel an dieser aktiven Beteiligung geändert wie folgendes Diagramm aufzeigt:

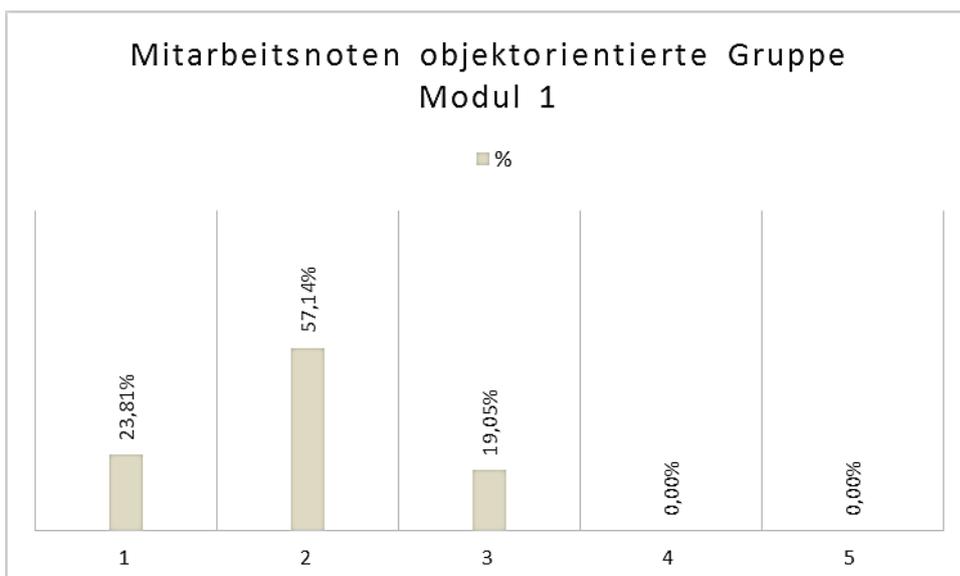


**Abbildung 7: Verteilung Mitarbeitnoten prozedurale Gruppe Modul 2**

Die durchschnittliche Mitarbeitnote ergibt daraus einen Wert von gerundeten 1,813 – was einer allgemeinen Verbesserung im Gegensatz zum Modul 1 entspricht. Nur zwei Teilnehmer/innen von den insgesamt 16 Teilnehmer/innen haben nur das nötigste für eine positive Mitarbeit getan.

### 5.7.2 Mitarbeitnoten objektorientierte Gruppen

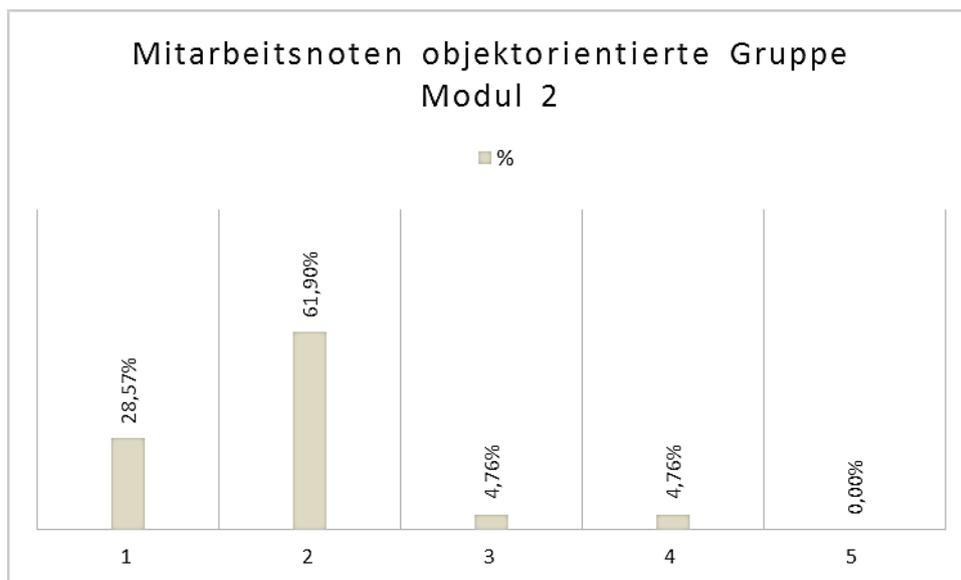
Die Mitarbeitnoten für die objektorientierte Gruppe wurden genauso wie die der prozeduralen Gruppe zusammengerechnet und ergeben für das Modul 1 folgende Verteilung:



**Abbildung 8: Verteilung Mitarbeitnoten objektorientierte Gruppe Modul 1**

Die hieraus berechnete durchschnittliche Mitarbeitsnote ergibt einen gerundeten Wert von 1,952. Dieser ist zwar nicht besser als der Wert aus der prozeduralen Gruppe aus Modul 1, doch gibt es in der objektorientierten Gruppe mit 21 Teilnehmer/innen keine/n einzige/n Teilnehmer/in der/die eine Note 4 oder 5 bekommen hat. 80,95% der Teilnehmer/innen des objektorientierten Lehransatzes haben eine sehr gute oder gute Mitarbeit und sind somit sehr aktiv am Unterrichtsgeschehen beteiligt.

Für das Modul 2 hat sich der Durchschnittswert der Mitarbeitsnote der objektorientierten Gruppe sogar noch ein bisschen auf gerundete 1,857 verbessert und die Notenverteilung folgend verschoben:

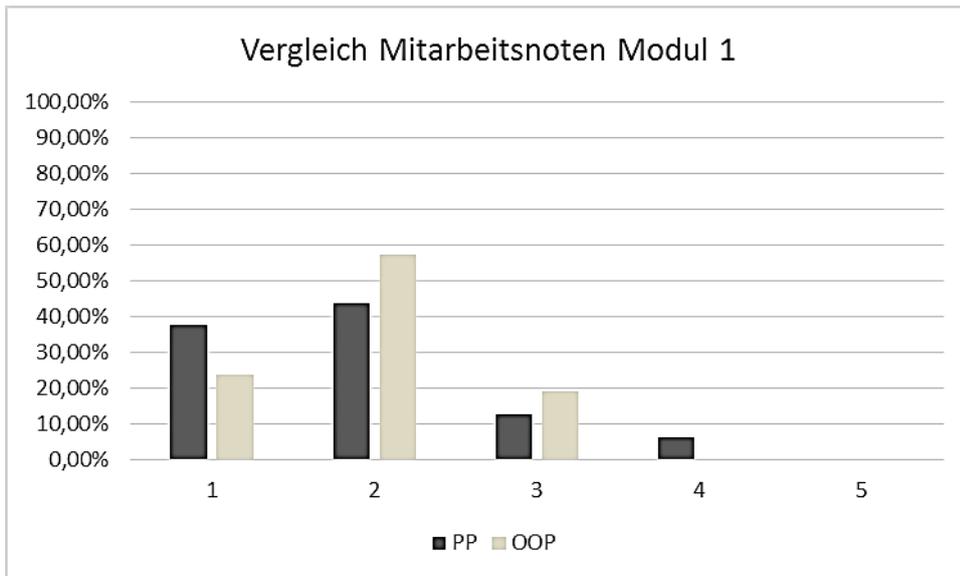


**Abbildung 9: Verteilung Mitarbeitsnoten objektorientierte Gruppe Modul 2**

Hierbei kann man erkennen, dass die sehr gute und gute Mitarbeit gemeinsam auf hervorragende 90,48% aller Teilnehmer/innen gestiegen ist. Während es im Modul 1 noch vier Teilnehmer/innen mit einer befriedigenden Mitarbeit gegeben hat, so gibt es im Modul 2 nur mehr eine/n Teilnehmer/in mit befriedigender Mitarbeit. Zwei Teilnehmer/innen aus dieser Gruppe haben sich hierbei verbessern können. Ein/e Teilnehmer/in konnte leider nur mehr eine genügende Mitarbeit erreichen.

### 5.7.3 Gegenüberstellung der Mitarbeitsnoten

Im direkten Vergleich der Mitarbeitsnoten des Moduls 1 zwischen dem prozeduralem und objektorientiertem Lehransatz kann man folgende Verteilung erkennen.



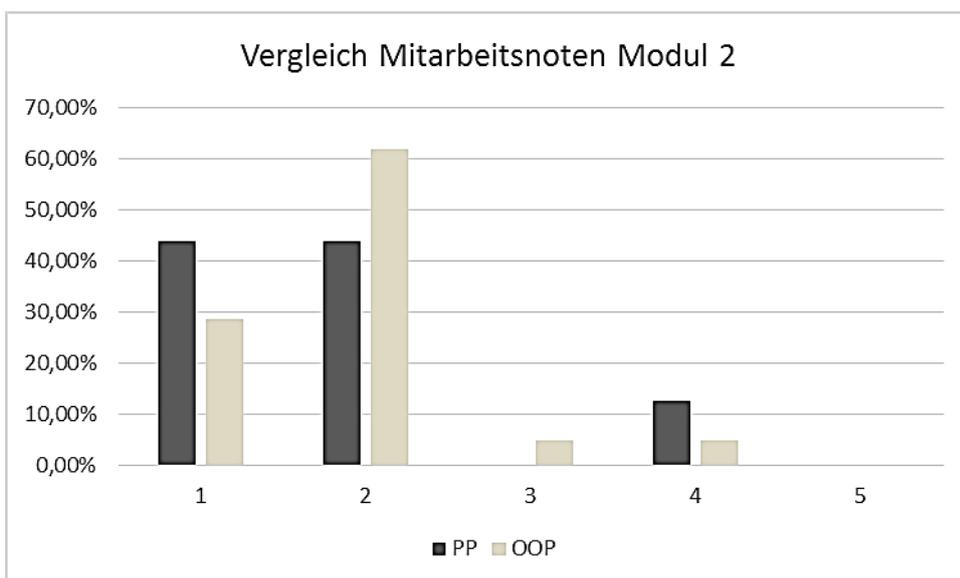
**Abbildung 10: Vergleich Mitarbeit Modul 1 PP vs. OOP**

Mehr Prozent der Teilnehmer/innen aus der prozeduralen Gruppe erlangen bezüglich der Mitarbeit eine sehr gute Benotung. Außerdem sind weniger Prozent der Teilnehmer/innen der prozeduralen Gruppe mit einem Befriedigend benotet worden.

Die objektorientierte Gruppe hat dafür im Modul 1 eine bessere prozentuelle Verteilung bei der guten Mitarbeit und hat keine/n Teilnehmer/in der/die nur ein Genügend in der Mitarbeit erreicht hat.

Die prozentuellen Unterschiede der beiden Gruppen sind so marginal, dass diese nicht besonders zu bewerten sind. Somit kann man bei beiden Gruppen von einer ungefähr gleich guten Mitarbeit sprechen.

Die Auswertung der prozentuellen Verteilung für das Modul 2 sieht folgend aus:



**Abbildung 11: Vergleich Mitarbeit Modul 2 PP vs. OOP**

Auch hierbei ist die prozedurale Gruppe besser mit sehr guten Mitarbeitsnoten vertreten. Genauer gesagt ist hierbei die Verteilung der sehr guten und guten Mitarbeit gleich mit jeweils 43,75% vertreten. Negativ wird diese Verteilung nur von den zwei Teilnehmer/innen beeinflusst die nur eine genügende Mitarbeit besitzen.

Die objektorientierte Gruppe hat wie im Modul 1 schon einen verstärkten Fokus in der guten Mitarbeit. Im Vergleich zu Modul 1 haben aber die beiden besten Mitarbeitsnoten – sehr gut und gut – nochmals jeweils marginal um ca. 5% zulegen können.

Beim Vergleich der beiden Lehransätze fällt gerade im Modul 2 auf, dass die prozedurale Gruppe eine ziemlich gleiche Verteilung zwischen sehr guter und guter Mitarbeit hat und die objektorientierte Gruppe, doch sehr stark in Richtung guter Mitarbeit tendiert. Nur bei den schwächeren Teilnehmer/innen, in Hinsicht der Mitarbeit, gewinnt die objektorientierte Gruppe den Vergleich minimal für sich.

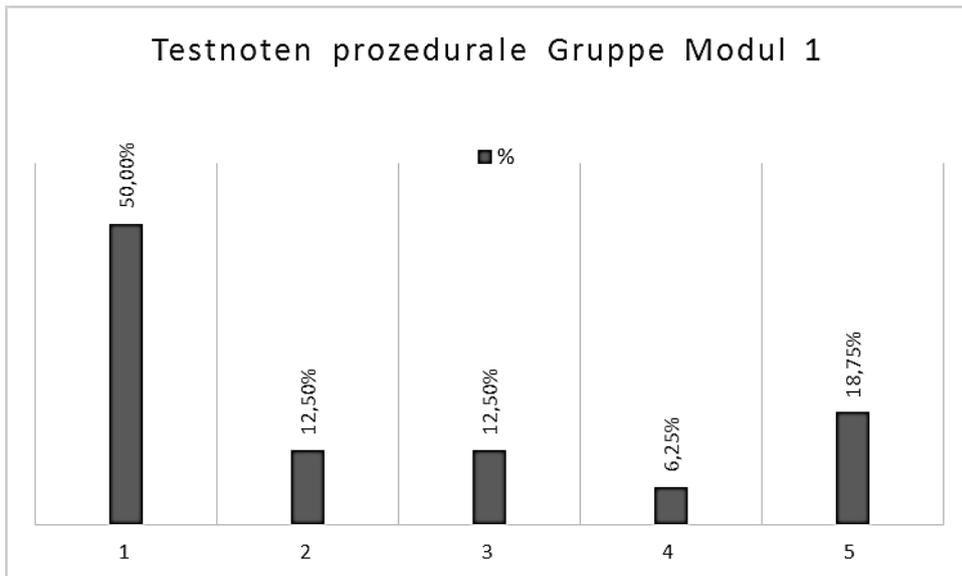
Im Allgemeinen sieht man aber sehr gut, dass sich in den einzelnen Lehrstilen vom Modul 1 hin zum Modul 2 nur sehr wenig bei der Verteilung der Mitarbeit getan hat, d.h. die Mitarbeit der Teilnehmer/innen ist modulübergreifend konstant geblieben.

Unabhängig von der Lehrmethode sind die Unterschiede in der Mitarbeit beider Gruppen so gering, dass es anhand des Lehrstils keinen gewichtigen Unterschied in dieser gibt.

#### **5.7.4 Testergebnisse prozedurale Gruppen**

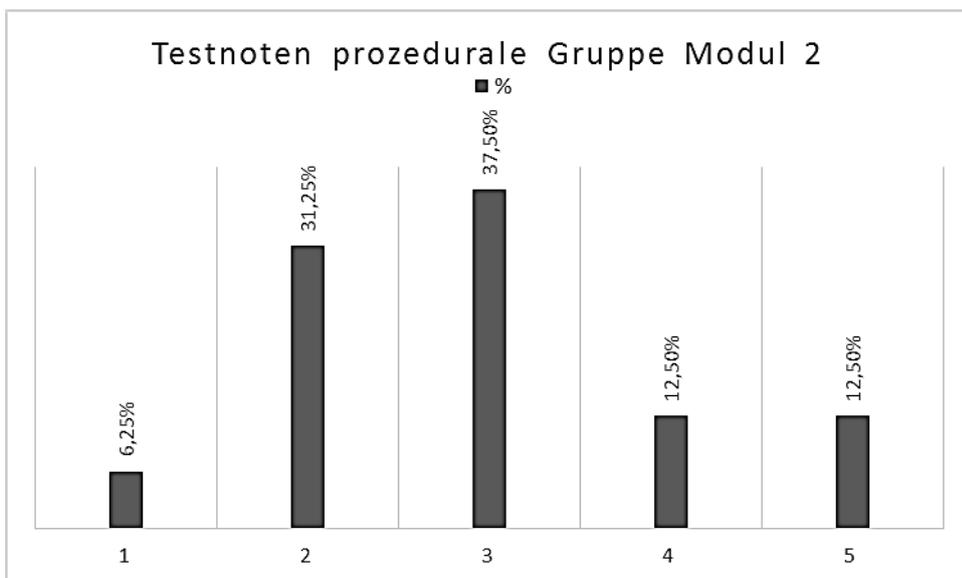
Die Testergebnisse wurden wie die Mitarbeitsnoten von beiden prozeduralen Gruppen zusammengerechnet und daraus eine prozentuelle Darstellung erstellt.

Im Modul 1 kann man bei der prozeduralen Gruppe gut erkennen, dass die Grundlagen der prozeduralen Programmierung für viele Teilnehmer/innen leicht umzusetzen waren. Die Note 1 ist mit 50% sehr dominant gegenüber den anderen Noten vertreten. Negativ fällt leider auf, dass die nächst dominante Note das nicht genügend ist, d.h. es hat sich eine leichte Schere zwischen den sehr guten und negativen Tests, wie in folgender Grafik ersichtlich, entwickelt.



**Abbildung 12: Verteilung Testnoten prozedurale Gruppe Modul 1**

Im Modul 2, wo die prozedurale Gruppe einen Fokus auf die objektorientierten Themengebiete der Programmierung nimmt, ändert sich die Notenverteilung sehr drastisch:



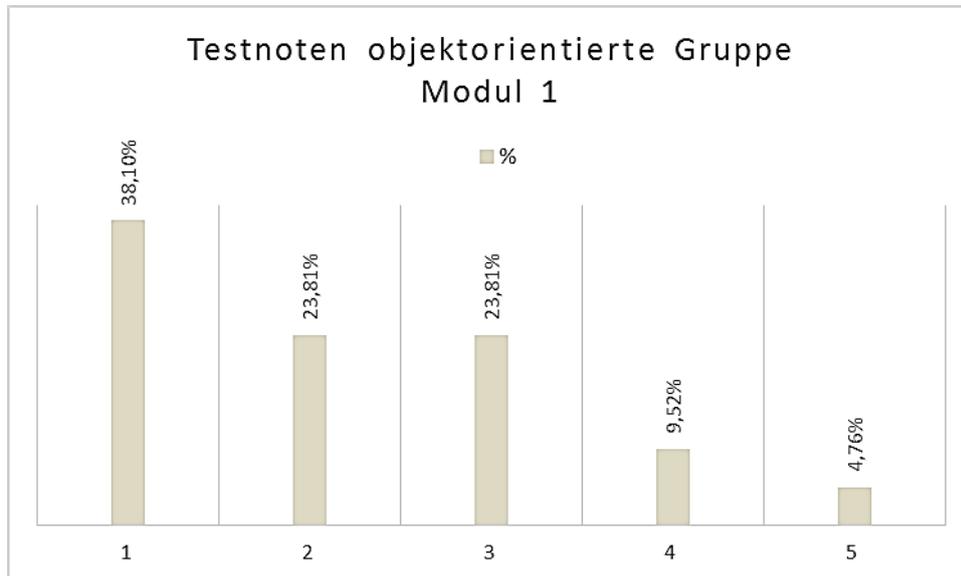
**Abbildung 13: Verteilung Testnoten prozedurale Gruppe Modul 2**

Hierbei ist sehr gut ersichtlich, dass die 50% der sehr guten Ergebnisse nun fast vollständig auf die Noten 2 und 3 aufgeteilt wurden. Somit macht nun diese Kategorie von gut bis befriedigend einen Anteil von 68,75% der Gesamtnoten aus. Der Gesamtanteil von 25% für genügend und nicht genügend ist gleich geblieben, wobei hier eine kleine positive Verschiebung stattgefunden hat.



### 5.7.5 Testergebnisse objektorientierte Gruppen

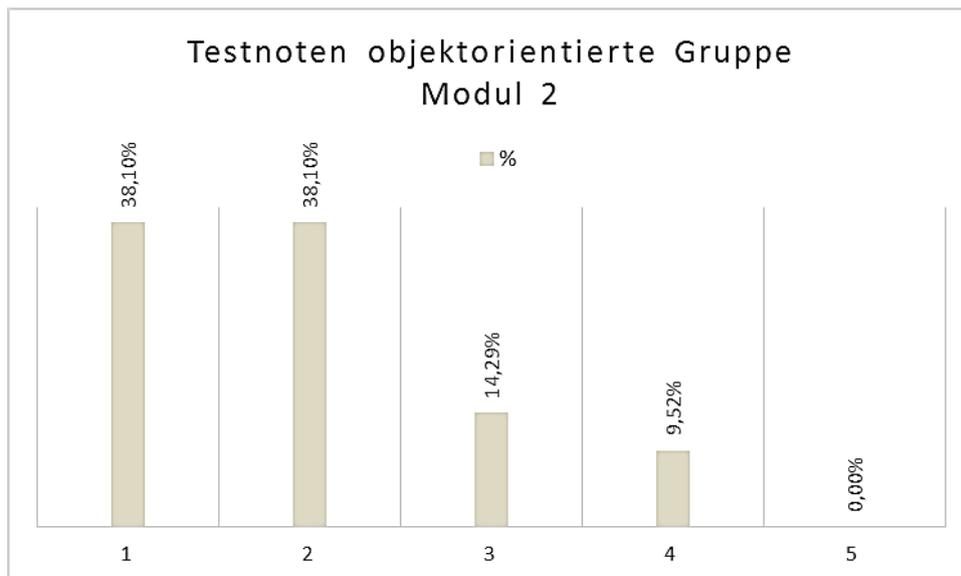
Die Testnoten der objektorientierten Gruppe verteilen sich sehr positiv von vielen sehr guten Noten bis zu wenigen nicht genügend wie folgend ersichtlich:



**Abbildung 14: Verteilung Testnoten objektorientierte Gruppe Modul 1**

Nur drei Teilnehmer/innen machen die 14,29% aller Genügend und nicht Genügend aus. Dies ist ein ganz guter Wert, wenn man diesen gegenüber den restlichen 18 Teilnehmer/innen stellt die besser abgeschnitten haben. Auch in der objektorientierten Gruppe dominiert die Note 1 ein bisschen gegenüber den anderen Noten. Im Vergleich zu der prozeduralen Gruppe gibt es hier aber eine schönere Verteilung der Noten und keine Schere zwischen sehr gut und nicht genügend.

Die nachfolgende Grafik verdeutlicht, dass die objektorientierte Gruppe im Modul 2 weniger Probleme mit den objektorientierten Vertiefungen hat und sich der Notendurchschnitt sogar noch anheben lies.



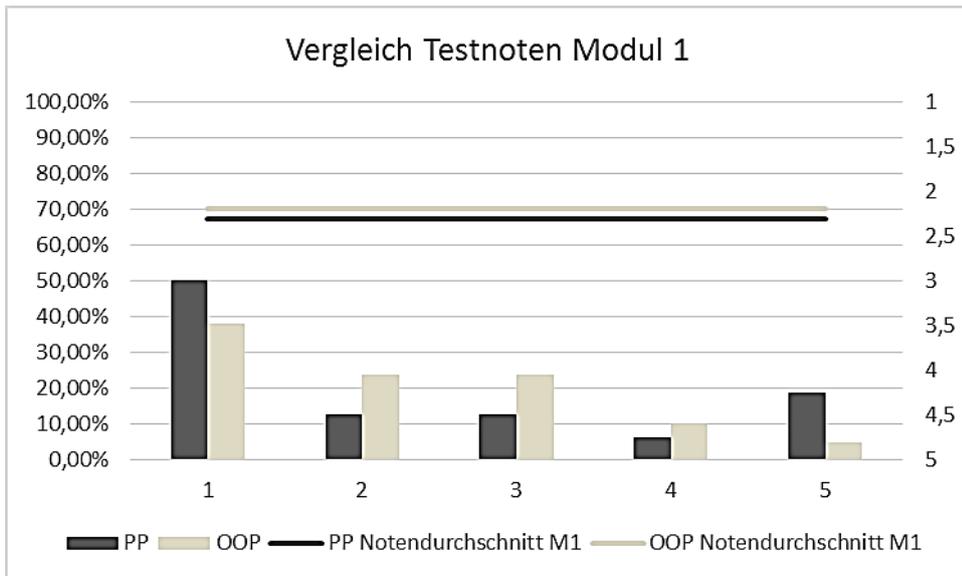
**Abbildung 15: Verteilung Testnoten objektorientierte Gruppe Modul 2**

Anhand der Abbildung kann man gut erkennen, dass die sehr guten Teilnehmer/innen konstant geblieben sind, sich aber die restlichen Noten in positiver Hinsicht verschoben haben. Kein/e Teilnehmer/in hat am Ende des zweiten Moduls einen negativen Test geschrieben und nur zwei Teilnehmer/innen haben nur ein Genügend erreicht.

### 5.7.6 Gegenüberstellung der Testergebnisse

Die Gegenüberstellung der Testnoten soll zeigen, in welchen Modulen sich welcher Lehransatz besser durchsetzen kann. Hierbei ist noch einmal zu erwähnen, dass sich die prozedurale Gruppe im Modul 1 nicht mit objektorientierten Schwerpunkten beschäftigt, sondern all diese Themen erst im Modul 2 zu hören bekommt. Im Gegensatz dazu lernt die objektorientierte Gruppe in beiden Modulen prozedurale, sowie auch objektorientierte Themen parallel kennen.

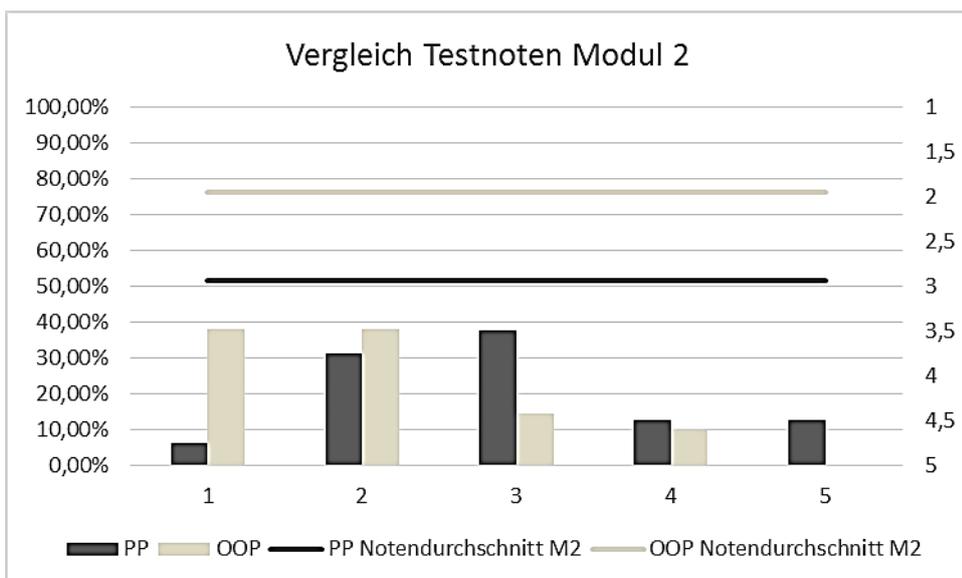
Der Vergleich der Testnoten des Moduls 1 ergibt folgende Auswertung:



**Abbildung 16: Vergleich Testnoten Modul 1 PP vs. OOP**

Hierbei ist gut abzulesen, dass die prozedurale Gruppe zwar bei den sehr guten Bewertungen besser abschneidet, doch in allen anderen Notenverteilungen schlechter. Gerade die negativen Tests fallen hierbei bei der prozeduralen Gruppe auf.

Die folgende Auswertung des Moduls 2 zeigt eine besonders interessante Verschiebung der Notenverteilung und ist besonders aussagekräftig für die Bewertung der Lehransätze, da am Ende des Moduls 2 der Wissenstand beider Gruppen wieder fast deckungsgleich sein soll. Der Test am Ende des Moduls 2 ist auch dahingehend ausgelegt, dass Fragestellungen bezüglich des gesamt erlernten Programmierstoffs gestellt werden.



**Abbildung 17: Vergleich Testnoten Modul 2 PP vs. OOP**

Anhand der Abbildung kann man gut erkennen, dass die prozedurale Gruppe am Ende des Modul 2 mit der praktischen Umsetzung der objektorientierten Techniken

mehr Probleme hatte als die objektorientierte Gruppe. Die prozedurale Gruppe schneidet in allen Bereichen schlechter ab, doch gerade der Einbruch bei der Note 1 ist besonders auffällig. Bei der prozeduralen Gruppe gibt es nur eine/n Teilnehmer/in, der/die alle objektorientierten Techniken soweit erlernt hat, dass diese/r ein sehr gut bekommen hat. Die daraus ergebenden 6,25% aller prozeduralen Teilnehmer/innen die eine sehr gute Note bekommen haben, sind um einiges geringer als die 38,1% aller objektorientierten Teilnehmer/innen die diese Note erreichen konnten. In der objektorientierten Gruppe haben 76,19% entweder eine 1 oder 2 als Testnote erhalten. In der prozeduralen Gruppe sind es hingegen nur 37,5% – was ungefähr der Hälfte entspricht. Hieraus kann man schon sehr gut ablesen, dass der objektorientierte Ansatz für diese Ausbildung der erfolgreichere Weg in Bezug auf die Tests im zweiten Modul ist.

## **5.8 Auswertung des Fragebogens**

Der Fragebogen wurde in mehreren Phasen von den Teilnehmer/innen ausgefüllt. In der ersten Phase haben die Teilnehmer/innen den Fragebogen gleich nach dem Kurs ausgefüllt. Hierbei geht es um den Eindruck, den die Teilnehmer/innen nach den 20 Kurswochen vom Unterrichtsgegenstand Programmieren vermittelt bekommen. Diese Auswertung soll den direkten Einfluss der verschiedenen Lehrmethoden aufzeigen. In der weiteren Phase wurde der Fragebogen nach der gesamten Ausbildung ausgefüllt. Hierbei haben die Teilnehmer/innen schon das Wissen der Module 1 und 2 in verschiedenen Szenarien anwenden können. Dies soll zeigen, welche Lehrmethode im weiteren Verlauf der Ausbildung die passendere war und wie sich die Teilnehmer/innen in einem praktischeren Rahmen verhalten. Außerdem zeigt der Fragebogen auch, wie sich die Teilnehmer/innen in Sachen Programmierung selbst einschätzen würden, wenn diese in die Berufswelt einsteigen. Der Fokus liegt auf alle Fälle in der ersten Phase, da sich diese hauptsächlich mit dem Kurs selbst beschäftigt. Die zweite Phase soll nur die langfristigen Auswirkungen der Unterrichtsform zeigen, wobei diese Auswirkungen durch andere Einflussfaktoren, wie z.B. das Praktikum, schon beeinflusst wurden.

### **5.8.1 Auswertung der Fragen nach dem Kurs**

#### ***5.8.1.1 Selbsteinschätzung***

##### ***Wie gut schätzen Sie ihre Kenntnisse mit Computern vor Kursbeginn ein?***

Diese Frage bezieht sich nicht auf die Programmierkenntnisse der Teilnehmer/innen, sondern auf den allgemeinen Umgang mit Computern, und wurde folgend beantwortet.

Lehrstil	Anfänger/in	Durchschnittlich	Fortgeschritten	Experte/in
Prozedural	11	4	2	0
Objektorientiert	10	7	4	0

Tabelle 13: Computerkenntnisse

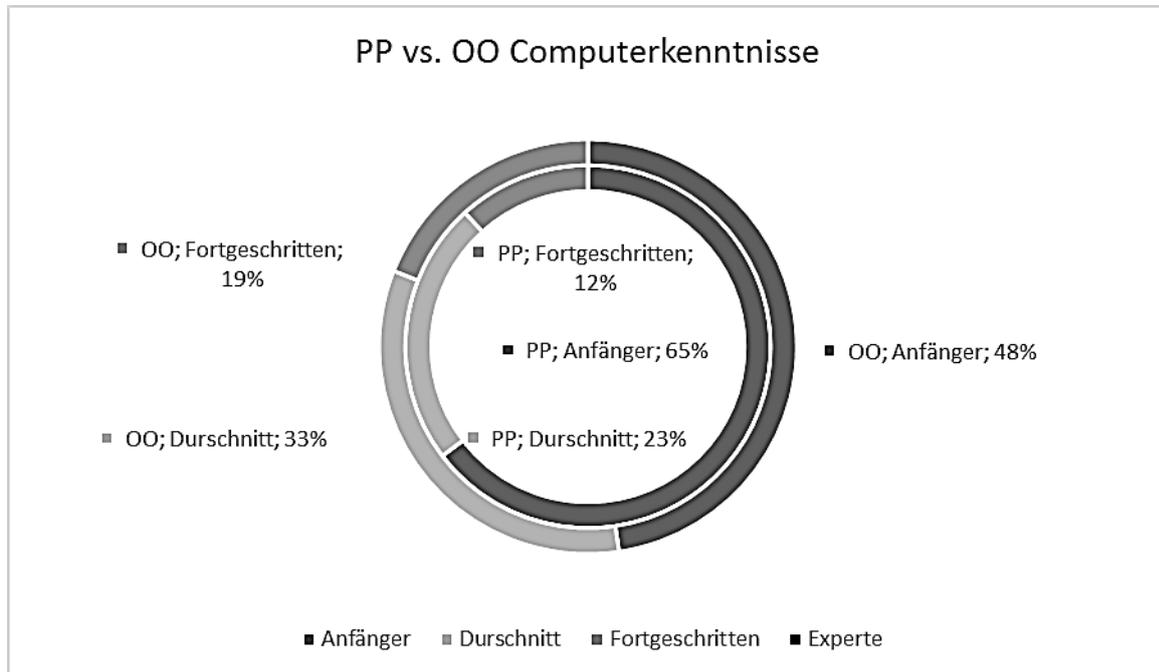


Abbildung 18: Computerkenntnisse

### ***Haben Sie schon vor dem Kurs Erfahrungen mit einer Programmiersprache gesammelt?***

Diese Frage ist unabhängig von einer Programmiersprache und wurde folgend beantwortet.

*Keine* – Ich habe noch nie programmiert

*Wenig* – Ich habe schon erste Schritte mit einer Programmiersprache erfahren

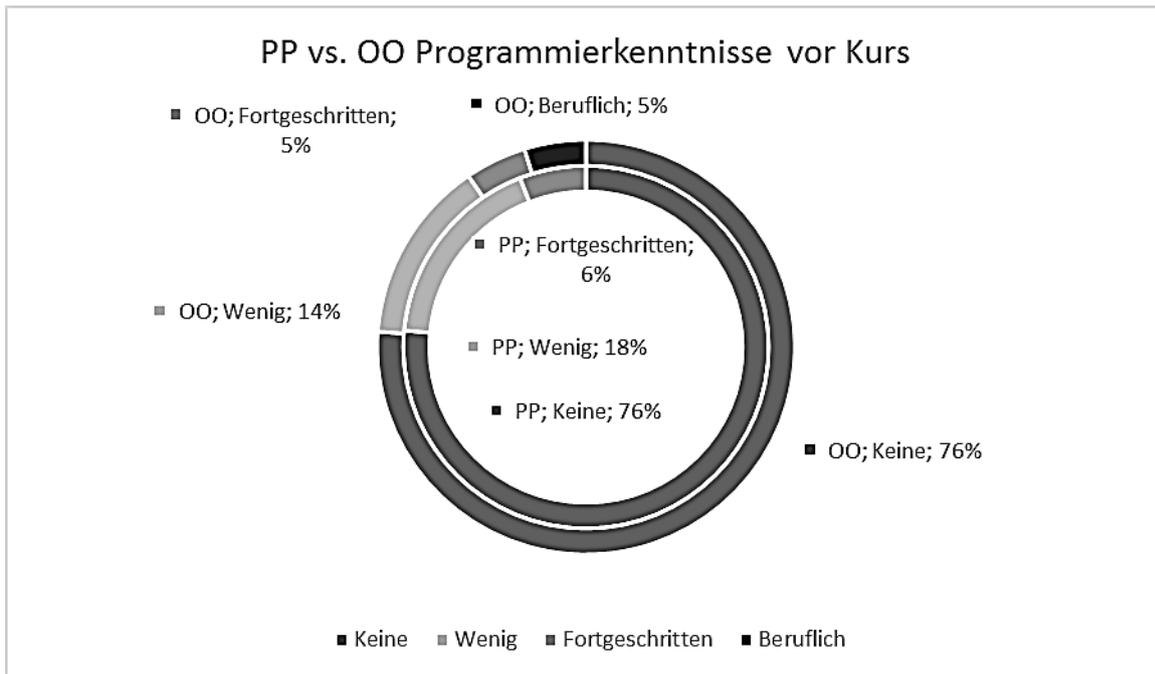
*Fortgeschritten* – Ich habe schon über einen längeren Zeitraum regelmäßig programmiert

*Beruflich* – Ich habe schon in einem Beruf meine Programmierfähigkeiten unter Beweis stellen dürfen

Auf die Kategorie der Programmiersprache, ob z.B. prozedural, funktional, logikorientiert oder objektorientiert, wurde nicht eingegangen.

Lehrstil	Keine	Wenig	Fortgeschritten	Beruflich
Prozedural	13	3	1	0
Objektorientiert	16	3	1	1

Tabelle 14: Programmierkenntnisse vor Kurs



**Abbildung 19: Programmierkenntnisse vor Kurs**

**Wie schätzen Sie ihre Programmierkenntnisse nach Abschluss des Kurses ein?**

Die Teilnehmer/innen bewerten selbständig ihre vom Kurs erlangten Programmierkenntnisse. Diese Frage ist allgemein gehalten und bezieht sich nicht auf die prozeduralen oder objektorientierten Gebiete der Programmierung. Hierbei geht es nur darum ob die Teilnehmer/innen sich eine Umsetzung von Programmierproblemen zutrauen würden.

Lehrstil	Anfänger	Kenner	Fortgeschritten	Experte
<b>Prozedural</b>	0	6	7	4
<b>Objektorientiert</b>	1	9	8	3

**Tabelle 15: Programmierkenntnisse nach Kurs**

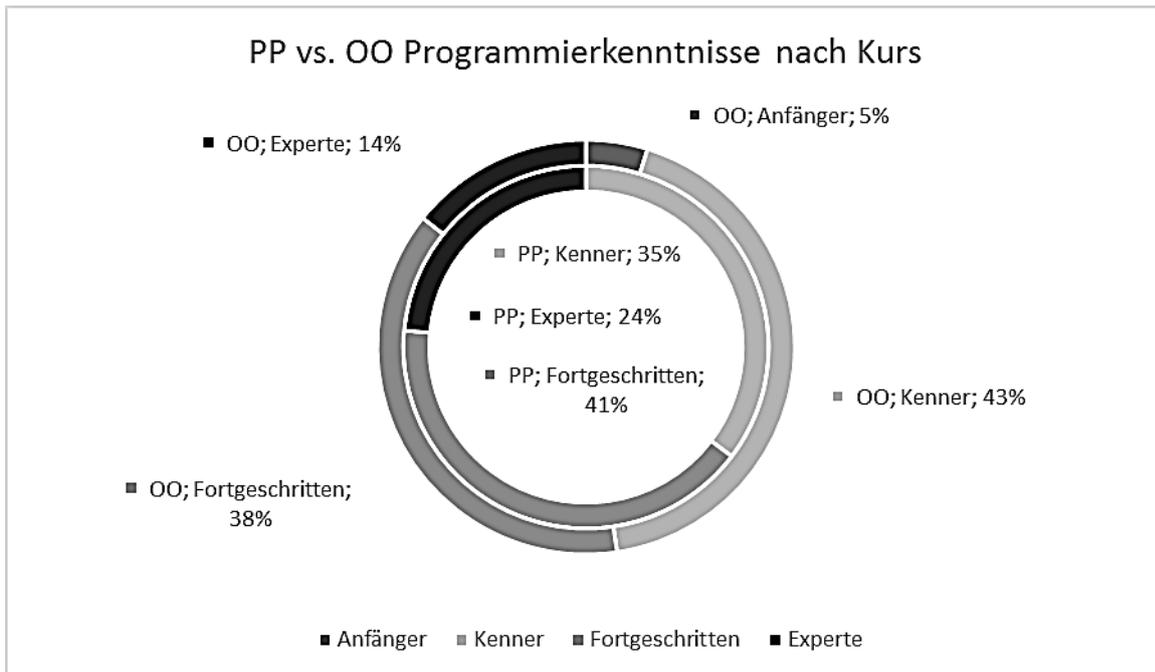


Abbildung 20: Programmierkenntnisse nach Kurs

### Verstehen Sie nun die Grundlagen der objektorientierten Programmierung?

Diese Frage bezieht sich auf die Besonderheiten der objektorientierten Programmierung, welche auch das endgültige Ziel des Kurses darstellen.

Lehrstil	Sehr wenig	Wenig	Gut	Sehr gut
Prozedural	4	7	5	1
Objektorientiert	3	9	6	3

Tabelle 16: Objektorientierte Grundlagen

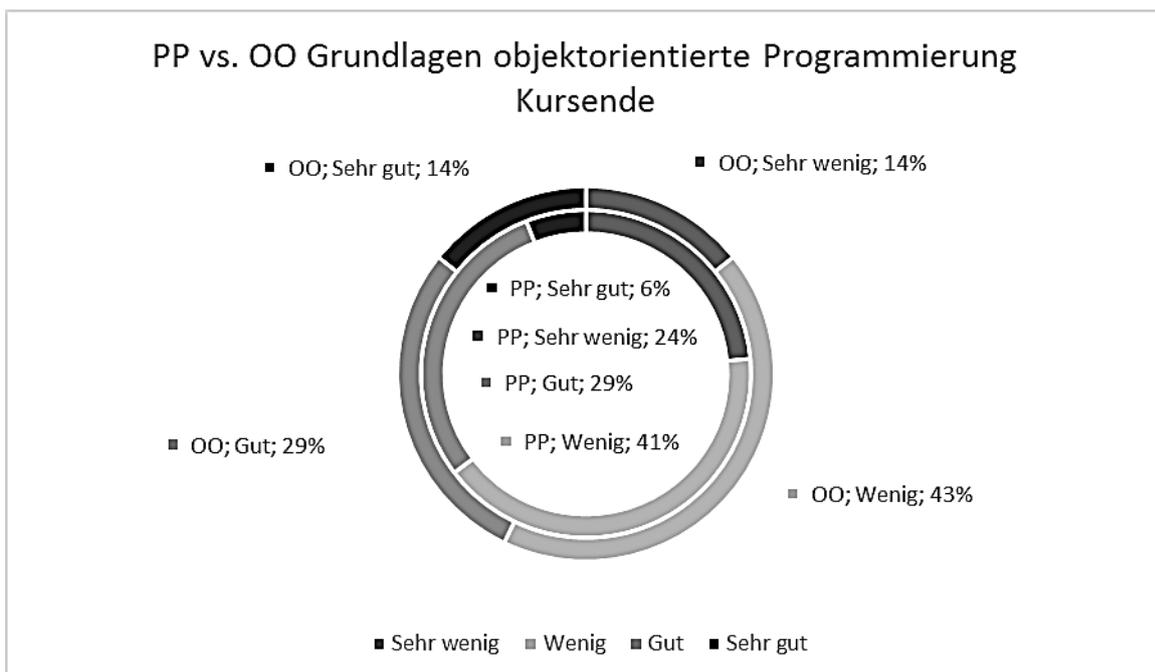


Abbildung 21: Objektorientierte Grundlagen

### 5.8.1.2 Kursbewertung

#### Haben Sie genügend Unterstützungen während des Unterrichts erhalten?

Haben die Trainer/innen ihre Fragen zufriedenstellend beantwortet und ihnen bei Problemen weitergeholfen.

Lehrstil	Sehr wenig	Wenig	Genug	Perfekt
Prozedural	3	4	6	4
Objektorientiert	1	7	9	4

Tabelle 17: Hilfestellung Trainer/innen

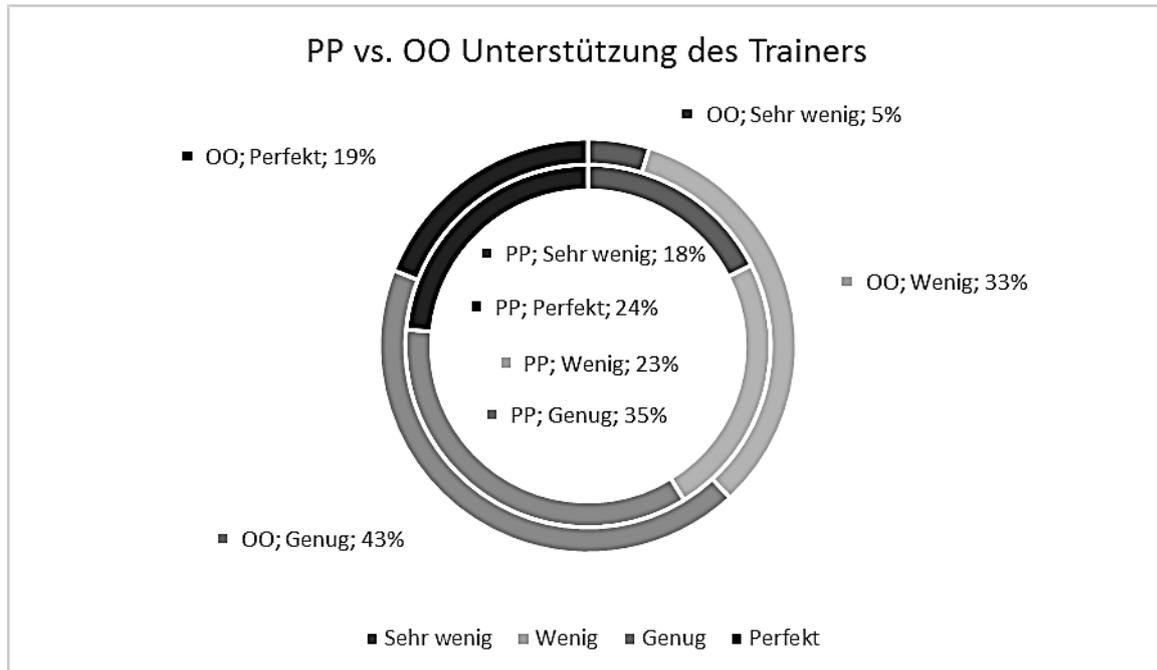


Abbildung 22: Hilfestellung Trainer/innen

#### Waren genügend Übungsbeispiele für den Unterricht vorhanden?

Diese Frage bezieht sich auf die vorhandenen Übungsbeispiele während des Kurszeitraumes. Hier wird festgestellt, ob die Teilnehmer/innen genügend Beispiele zum Üben zur Verfügung gestellt bekommen.

Lehrstil	Sehr wenig	Wenig	Genug	Perfekt
Prozedural	2	2	5	8
Objektorientiert	1	4	9	7

Tabelle 18: Übungsbeispiele



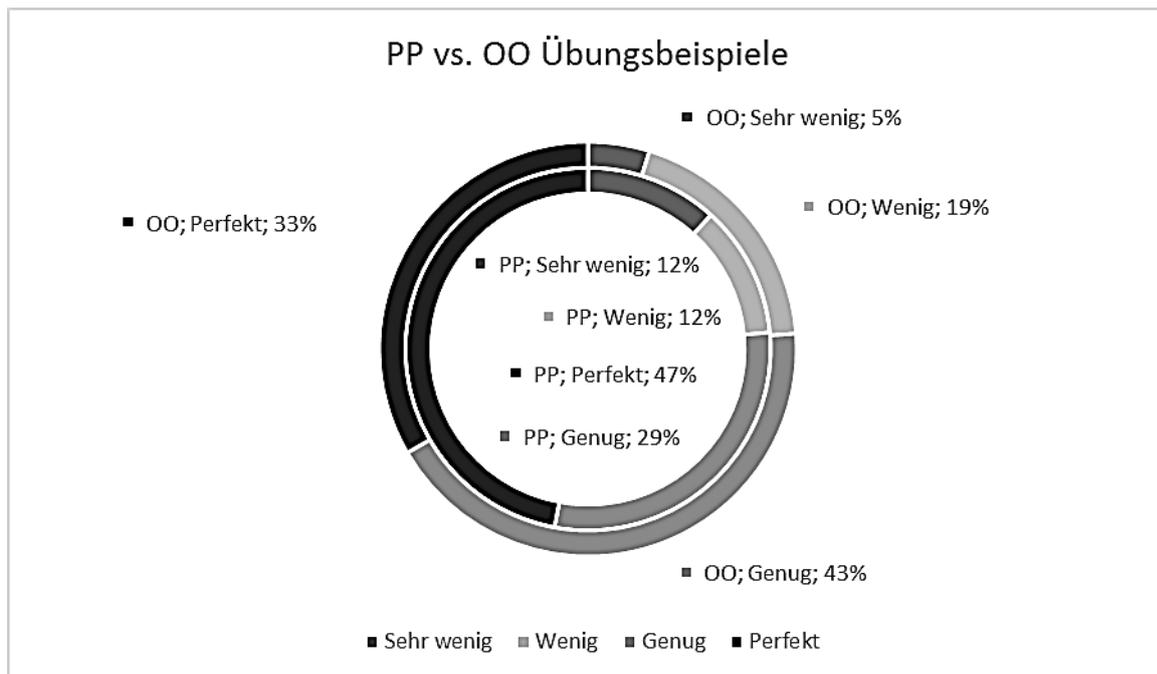


Abbildung 23: Übungsbeispiele

### 5.8.1.3 Unterrichtsmethoden

*Hätten Sie gerne mehr Frontalunterricht gehabt?*

Lehrstil	Nein	Bisschen mehr	Mehr	Viel mehr
Prozedural	3	5	4	4
Objektorientiert	4	9	7	1

Tabelle 19: Menge Frontalunterricht

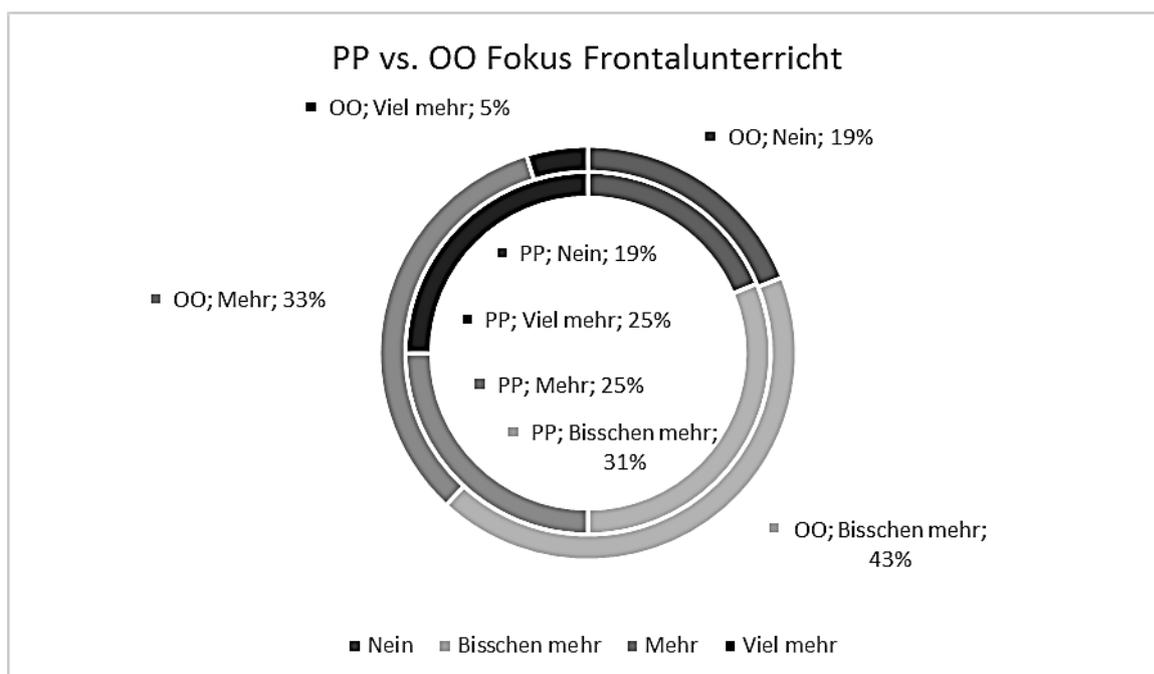


Abbildung 24: Menge Frontalunterricht

**Wie hilfreich war für Sie der Frontalunterricht?**

Lehrstil	Sehr wenig	Wenig	Ausreichend	Perfekt
Prozedural	1	3	8	4
Objektorientiert	0	6	10	5

Tabelle 20: Hilfe Frontalunterricht

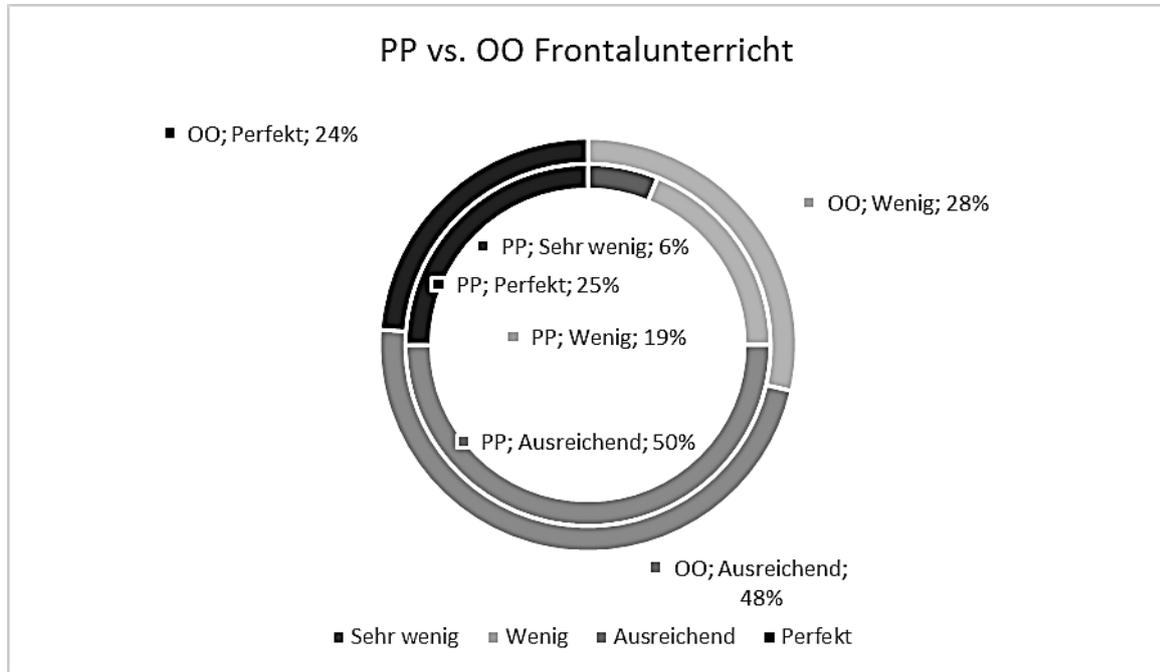
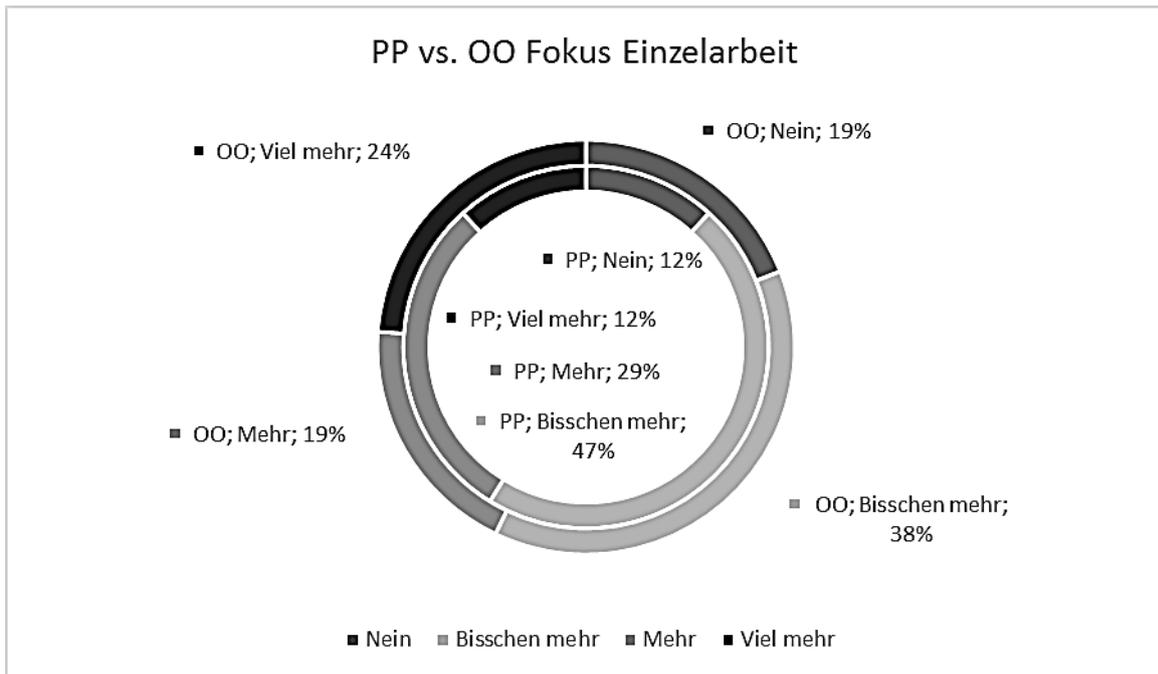


Abbildung 25: Hilfe Frontalunterricht

**Hätten Sie gerne mehr Einzelarbeiten gehabt?**

Lehrstil	Nein	Bisschen mehr	Mehr	Viel mehr
Prozedural	2	8	5	2
Objektorientiert	4	8	4	5

Tabelle 21: Menge Einzelarbeit

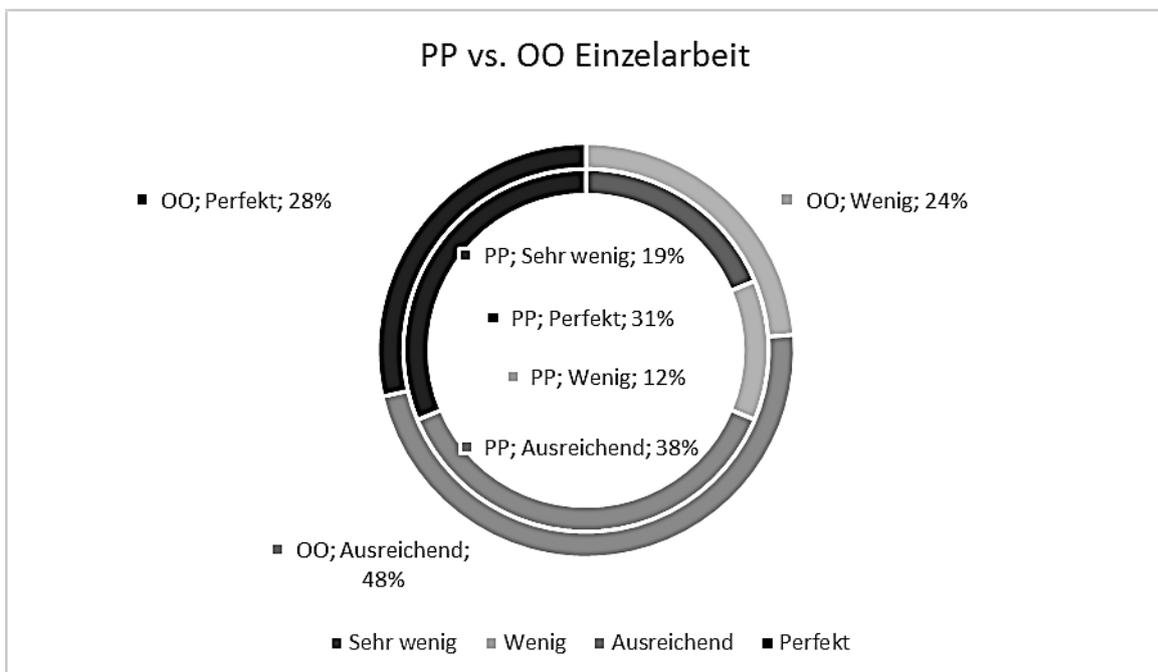


**Abbildung 26: Menge Einzelarbeit**

**Wie hilfreich waren für Sie die Einzelarbeiten?**

Lehrstil	Sehr wenig	Wenig	Ausreichend	Perfekt
Prozedural	3	2	6	5
Objektorientiert	0	5	10	6

**Tabelle 22: Hilfe Einzelarbeit**



**Abbildung 27: Hilfe Einzelarbeit**

### Hätten Sie gerne mehr Gruppenarbeiten gehabt?

Lehrstil	Nein	Bisschen mehr	Mehr	Viel mehr
Prozedural	4	5	5	3
Objektorientiert	1	13	6	1

Tabelle 23: Menge Gruppenarbeit

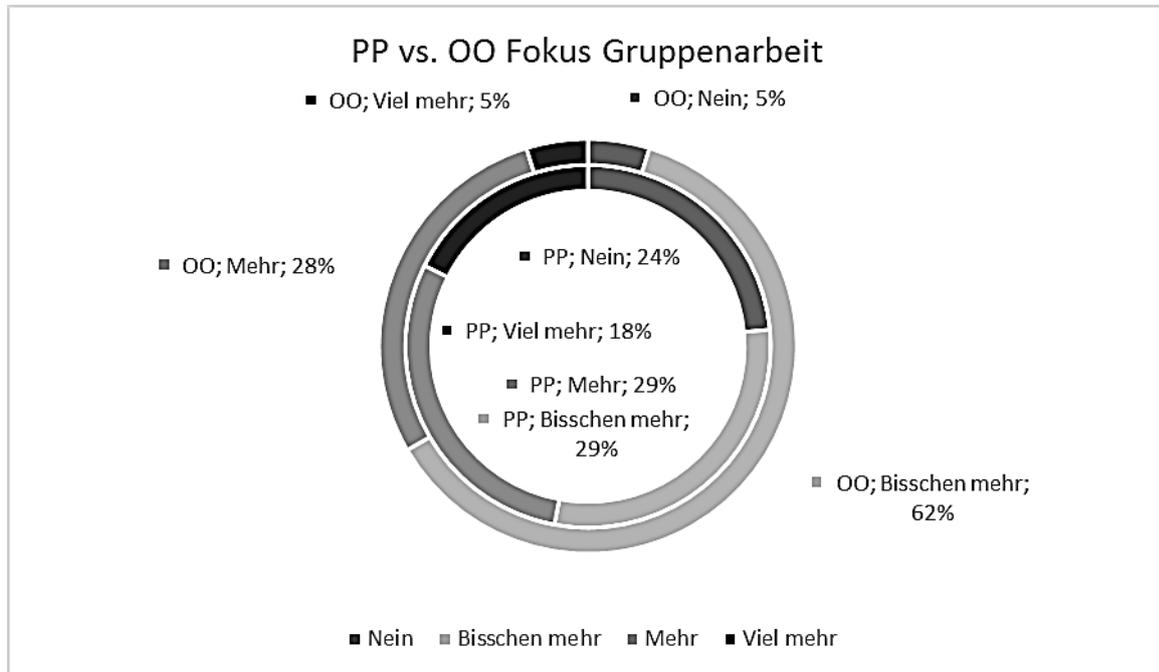


Abbildung 28: Menge Gruppenarbeit

### Wie hilfreich waren für Sie die Gruppenarbeiten?

Lehrstil	Sehr wenig	Wenig	Ausreichend	Perfekt
Prozedural	2	6	5	4
Objektorientiert	0	4	11	6

Tabelle 24: Hilfe Gruppenarbeit

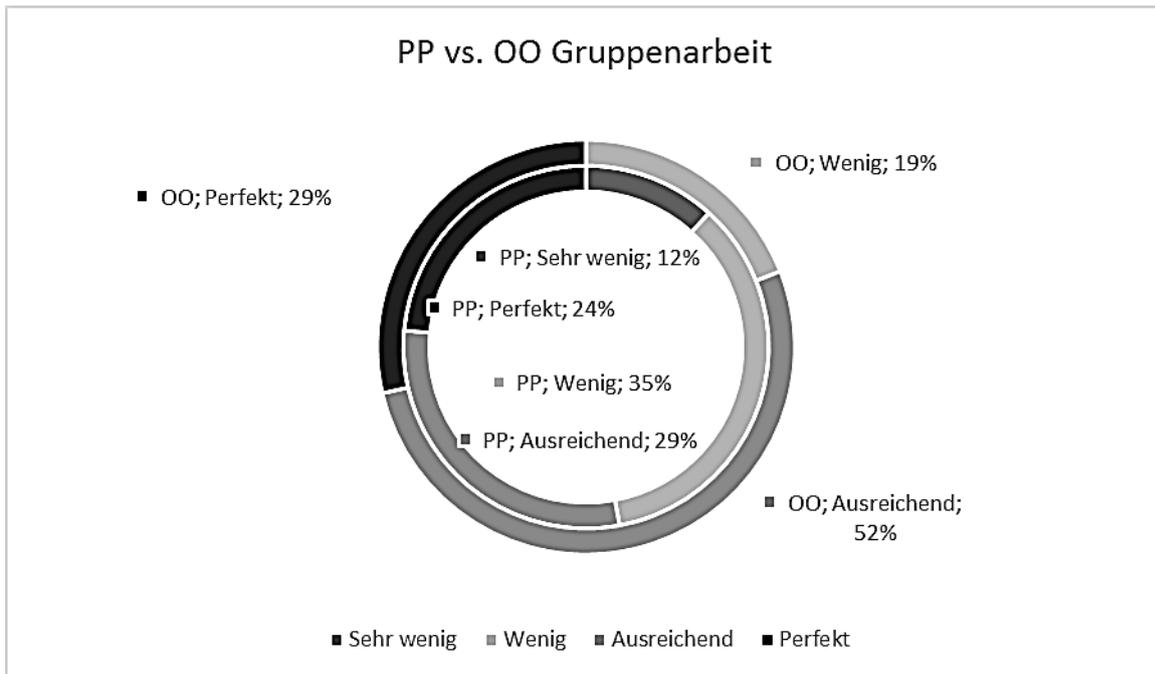


Abbildung 29: Hilfe Gruppenarbeit

### Hätten Sie gerne mehr Partnerarbeiten gehabt?

Lehrstil	Nein	Bisschen mehr	Mehr	Viel mehr
Prozedural	4	5	7	1
Objektorientiert	1	12	6	2

Tabelle 25: Menge Partnerarbeit

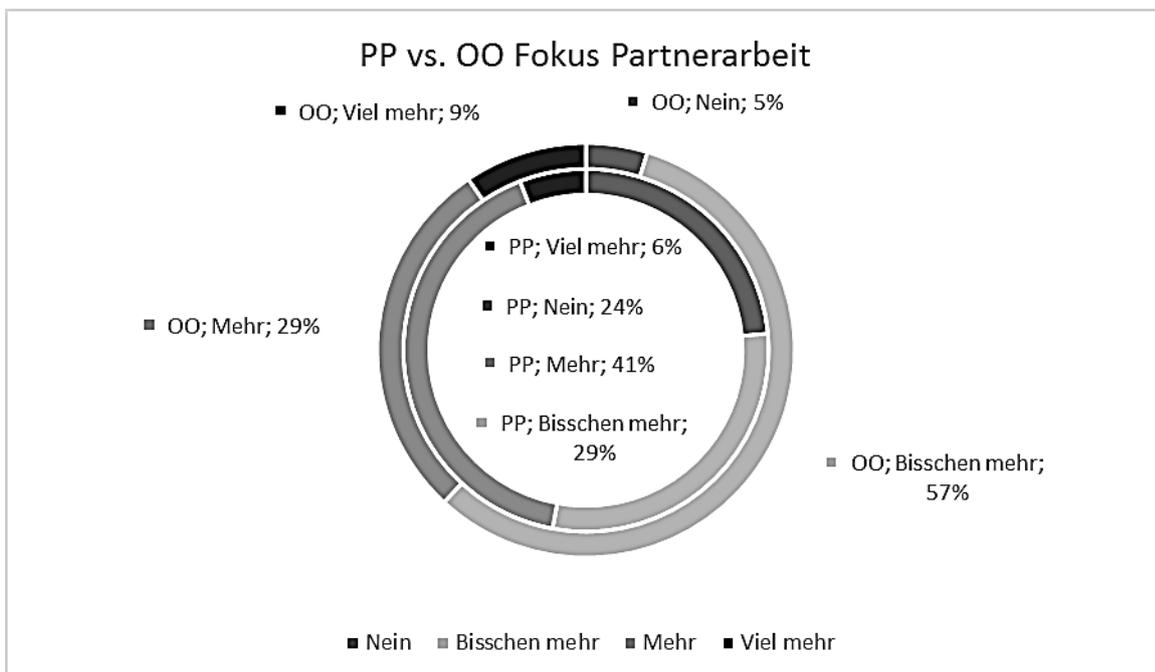


Abbildung 30: Menge Partnerarbeit

**Wie hilfreich war für Sie die Partnerarbeit?**

Lehrstil	Sehr wenig	Wenig	Ausreichend	Perfekt
Prozedural	0	4	5	7
Objektorientiert	0	2	10	9

Tabelle 26: Hilfe Partnerarbeit

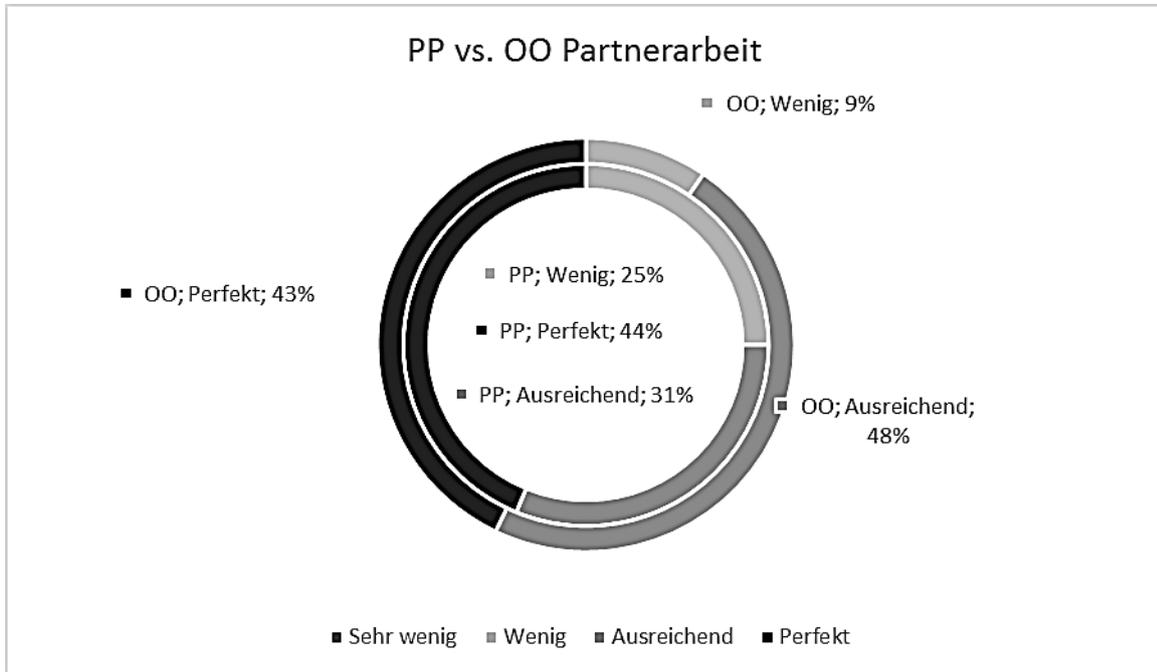


Abbildung 31: Hilfe Partnerarbeit

**Hätten Sie gerne mehr Referate gehabt?**

Lehrstil	Nein	Bisschen mehr	Mehr	Viel mehr
Prozedural	4	5	6	2
Objektorientiert	2	6	10	3

Tabelle 27: Menge Referat

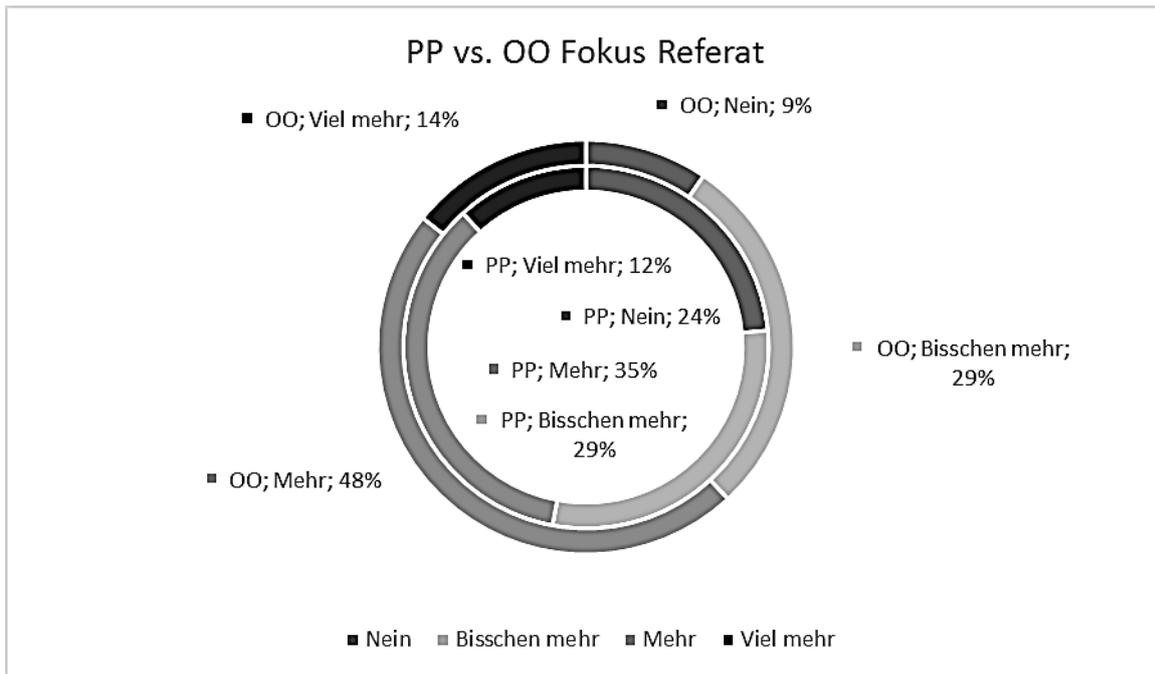


Abbildung 32: Menge Referat

*Wie hilfreich waren für Sie die Referate?*

Lehrstil	Sehr wenig	Wenig	Ausreichend	Perfekt
Prozedural	2	5	4	5
Objektorientiert	1	7	5	8

Tabelle 28: Hilfe Referat

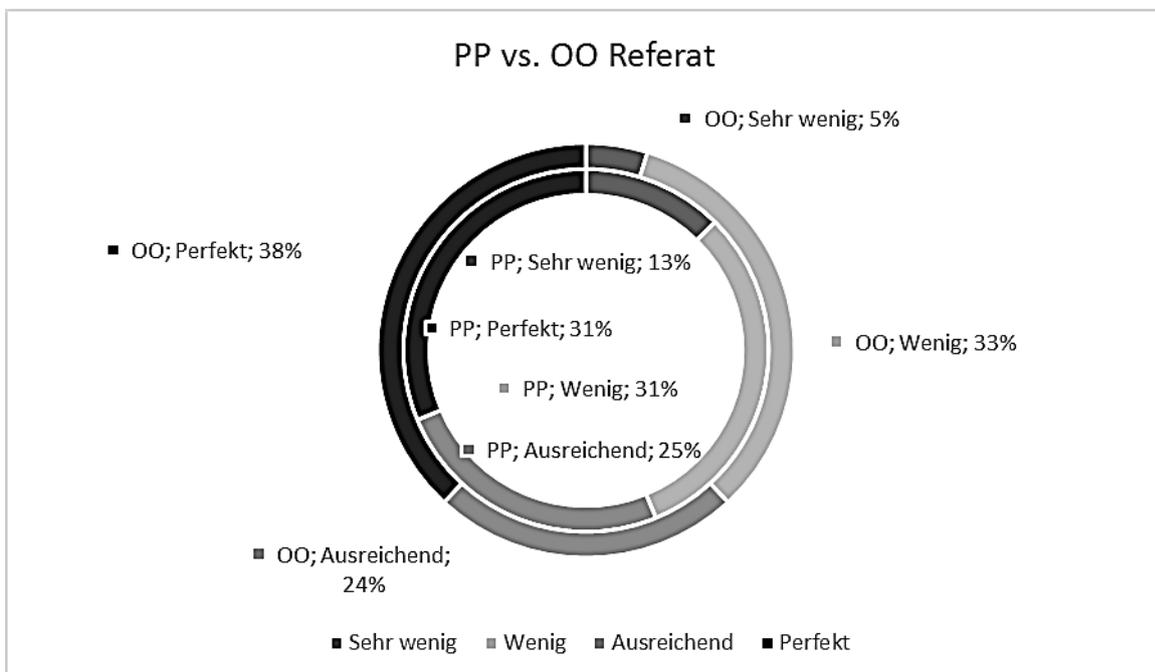


Abbildung 33: Hilfe Referat

### 5.8.1.4 Unterrichtsmittel

Wie hilfreich waren folgende Medien in der prozeduralen Gruppe?

	Gar nicht	Eher nicht	Eher schon	Sehr hilfreich
Whiteboard, Tafel, Flipchart	0	2	5	9
Bücher, Fachliteratur	2	3	6	5
Videos	3	6	5	2
Online Tutorials	1	2	5	8
Trainerunterlagen	1	3	6	6

Tabelle 29: Medien prozedurale Gruppe

Daraus ergibt sich folgende Aufteilung:

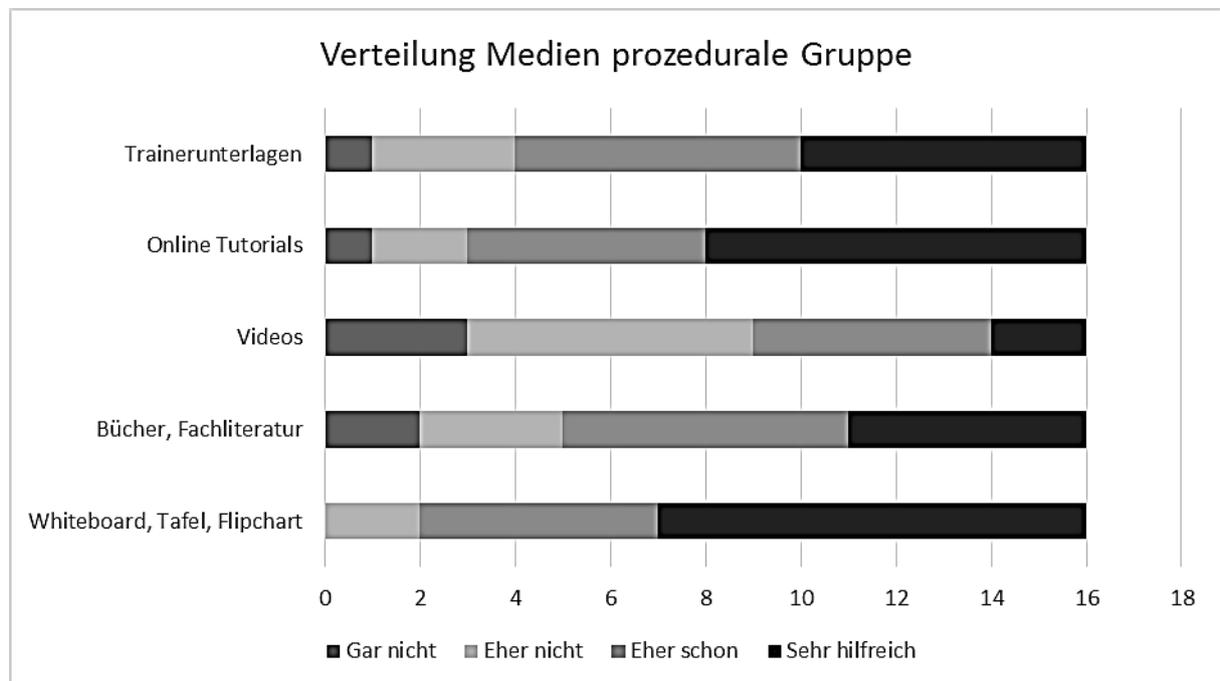


Abbildung 34: Medien prozedurale Gruppe

Wie hilfreich waren folgende Medien in der objektorientierten Gruppe?

	Gar nicht	Eher nicht	Eher schon	Sehr hilfreich
Whiteboard, Tafel, Flipchart	0	2	7	12
Bücher, Fachliteratur	0	6	7	8
Videos	2	5	10	4
Online Tutorials	0	5	9	7
Trainerunterlagen	1	3	5	12

Tabelle 30: Medien objektorientierte Gruppe



Daraus ergibt sich folgende Aufteilung:

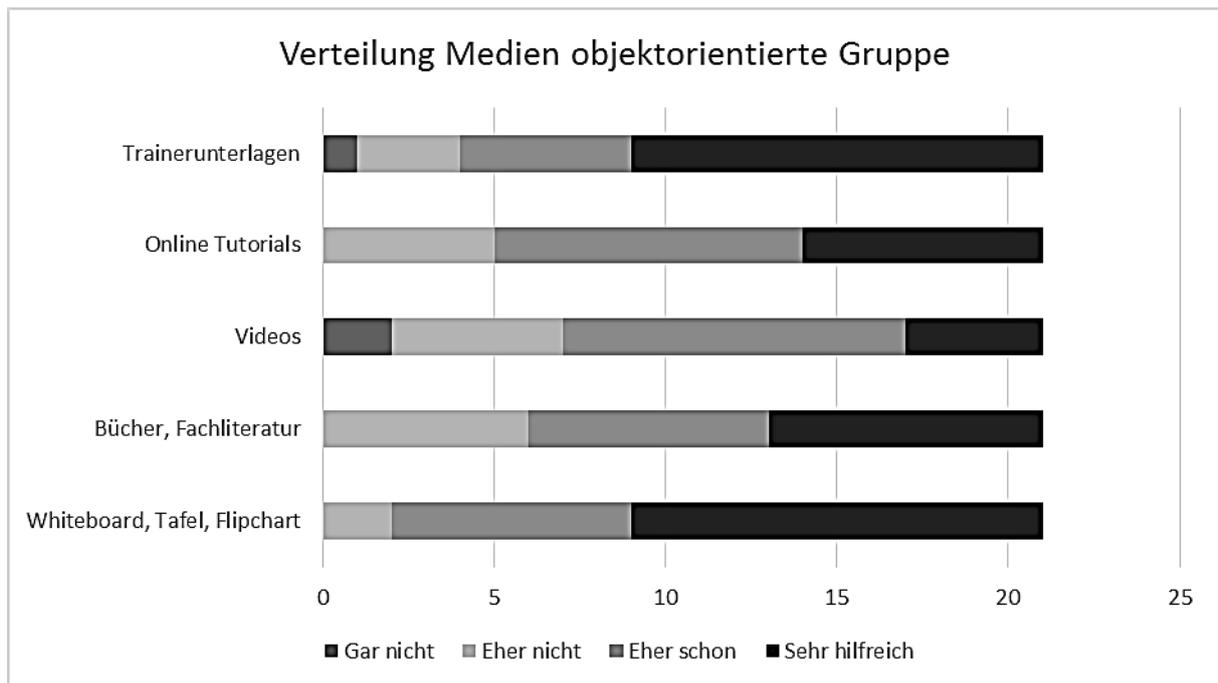


Abbildung 35: Medien objektorientierte Gruppe

### 5.8.1.5 Allgemeine Fragen

*Hat der Kurs ihr Interesse an der Programmierung erhöht?*

Lehrstil	Nicht	Wenig	Mehr	Viel
Prozedural	2	4	3	8
Objektorientiert	1	4	9	7

Tabelle 31: Programmierinteresse

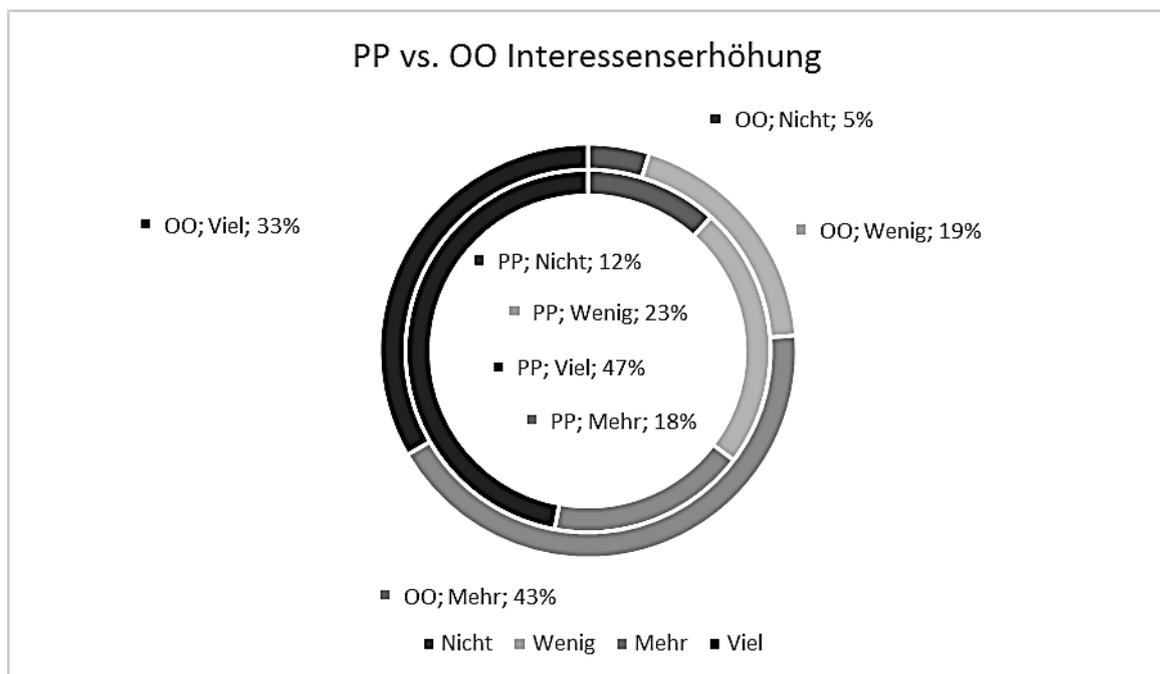


Abbildung 36: Programmierinteresse

**Wie hoch schätzen Sie den Zeitaufwand, den Sie in Ihrer Freizeit in die Programmierausbildung investiert haben ab?**

Als Richtwert haben die Teilnehmer/innen folgende Tabelle bekommen.

*Kein* – < 1 Stunde die Woche

*Wenig* – 1 - 3 Stunden die Woche

*Mehr* – 3 - 6 Stunden die Woche

*Viel* – > 6 Stunden die Woche

Lehrstil	Kein	Wenig	Mehr	Viel
Prozedural	5	6	4	2
Objektorientiert	6	7	3	5

Tabelle 32: Zeitaufwand Freizeit

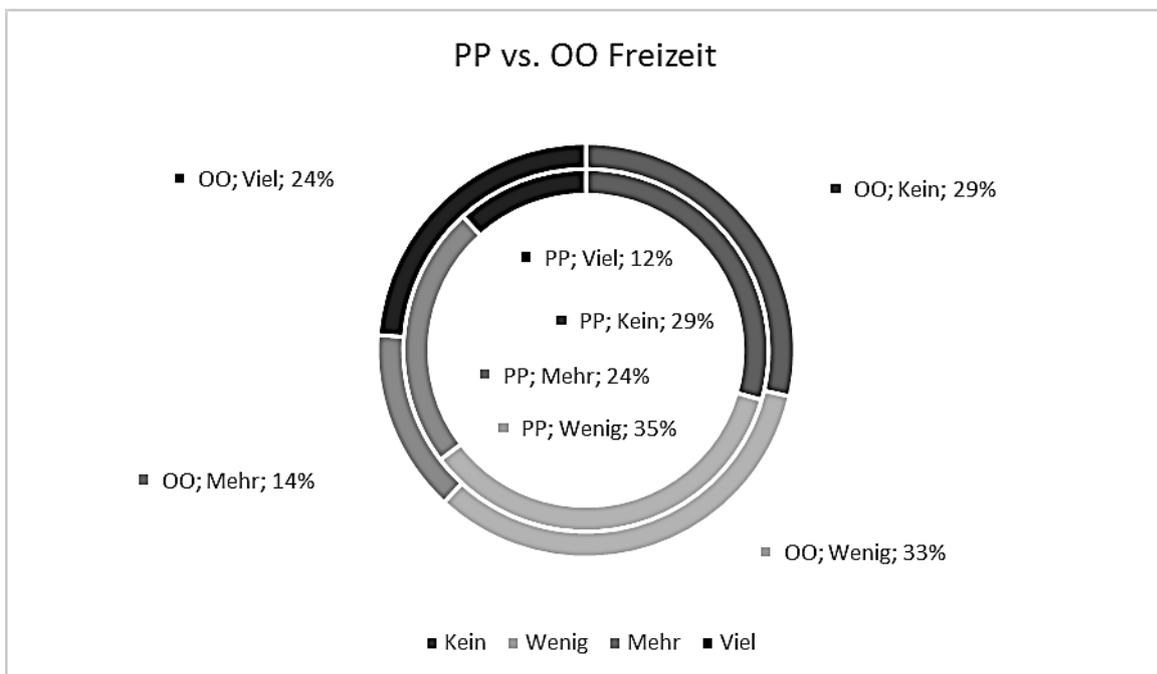


Abbildung 37: Zeitaufwand Freizeit

### 5.8.2 Auswertung der Fragen in den weiteren Modulen

**Wie sehr hat Ihnen der Kurs über Modul 1 und 2 in Bezug auf das LAP-Projekt geholfen?**

Diese Frage soll zeigen, ob die objektorientierten Grundlagen, die für das LAP-Projekt sehr wichtig sind, längerfristig verstanden wurden. Gerade in Bezug auf die theoretische Planung und die praktischen Grundlagen, da diese besonders in dieser Phase benötigt werden.

Lehrstil	Sehr wenig	Wenig	Ausreichend	Perfekt
Prozedural	3	3	6	5
Objektorientiert	0	5	7	9

Tabelle 33: LAP-Projektunterstützung

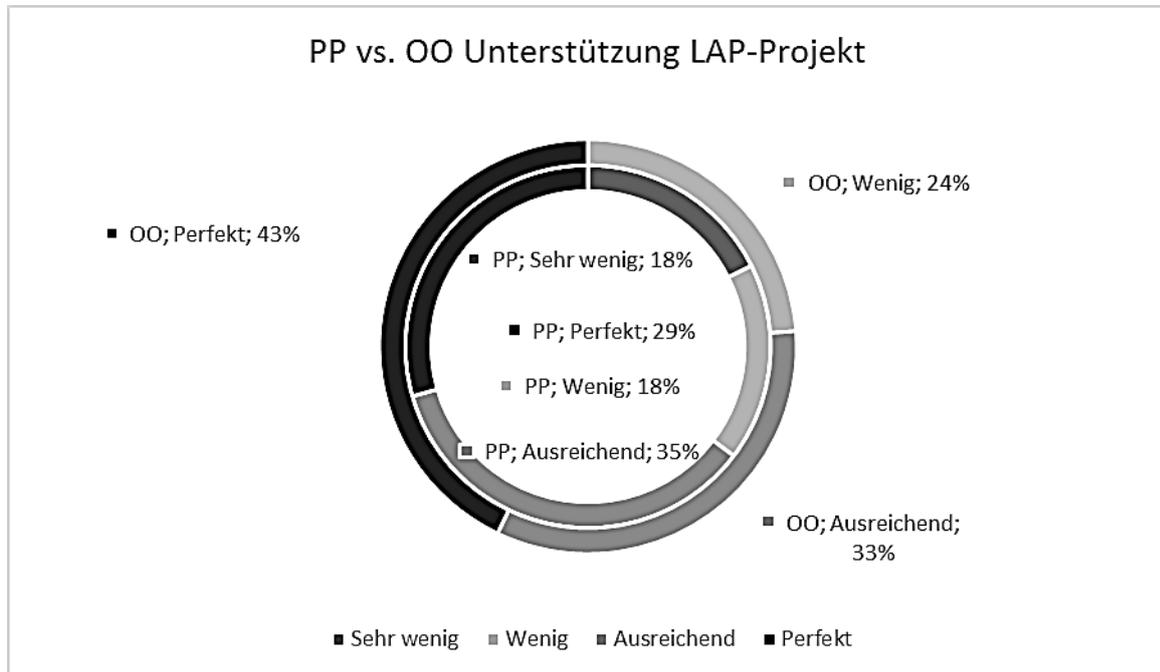


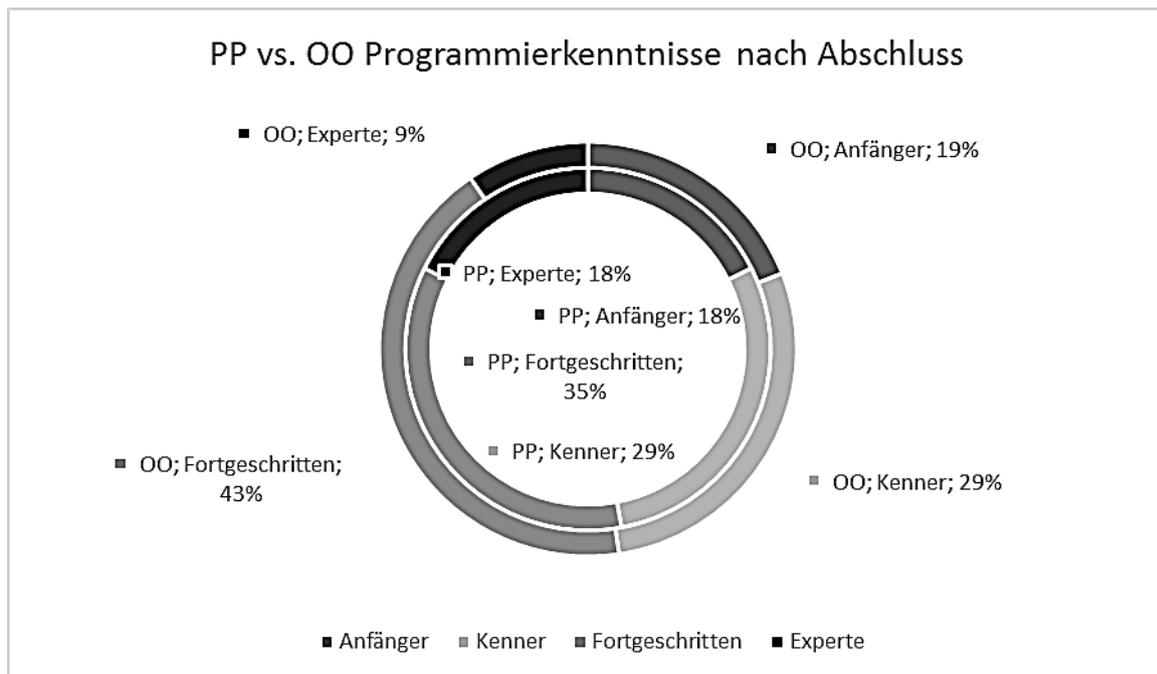
Abbildung 38: LAP-Projektunterstützung

### 5.8.2.1 Selbsteinschätzung

Wie schätzen Sie ihre Programmierkenntnisse nach Abschluss der Ausbildung ein?

Lehrstil	Anfänger	Kenner	Fortgeschritten	Experte
Prozedural	3	5	6	3
Objektorientiert	4	6	9	2

Tabelle 34: Programmierkenntnisse nach Ausbildung



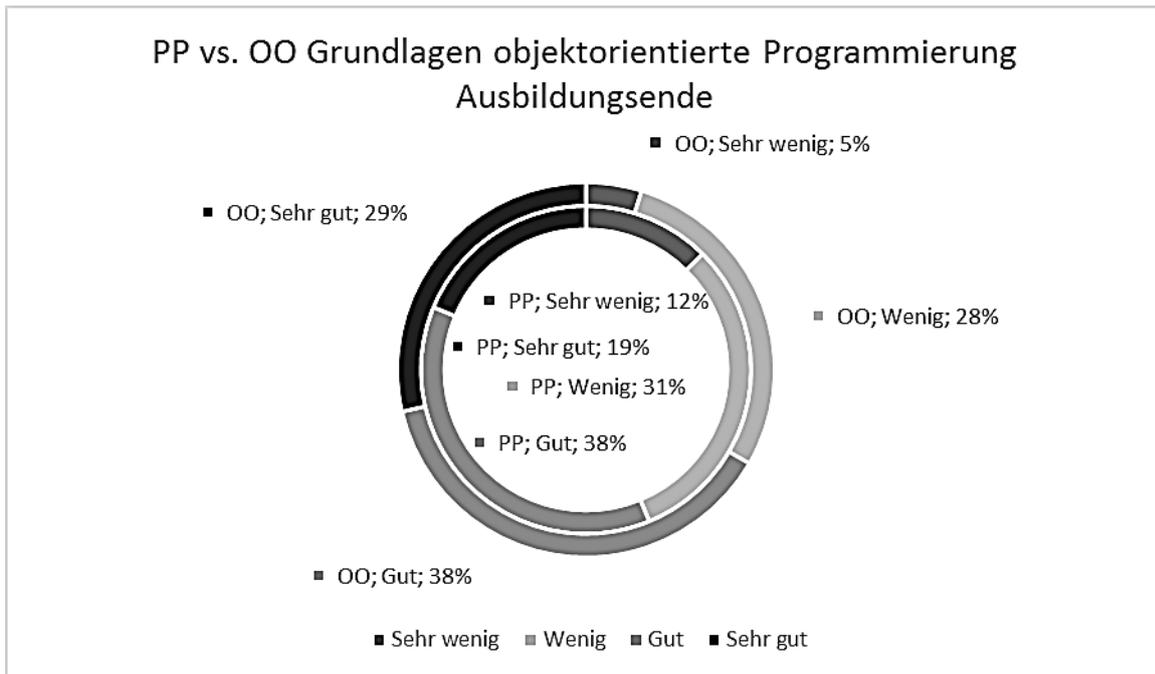
**Abbildung 39: Programmierkenntnisse nach Ausbildung**

***Verstehen Sie nun die Grundlagen der objektorientierten Programmierung?***

Diese Frage bezieht sich auf die Besonderheiten der objektorientierten Programmierung, welche auch die endgültigen Ziele des Kurses darstellen. Diesmal findet die Frage nach der gesamten Ausbildung statt, wobei die Teilnehmer/innen auch schon praktischere Module wie AKT und das Berufspraktikum hinter sich gebracht haben.

Lehrstil	Sehr wenig	Wenig	Gut	Sehr gut
<b>Prozedural</b>	2	5	6	3
<b>Objektorientiert</b>	1	6	8	6

**Tabelle 35: OO-Grundlagen nach Ausbildung**



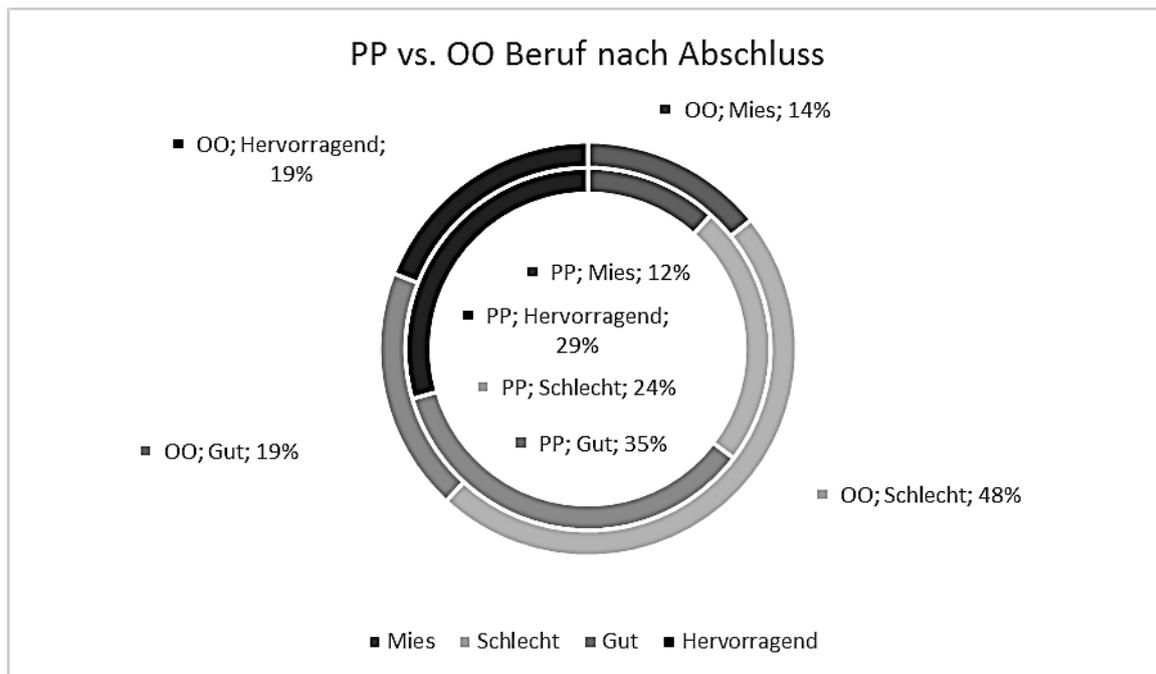
**Abbildung 40: OO-Grundlagen nach Ausbildung**

**Wie schätzen Sie ihre Chancen ein, eine berufliche Stelle als Programmierer/in erlangen zu können ein?**

Diese Frage ist bezogen auf die Programmierkenntnisse und unabhängig davon, ob man auch tatsächlich Programmierer/in als Beruf ausüben möchte.

Lehrstil	Mies	Schlecht	Gut	Hervorragend
Prozedural	2	4	6	5
Objektorientiert	3	10	4	4

**Tabelle 36: Berufschancen nach Ausbildung**



**Abbildung 41: Berufschancen nach Ausbildung**

## 5.8.3 Erkenntnisse nach dem Kurs

### 5.8.3.1 Selbsteinschätzung

Das Kapitel 5.8.1 Auswertung der Fragen nach dem Kurs zeigt die Auswertung des Fragebogens, der direkt nach dem Kurs von den Teilnehmern/innen ausgefüllt wurde. In Bezug auf die Selbsteinschätzung der unterschiedlichen Lehrmethoden hat sich folgendes ergeben:

Die Teilnehmer/innen der prozeduralen Gruppe sehen sich im Allgemeinen mehr als Computeranfänger/innen als die Teilnehmer/innen aus der objektorientierten Gruppe. Dieser Trend zeigt sich aber nicht direkt in der Frage, ob die einzelnen Teilnehmer/innen erste Erfahrungen mit Programmiersprachen im Allgemeinen haben. Bei dieser Frage stehen sich beide Gruppen in nichts nach, die Ergebnisse hierbei sind fast deckungsgleich. Ein/e Teilnehmer/in aus der objektorientierten Gruppe hat schon vor Kursbeginn berufliche Programmiererfahrungen gesammelt.

Direkt nach dem Kurs sehen sich die Teilnehmer/innen der prozeduralen Gruppe in einer positiveren Stellung zu ihren Programmierfähigkeiten als die objektorientierte Gruppe. Daraus ergibt sich, dass sich die prozedurale Gruppe selbst fähiger in Sachen Programmierung einschätzt als die objektorientierte Gruppe. Bei der Detailfrage, wie gut man die Grundlagen der objektorientierten Programmierung beherrscht und somit Konzepte der Kapselung, Vererbung und Polymorphie verstanden hat, wendet sich das Blatt ein bisschen. Hier ist die Tendenz leicht positiver bei der objektorientierten Gruppe, die sich diese objektorientierten Grundlagen eher zutraut.

Dieses Ergebnis ist aber auch verständlich, da diese Gruppe einen fast doppelt so langen Zeitraum hat wie die prozedurale Gruppe, um sich mit diesen Themen zu auseinandersetzen.

Zusammengefasst kann man sagen, dass die prozedurale Gruppe sich schlechter in den allgemeinen Computerkenntnissen vor dem Kurs eingeschätzt hat, aber besser in den allgemeinen Programmierkenntnissen nach dem Kurs. Die objektorientierte Gruppe hat sich leicht besser in den allgemeinen Computerkenntnissen vor dem Kurs eingeschätzt, aber nach dem Kurs ein bisschen schlechter in den allgemeinen Programmierkenntnissen, wiederum aber besser in den objektorientierten Grundlagen.

### **5.8.3.2 Kursbewertung**

Anhand der Fragen bezüglich der Kursbewertung und im speziellen auf die Unterstützung der Trainer/innen bezogen, ergibt sich folgendes:

Mit 18% fühlen sich die Teilnehmer/innen der prozeduralen Gruppe von den Trainer/innen zu wenig unterstützt und mehr allein gelassen als die 5% der Teilnehmer/innen der objektorientierten Gruppe. Im Allgemeinen finden 41% der Teilnehmer/innen der prozeduralen Gruppe sich zu wenig unterstützt und 38% der objektorientierten Gruppe. Beide Werte sind nicht sehr gut und sollten in kommenden Kursen verbessert werden. Diese nicht sehr positive Bilanz wird durch die nächste Frage bezüglich der Anzahl der Übungsbeispiele ein bisschen verbessert, wo nur 12% der prozeduralen Gruppe und 5% der objektorientierten Gruppe die Anzahl der Übungsbeispiele als viel zu wenig empfinden. Für 76% der prozeduralen und 76% der objektorientierten Gruppe sind es genügen Übungsbeispiele.

Anhand der Unterrichtsmethoden hat sich folgendes Bild ergeben:

#### **Frontalunterricht**

Die Dauer des Frontalunterrichts ist für 50% der Teilnehmer/innen der prozeduralen Gruppe fast perfekt. Die restlichen 50% der Gruppe wünschen sich gerne mehr Frontalunterricht. Somit ist hier keine Tendenz in eine Richtung auszumachen.

Bei der objektorientierten Gruppe sind 62% mit der Menge fast ganz zufrieden und 38% wünschen sich gerne mehr Frontalunterricht.

Mit der Qualität des Frontalunterrichts sind die Teilnehmer/innen beider Gruppen – 75% in der prozeduralen Gruppe bzw. 72% in der objektorientierten Gruppe – zufrieden. Von diesen über 70% empfinden 25% der prozeduralen Gruppe bzw. 24% der objektorientierten Gruppe die Qualität als perfekt.

### ***Einzelarbeit***

41% der Teilnehmer/innen der prozeduralen Gruppe bzw. 43% der Teilnehmer/innen der objektorientierten Gruppe hätten gerne mehr Einzelarbeiten. Dieses Verlangen liegt wahrscheinlich auch an der Qualität der Übungsaufgaben, da 69% der prozeduralen Gruppe und 76% der objektorientierten Gruppe die Einzelarbeiten als hilfreich bezeichnen.

### ***Partnerarbeit***

Mit der Anzahl an Partnerarbeiten sind in der prozeduralen Gruppe 43% und in der objektorientierten Gruppe 62% fast komplett zufrieden. Nur 5% der objektorientierten Gruppe lehnen mehr Partnerarbeit ab.

75% der prozeduralen Gruppe und 91% der objektorientierten Gruppe finden die Partnerarbeit ausreichend bis perfekt hilfreich. Dies ist gerade in der objektorientierten Gruppe ein sehr hoher Wert und es sollte in zukünftigen Übungen die Partnerarbeit noch mehr fokussiert werden um die Teilnehmer/innen-Zufriedenheit zu steigern.

### ***Gruppenarbeit***

Die prozedurale Gruppe ist zu 53% zufrieden mit der Anzahl der Gruppenarbeiten. Die objektorientierte Gruppe ist mit 67% zufrieden mit der Anzahl der Gruppenarbeiten. Somit sind bei beiden Gruppen mehr als die Hälfte zufrieden mit der Anzahl der Gruppenarbeiten, wobei diese im Unterricht viel weniger als die Einzelarbeiten vorkommt.

In der prozeduralen Gruppe sind 53% bzw. in der objektorientierten Gruppe sind 83% der Teilnehmer/innen mit der Qualität zufrieden. Dies ist ein sehr starker Unterschied, wobei anhand der Unterrichtsbeobachtung hierbei schon einmal erwähnt werden kann, dass die Gruppenzusammenarbeit in der prozeduralen Gruppe im Allgemeinen nicht so gut funktioniert hat wie in der objektorientierten Gruppe. Somit spielt hierbei der starke Zusammenhalt in der objektorientierten Gruppe wahrscheinlich auch eine große Rolle in dieser unterschiedlichen Beurteilung.

### ***Referat***

Referate sind sehr zeitaufwändig und werden in den Modulen 1 und 2, wo der evaluierte Kurs stattfindet nur sporadisch verwendet. Erst in den späteren Modulen findet eine Verstärkung dieser Methodik statt. In der prozeduralen Gruppe haben sich 47% der Teilnehmer/innen mehr Referate gewünscht, wobei 24% dieser Gruppe mehr Referate definitiv ablehnen. In der objektorientierten Gruppe haben sich 62% der Teilnehmer/innen mehr Referate gewünscht, wobei hier auch nur 9% Referate definitiv ablehnen. Hieraus lässt sich gut erkennen, dass die objektorientierte Gruppe tendenziell lieber Referate hält als die prozedurale Gruppe.



Bei den eingesetzten Unterrichtsmitteln haben sich folgende Präferenzen und Abneigungen in den folgenden Gruppen ergeben:

### ***Unterrichtsmittel prozedurale Gruppe***

Am hilfreichsten sieht die Gruppe die Kategorie Whiteboard, Tafel und Flipchart, gleich gefolgt von Online-Tutorials. Wobei das Whiteboard, die Tafel und das Flipchart hauptsächlich für den Frontalunterricht verwendet werden und die Online-Tutorials eher bei den Übungen die die Teilnehmer/innen selbstständig erfüllen müssen. Am schlechtesten sind die Videos bei den Teilnehmer/innen angekommen, wobei erwähnt werden sollte, dass die Videos immer in englischer Sprache vorgeführt wurden und diese zumeist von den Teilnehmer/innen abgelehnt wird.

### ***Unterrichtsmittel objektorientierte Gruppe***

In der objektorientierten Gruppe sind das Whiteboard, Tafel und Flipchart, sowie die Trainerunterlagen besonders gut angekommen. Die Trainerunterlagen umfassen die Erklärungs- und Übungsblätter, die an die Teilnehmer ausgegeben werden und diese bei den verschiedenen Übungen unterstützen. Auch in dieser Gruppe sind die Videos am schlechtesten bei den Teilnehmer/innen angekommen und aus der Unterrichtsbeobachtung kann man auch hierbei wieder sagen, dass die englische Sprache ein wichtiger Faktor für dieses schlechte Abschneiden ist.

Als Fazit für die Unterrichtsmittel kann man mitnehmen, dass englische Videos bei den Teilnehmer/innen nicht gut ankommen und sich die Teilnehmer/innen mit dem klassischen Frontalunterricht, wo sehr häufig das Whiteboard zum Einsatz kommt, am wohlsten fühlen.

### ***Interesse und Zeitaufwand***

Bei der Frage, ob der Kurs das Interesse an der Programmierung für die Teilnehmer/innen erhöhte, hat sich folgendes ergeben:

Beide Gruppen, prozedurale und objektorientierte, haben durch den Kurs ein höheres Interesse an der Programmierung entwickelt. Alleine 47% der prozeduralen Gruppe hat ein viel höheres Interesse an der Programmierung entwickelt als vor dem Kurs. Nur für 35% hat sich das Interesse nicht oder nur wenig erhöht.

Bei der objektorientierten Gruppe zeigen 33% ein viel höheres Interesse und 43% mehr Interesse an der Programmierung. Nach dem Kurs zeigen nur 5% kein Interesse, sowie 19% nur wenig Interesse an der Programmierung.

In Sachen Zeitaufwand haben 64% der prozeduralen Gruppe bis zu drei Stunden die Woche zusätzlich zum Kurs Zeit für das Unterrichtsfach Programmieren investiert.

24% haben zwischen drei und sechs Stunden pro Woche investiert und 12% sogar noch mehr Zeit für das Unterrichtsfach Programmieren aufgewendet.

In der objektorientierten Gruppe haben 62% der Teilnehmer/innen bis zu drei Stunden die Woche in die Programmierung investiert. 14% haben zwischen drei und sechs Stunden investiert und ganze 24% haben mehr als sechs Stunden pro Woche für das Unterrichtsfach programmieren investiert.

Hierbei kann man gut erkennen, dass die objektorientierte Gruppe mehr Freizeit für die Programmierung investiert.

Laut Unterrichtsbeobachtung kann man aber hierbei auch sagen, dass dies auch sehr viel mit der Klassendynamik zu tun hat, da sich in der objektorientierten Gruppe sehr schnell nach Kursbeginn eine Lerngruppe gebildet hat, die regelmäßig noch nach dem Unterricht in den Kursräumen verblieb und den Programmierunterricht nochmals durchnahm und besprach. Dies hat sich auch daraus ergeben, dass die besseren Teilnehmer/innen so kollegial waren und die schlechteren Teilnehmer/innen ganz tatkräftig unterstützten.

#### **5.8.4 Erkenntnisse nach der Ausbildung**

Dieser Teil des Fragebogens wurde nach der kompletten Ausbildung, also nach dem Modul 8 und der Lehrabschlussprüfung ausgegeben. Ziel dieser Fragen ist festzustellen, inwieweit die Unterrichtsform – prozedurale Kenntnisse oder objektorientierte Kenntnisse zuerst – die Entwicklung zum/r Programmierer/in beeinflusst.

Auf die Frage, ob der Kurs über die Module 1 und 2 den Teilnehmer/innen bei dem LAP-Projekt hilft, antworteten die Teilnehmer/innen folgend:

In der prozeduralen Gruppe finden 36% der Teilnehmer/innen, dass ihnen der Kurs für die LAP-Phase zu gleichen Teilen nur wenig bis sehr wenig geholfen hat. Im Gegenzug dazu fand kein/e Teilnehmer/in der objektorientierten Gruppe, dass der Kurs gar nicht hilft, sondern 24% dass der Kurs nur wenig hilft. Außerdem wird dieser positivere Trend auch dadurch bei der objektorientierten Gruppe untermauert, indem 43% der Teilnehmer/innen den Kurs als perfekte Grundlage für die LAP-Phase empfinden und in der prozeduralen dies nur 29% so empfinden. Im Allgemeinen kann man hierbei schon sagen, dass der objektorientierte Kurs in Hinsicht auf die LAP-Phase positiver über einen längeren Zeitraum im Gedächtnis geblieben ist als der prozedurale Kurs.

Interessanterweise findet sich diese positive Einstellung in der nächsten Frage nicht mehr so deutlich bei der objektorientierten Gruppe. Hierbei geht es um die Selbsteinschätzung der Programmierkenntnisse nach Abschluss der Ausbildung. Hierbei war

die objektorientierte Gruppe zurückhaltender. In dieser finden sich nur 9% der Teilnehmer/innen, die sich als Expert/innen bezeichnen würden. In der prozeduralen Gruppe findet man einen doppelt so hohen Wert, mit 18% von Teilnehmer/innen, die sich als Expert/innen sehen. Sieht man sich die Zahlen für fortgeschrittene Kenntnisse an, so ergibt sich wieder ein homogeneres Bild, wo auch 43% der objektorientierten Gruppe sich als fortgeschritten sieht. In der prozeduralen Gruppe beträgt dieser Wert 35% und rechnet man nun die Expert/innen und Fortgeschrittenen Teilnehmer/innen zusammen, dann ergibt dies für die prozedurale Gruppe 53% und für die objektorientierte Gruppe 52%. Das heißt, ein bisschen mehr als die Hälfte jeder Gruppe sieht seine/ihre Programmierkenntnisse in einem sehr positiven Licht. Als Anfänger/in betiteln sich in der prozeduralen Gruppe nur 18% und in der objektorientierten Gruppe 19%.

Geht es konkret um die objektorientierten Kenntnisse, so sieht man wie auch schon bei der Umfrage direkt nach dem Kurs, dass die objektorientierte Gruppe weiterhin sicherer mit den objektorientierten Themen umgeht als die prozedurale Gruppe. 67% der objektorientierten Gruppe glauben die Grundlagen der objektorientierten Programmierung gut oder sehr gut verstanden zu haben, nur bei 57% der prozeduralen Gruppe ist dies der Fall. Dieser 10% Unterschied findet man bei den Teilnehmer/innen, die sich als sehr gut in dieser Materie bezeichnen würden. Eine äußerst positive Entwicklung zeigen die 12% bei der prozeduralen und nur 5% der objektorientierten Teilnehmer/innen, die ihr Wissen über die objektorientierten Grundlagen nur als sehr geringfügig einstufen.

Die eher pessimistische Einschätzungen der objektorientierten Gruppe in Bezug auf berufliche Chancen als Programmierer/innen, drücken sich in den 48% von Teilnehmer/innen aus, die ihre Chancen eher schlecht sehen auch als Programmierer/in arbeiten zu können. Diese Zahl ist bei der prozeduralen Gruppe um einiges besser und beträgt nur 24%. Diese Antworten können aber auch sehr stark von den Erfahrungen beim Praktikum und der aktuellen Arbeitsmarktlage beeinflusst worden sein, da zwischen dem Ausbildungszeitraum der prozeduralen Gruppe und der objektorientierten Gruppe doch ein Zeitraum von circa einem halben Jahr liegt.

## **5.9 Überprüfung der Hypothese**

Wie in der Einleitung schon beschrieben lautet die Hypothese, dass der prozedurale Lehransatz zwar am Anfang leichter zu verstehen ist, doch der objektorientierte Lehransatz schlussendlich die besseren Ergebnisse in Bezug auf Projektqualität und tieferes Verständnis für die objektorientierte Programmierung bringt. Folgend werden diese drei Hypothesen separat näher behandelt.

### **5.9.1.1 Leichter zu erlernen**

Die Hypothese, dass der objektorientierte Ansatz zuerst schwerer zu erlernen ist, kann anhand der Gegenüberstellung der Testergebnisse lt. Kapitel 5.7.6 aus dem Modul 1 nicht direkt bestätigt werden. Hierbei hat lt. Abbildung 16: Vergleich Testnoten Modul 1 PP vs. OOP die prozedurale Gruppe zwar mehr *sehr gut*, doch auch mehr *nicht genügend* als die objektorientierte Gruppe. Als Gesamtbild sieht die objektorientierte Gruppe homogener aus, indem von vielen positiven Noten ein Trend zu wenig negativen Noten ersichtlich ist. Die prozedurale Gruppe zeigt eher eine Notenschere, wobei das den Eindruck erwecken kann, dass man entweder gut programmieren oder nicht programmieren kann, die restlichen Noten sind nur spärlich vertreten.

Nimmt man Abstand von den Testnoten und betrachtet die Unterrichtsbeobachtungen des Kurses, so zeigt sich ein leicht verändertes Bild. Beide Gruppen hatten ihre Schwierigkeiten im Unterrichtsfach Programmieren zu meistern, aber die objektorientierte Gruppe hat im Allgemeinen eine bessere Zusammenarbeit. Dieser Zusammenhalt innerhalb der Gruppe hat, laut Beobachtungen der Trainer/innen, den objektorientierten Gruppen das Erlernen einer objektorientierten Programmiersprache erleichtert und die Gruppe insgesamt mehr motiviert. Dies spiegelt sich auch leicht in Abbildung 37: Zeitaufwand Freizeit, wo sich die objektorientierte Gruppe doch auch mehr in der Freizeit beschäftigt. Außerdem wird in Abbildung 19: Programmierkenntnisse vor Kurs ersichtlich, dass die objektorientierte Gruppe einen leichten Vorsprung in Sachen Programmiergrundlagen besaß. Dies könnte ein Indiz sein, dass die objektorientierte Gruppe eigentlich im Allgemeinen eine leicht geringere Einstiegshürde in den Kurs hatte.

Laut der erhobenen Daten muss man aber sagen, dass der objektorientierte Lehransatz effektiver in der Wissensvermittlung für objektorientierte Programmierproblematiken ist und anhand der Unterrichtsbeobachtung auch die Teilnehmer/innen während des Kurszeitraums mehr motiviert, obwohl diese sich selbst schlechter einschätzen.

### **5.9.1.2 Bessere Projektqualität**

Die Hypothese, dass ein objektorientierter Lehransatz schlussendlich eine bessere Projektqualität am Ende des Kurses, sowie der gesamten Ausbildung, bringen kann, wurde mit Hilfe der Gruppenarbeiten, Referate und den AKT-, sowie LAP-Projekten anhand der Unterrichtsbeobachtung untersucht.

Während des Kurszeitraums finden nur begrenzt Gruppenarbeiten und Referate statt, da für diese nur wenige Freiflächen eingeplant sind. Trotzdem ist die Qualität und der Eindruck von den Präsentationen bei den objektorientierten Gruppen im Allgemeinen positiver zu beurteilen. Dies kann wiederum an der funktionierenderen Zusammenarbeit innerhalb der objektorientierten Gruppen liegen. Trotzdem waren die Unterschiede der Ausarbeitungen nicht außergewöhnlich weit voneinander entfernt, sodass die

prozeduralen Gruppen auch solide Gruppenarbeiten und Referate leisteten. Meist hat es bei den prozeduralen Gruppen einfach nur mehr Führung durch die Trainer/innen benötigt, damit die Zusammenarbeit innerhalb der Gruppe funktionierte.

Der Unterscheid zwischen den Gruppen fällt während der AKT- und LAP-Projektphasen gravierender aus. Die objektorientierte Gruppe investiert mehr Zeit in die Planung von Klassenstrukturen und den Aufbau von sauberen Schnittstellen zwischen verschiedenen Ebenen. Somit tut sich die objektorientierte Gruppe auch leichter mit der Kommunikation zwischen einzelnen Teilnehmer/innen und hat im Allgemeinen weniger Probleme, fremde Schnittstellen in ihr eigenes Programm zu integrieren. Die prozeduralen Gruppen haben hierbei einen pragmatischeren Weg verfolgt und zumeist gleich mit der Programmierung begonnen, sofern die Trainer/innen nicht explizit eine Planung verlangten. Die Endresultate sind aber bei beiden Gruppen zufriedenstellend, denn die objektorientierte Gruppe hat meist mehr Probleme die geplanten Szenarien auch tatsächlich im Detail umzusetzen. Für größere Projekte ist der objektorientierte Ansatz bestimmt der zu bevorzugende, aber für die erforderliche Projektgröße eines AKT- oder LAP-Projekts, war der prozedurale Ansatz gleichzusetzen. Mit Hilfe von externen APIs sehen aber die Projekte der objektorientierten Gruppen ansehnlicher aus, obwohl diese Gruppen meist weniger selbst implementierten.

Daraus ergibt sich, dass die objektorientierten Gruppen zumeist eine bessere Qualität in der Planung und Dokumentation von Projekten ablieferten, aber die prozeduralen Gruppen zumeist bessere Implementierungen einzelner Abschnitte haben.

Zusammengefasst können alle Teilnehmer/innen aus beiden Gruppen die für die Lehrabschlussprüfung erwünschte Qualität erreichen und somit das AKT-Projekt und das LAP-Projekt zufriedenstellend abschließen.

Welcher Lehransatz effektiver für die Wissensvermittlung in der Praxis, und somit für das Berufsleben, ist kann hiermit nicht eindeutig gesagt werden. Dies hängt von der Spezialisierung im Berufsleben ab. Beide Lehransätze haben hierbei ihre selbstgemachten Schwächen – die objektorientierte Gruppe in der Implementierung und die prozedurale Gruppe in der objektorientierten Planung.

### **5.9.1.3 Objektorientiertes Verständnis**

Das objektorientierte Verständnis ist in der objektorientierten Gruppe, anhand der Testnoten aus dem zweiten Modul, auf alle Fälle gefestigter zu sehen als in der prozeduralen Gruppe. Im Vergleich der Testnoten aus Modul 2 in Abbildung 17: Vergleich Testnoten Modul 2 PP vs. OOP kann man sehr gut sehen, dass die prozedurale Gruppe im Allgemeinen mit schlechteren Testnoten als die objektorientierte Gruppe abschneidet. Gerade in der Notenkategorie *sehr gut* ist die prozedurale Gruppe so gut wie nicht vorhanden, wobei gerade für eine sehr gute Note die Testfragen im

Modul 2 auf objektorientierte Grundlagen abzielen. Offensichtlich hat die objektorientierte Gruppe hierbei den Vorteil genossen, die objektorientierte Denkweise länger üben zu können um diese zu verinnerlichen.

Die objektorientierten Grundlagen werden auch besonders für das AKT-Projekt und die LAP-Projekte der jeweiligen Gruppen benötigt. Zusätzlich wird gerade das theoretische Verständnis für die dazugehörigen Prüfungssituationen, wie z.B. das Fachgespräch für die Lehrabschlussprüfung, benötigt. Die Unterrichtsbeobachtungen dieser Module zeigen auch, dass sich im Allgemeinen die objektorientierte Gruppe leichter mit den objektorientierten Ansätzen und der Planung von Projekten tut als die prozedurale Gruppe. Andererseits tut sich die prozedurale Gruppe mit der Umsetzung von einfachen Algorithmen im Allgemeinen ein bisschen leichter als die objektorientierte Gruppe. Die Kommunikation ist in der objektorientierten Gruppe durch das annähernd gleiche Verständnis von objektorientierten Ansätzen besser zu bewerten.

Beide Gruppen haben aber alle die Lehrabschlussprüfung in Bezug auf das Fach Programmieren positiv absolviert. Wo die eine Gruppe in objektorientierten Planen und Umsetzen der anderen überlegen ist, so ist die andere Gruppe in der praktischen Umsetzung von Algorithmen überlegen.

Besonders auffällig ist bei beiden Lehrmethoden, dass sich die Teilnehmer/innen, die schon vor dem Kurs prozedurale Programmiererfahrung hatten, am schwersten mit dem Umstieg auf die objektorientierte Denkweise taten. Dies fiel auch gerade im Modul 3 auf, da jedes Steuerelement eine Klasse ist und der Umgang mit diesen leichter fällt, sofern man objektorientierte Programmiertechniken anwendet.

Außerdem hat man in der prozeduralen Gruppe bemerkt, dass die Motivation am Unterricht aktiv teilzunehmen und das Interesse am Programmierunterricht im Allgemeinen in Modul 2 abnahm, als die ersten Teilnehmer/innen Probleme mit der objektorientierten Denkweise bekamen. Zumeist können die Teilnehmer/innen nicht zwischen den verschiedenen Rollen, einerseits des Klassendesigners und andererseits der des/der Programmierers/in, die die Klasse verwendet, umschalten. Daraus folgend ist festzustellen, dass der objektorientierte Ansatz das Interesse an der Programmierung mehr fördert und im prozeduralem Ansatz das Risiko einer Demotivation ab dem Modul 2 herrscht.

## 6 Schlussfolgerungen

Da die erfassten Daten für diese Arbeit anhand einer tatsächlichen Umschulung an einem Bildungsinstitut erfasst wurden, ist diese Arbeit besonders von den Programmiertrainer/innen kritisch betrachtet worden, um weitere Schritte für die nächsten geplanten Kurse zu setzen.

Die Auffälligkeit, dass sich die objektorientierte Gruppe mit dem Einstieg in die Programmierung nicht bemerkenswert schwerer tut als die prozedurale Gruppe, brachte ein paar Programmiertrainer/innen zu einem Umdenken, ihren Unterrichtsstil in eine objektorientierte Richtung zu lenken. Gerade die Qualität der Planung wurde von den Programmiertrainer/innen gelobt. Kritisch beobachtet wurde aber, dass andere Grundlagen – besonders diese der prozeduralen Programmierung – leider von den objektorientierten Gruppen nicht mehr so qualitativ hochwertig erfüllt werden. Im Speziellen werden optionale Themen wie die mehrdimensionalen Arrays außen vor gelassen, was nicht bei allen Trainer/innen Zuspruch findet. Da die objektorientierte Einführung doch ein bisschen mehr Zeit beansprucht als die prozedurale Einführung, wurde der Lehrplan kritisch betrachtet und einige Spezialthemen anhand der Unterrichtsbeobachtung schlussendlich entfernt, gekürzt oder in andere Module verschoben. Besonders die Strukturen und Arrays mussten eine Kürzung hinnehmen. In weiteren Kursen werden Strukturen nur mehr rudimentär umgesetzt und dafür der Fokus auf Klassenübungen gerichtet. Arrays werden nur mehr in ihrer einfachsten Form durchgenommen und Übungen zu Arrays auf ein Minimalprogramm gekürzt. Zukünftige Klassen werden sich mehr mit Listen beschäftigen, da sich diese als verständlicher und umgänglicher erwiesen haben.

Nach den stofflichen Veränderungen im Lehrplan werden im nächsten Schritt auch die Übungsbeispiele, die anhand der Unterrichtsbeobachtung nicht gut ankommen, überarbeitet oder in der jetzigen Form entfernt. Da die Partnerarbeit besonders gut bei den Teilnehmer/innen ankommt, wird diese noch mehr in die Übungen integriert. Referate und Gruppenarbeiten bleiben aus zeittechnischen Gründen weiter in den späteren Modulen, werden aber besonders in der AKT-Phase gefördert, da diese auch ganz gut ankommen und für die Ausbildung essenziell sind. Englische Videos wurden nach Absprache der Trainer/innen aus dem Unterricht entfernt. Zwar empfinden alle Trainer/innen die englische Sprache für Informatiker/innen als besonders wichtig, aber die Hürde der Fremdsprache ist für viele der Teilnehmer/innen doch zu hoch und man baut somit künstlichen Druck auf, den man aber eigentlich nehmen möchte. Damit Englisch im Programmierunterricht trotzdem seinen Platz findet, werden von den Trainer/innen einfache englische Tutorials zur Webprogrammierung für die Module 7 und 8 vorbereitet. Der Ansatz *„Zusammensetzung von geeigneten Inhalten ist von zentraler Bedeutung für die Vermittlung von Begriffen und Prinzipien der Informatik.“* (Schubert & Schwill, 2011) wird für die weitere Kursplanung beachtet und diktiert eine wiederholende Anpassung der Kursunterlagen für die Teilnehmer/innen der Ausbildung.

Als kritische Bemerkung zu dieser Gesamtarbeit ist anzumerken, dass diese Erhebung durch die lange Laufzeit über mehrere Monate, da die Kurse zeitlich verteilt geführt werden, schwierig zu reproduzieren ist. Als Gegenmaßnahme wurde bei der Evaluierung der Fokus auf den Kernteil dieser Arbeit, nämlich den Kurs über die Module 1 und 2, gerichtet. Dies macht den Untersuchungszeitraum mit 20 Wochen überschaubar und reproduzierbar. Als zweiten kritischen Aspekt kann hinzugefügt werden, dass der Verfasser dieser Arbeit nicht ganz objektiv gegenüber den Teilnehmer/innen sein kann, da auch in gewisser Weise ein persönlicher Kontakt zu den einzelnen Gruppen im Laufe der Ausbildung aufgebaut wird. Um dem entgegenzuwirken werden besonders die Mitarbeits- und Testnoten gewichtet betrachtet. Außerdem vermutet der Verfasser dieser Arbeit, dass die positiveren Ergebnisse der objektorientierten Gruppen teilweise von der Gruppendynamik abhängig waren.

Als Resultat dieser Erhebung und Evaluierung wird im Bildungsinstitut die Lehrmethode auf den objektorientierten Ansatz gewechselt. Die Programmiertrainer/innen beobachten diese Veränderung mit wachsamem Augen, ob dieser Ansatz auch bei zukünftigen Gruppen der effektivere Weg ein/e Informatiker/in zu werden ist.



## 7 Literaturverzeichnis

- Ausbildung IT-InformatikerIn. (2015). BBRZ Reha GmbH. Retrieved from [http://www.bbrz.at/fileadmin/user\\_upload/Produktblaetter/Reha-Ausbildung/Wien/IT\\_InformatikerIn-Wien.pdf](http://www.bbrz.at/fileadmin/user_upload/Produktblaetter/Reha-Ausbildung/Wien/IT_InformatikerIn-Wien.pdf)
- bmwfw. (2006). Informationstechnologie-Ausbildungsordnung. Retrieved from <http://www.bmwfw.gv.at/Berufsausbildung/LehrberufelnOesterreich/ListeDerLehrberufe/Seiten/Informatik.aspx>
- Brilliant, S. S., & Wiseman, T. R. (1996). The first programming paradigm and language dilemma. In *Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education - SIGCSE '96* (Vol. 28, pp. 338–342). New York, New York, USA: ACM Press. Retrieved from <http://dl.acm.org/citation.cfm?id=236452.236572>
- Burkard, C., & Eikenbusch, G. (2000). *Praxisbuch: Praxishandbuch Evaluation in der Schule*. Cornelson Lehrbuch.
- Burton, P. J., & Bruhn, R. E. (2003). Teaching programming in the OOP era. *ACM SIGCSE Bulletin*, 35(2), 111–114. Retrieved from <http://dl.acm.org/citation.cfm?id=782941.782993>
- Ehlert, A., & Schulte, C. (2009). Empirical comparison of objects-first and objects-later. In *Proceedings of the fifth international workshop on Computing education research workshop - ICER '09* (p. 15). New York, New York, USA: ACM Press. doi:10.1145/1584322.1584326
- Felder, R. M., & Silverman, L. K. (1988). LEARNING AND TEACHING STYLES. *Engineering Education*, 78(7), 674–681.
- Grün, G., Tritscher-Archan, S., & Silvia, W. (2009). Leitfaden zur Beschreibung von Lernergebnissen. Retrieved from [http://ibw4.m-services.at/zoom/pdf/wp2/Leitfaden\\_DE\\_final\\_2.pdf](http://ibw4.m-services.at/zoom/pdf/wp2/Leitfaden_DE_final_2.pdf)
- Hu, C. (2004). Rethinking of Teaching Objects-First. *Education and Information Technologies*, 9(3), 209–218. doi:10.1023/B:EAIT.0000042040.90232.88
- Hubwieser, P. (2007). *Didaktik der Informatik: Grundlagen, Konzepte, Beispiele*. Springer Berlin Heidelberg.
- Louis, D., Kansy, T., & Strasser, S. (2013). *Microsoft Visual C# 2012 - Das Entwicklerbuch*. O'Reilly Verlag GmbH & Co. KG.
- Meyers enzyklopädisches Lexikon. (1974). In *Meyers Enzyklopädie*.
- Reich, K. (2007). Methodenpool. Retrieved from <http://www.uni-koeln.de/hf/konstrukt/didaktik/zitieren.html>

- Reich, K. (2012). *Konstruktivistische Didaktik*. Beltz Verlagsgruppe. Retrieved from [http://www.beltz.de/fachmedien/paedagogik/buecher/produkt\\_produktdetails/4634-konstruktivistische\\_didaktik.html](http://www.beltz.de/fachmedien/paedagogik/buecher/produkt_produktdetails/4634-konstruktivistische_didaktik.html)
- Schubert, S., & Schwill, A. (2011). *Didaktik der Informatik*. Spektrum Akademischer Verlag: Heideberg.
- Schwaber, K., & Beedle, M. (2008). *Agile Software Development with Scrum* (Internatio.). PEARSON STUDIUM.
- Schwaber, K., & Sutherland, J. (2013). Scrum Guide. Retrieved from <http://www.scrumguides.org/scrum-guide.html>
- Suh, B. (2007). Development and application of an integrated course for procedural and object-oriented programming languages in EE. In *TENCON 2007 - 2007 IEEE Region 10 Conference* (pp. 1–4). IEEE.  
doi:10.1109/TENCON.2007.4428850
- Vilner, T., Zur, E., & Gal-Ezer, J. (2007). Fundamental concepts of CS1. *ACM SIGCSE Bulletin*, 39(3), 171. doi:10.1145/1269900.1268835
- Wulf, C. (1982). *Evaluation. Beschreibung und Bewertung von Unterricht, Curricula und Schulversuchen*. Piper Verlag GmbH. Retrieved from [http://www.amazon.de/Evaluation-Beschreibung-Bewertung-Unterricht-Schulversuchen/dp/3492019854/ref=sr\\_1\\_1?s=books&ie=UTF8&qid=1437554862&sr=1-1&keywords=Evaluation.+Beschreibung+und+Bewertung+von+Unterricht%2C+Curricula+und+Schulversuchen](http://www.amazon.de/Evaluation-Beschreibung-Bewertung-Unterricht-Schulversuchen/dp/3492019854/ref=sr_1_1?s=books&ie=UTF8&qid=1437554862&sr=1-1&keywords=Evaluation.+Beschreibung+und+Bewertung+von+Unterricht%2C+Curricula+und+Schulversuchen)

## 8 Abbildungsverzeichnis

Abbildung 1: Einbettung der Didaktik laut (Schubert und Schwill, 2011) .....	35
Abbildung 2: Einbettung der Didaktik am Bildungsinstitut .....	35
Abbildung 3: Unterrichtszyklus Einheitenblock .....	44
Abbildung 4: Lernergebnisse Kurs Allgemein .....	50
Abbildung 5: Unterrichtsnotiz.....	92
Abbildung 6: Verteilung Mitarbeitsnoten prozedurale Gruppe Modul 1 .....	107
Abbildung 7: Verteilung Mitarbeitsnoten prozedurale Gruppe Modul 2 .....	108
Abbildung 8: Verteilung Mitarbeitsnoten objektorientierte Gruppe Modul 1 .....	108
Abbildung 9: Verteilung Mitarbeitsnoten objektorientierte Gruppe Modul 2 .....	109
Abbildung 10: Vergleich Mitarbeit Modul 1 PP vs. OOP.....	110
Abbildung 11: Vergleich Mitarbeit Modul 2 PP vs. OOP.....	110
Abbildung 12: Verteilung Testnoten prozedurale Gruppe Modul 1 .....	112
Abbildung 13: Verteilung Testnoten prozedurale Gruppe Modul 2.....	112
Abbildung 14: Verteilung Testnoten objektorientierte Gruppe Modul 1 .....	113
Abbildung 15: Verteilung Testnoten objektorientierte Gruppe Modul 2 .....	114
Abbildung 16: Vergleich Testnoten Modul 1 PP vs. OOP .....	115
Abbildung 17: Vergleich Testnoten Modul 2 PP vs. OOP .....	115
Abbildung 18: Computerkenntnisse .....	117
Abbildung 19: Programmierkenntnisse vor Kurs .....	118
Abbildung 20: Programmierkenntnisse nach Kurs .....	119
Abbildung 21: Objektorientierte Grundlagen .....	119
Abbildung 22: Hilfestellung Trainer/innen .....	120
Abbildung 23: Übungsbeispiele .....	121
Abbildung 24: Menge Frontalunterricht.....	121
Abbildung 25: Hilfe Frontalunterricht.....	122
Abbildung 26: Menge Einzelarbeit .....	123
Abbildung 27: Hilfe Einzelarbeit.....	123
Abbildung 28: Menge Gruppenarbeit.....	124
Abbildung 29: Hilfe Gruppenarbeit.....	125
Abbildung 30: Menge Partnerarbeit .....	125
Abbildung 31: Hilfe Partnerarbeit.....	126
Abbildung 32: Menge Referat .....	127
Abbildung 33: Hilfe Referat.....	127
Abbildung 34: Medien prozedurale Gruppe .....	128
Abbildung 35: Medien objektorientierte Gruppe .....	129
Abbildung 36: Programmierinteresse.....	129
Abbildung 37: Zeitaufwand Freizeit .....	130
Abbildung 38: LAP-Projektunterstützung .....	131
Abbildung 39: Programmierkenntnisse nach Ausbildung .....	132
Abbildung 40: OO-Grundlagen nach Ausbildung .....	133
Abbildung 41: Berufschancen nach Ausbildung.....	134

## 9 Tabellenverzeichnis

Tabelle 1: Berufsbild laut (bmwfw, 2006).....	17
Tabelle 2: Abkürzungen Unterrichtsfächer.....	22
Tabelle 3: Modulaufteilung .....	24
Tabelle 4 Grundlagenmodul für Informatiker/innen .....	26
Tabelle 5: Lehrplan IT-Informatiker/innen .....	26
Tabelle 6: Zuordnung Bildungsinstitut und Berufsbild .....	30
Tabelle 7: Übersicht Lehreinheiten .....	55
Tabelle 8: Lehreinheiten prozedurale Gruppe Module 1 und 2 .....	63
Tabelle 9: Lehreinheiten objektorientierte Gruppe Module 1 und 2.....	72
Tabelle 10: Lehreinheiten UI Modul 3.....	77
Tabelle 11: Lehreinheiten AKT Module 4 und 5.....	81
Tabelle 12: Lehreinheiten LAP Module 7 und 8 .....	84
Tabelle 13: Computerkenntnisse .....	117
Tabelle 14: Programmierkenntnisse vor Kurs .....	117
Tabelle 15: Programmierkenntnisse nach Kurs .....	118
Tabelle 16: Objektorientierte Grundlagen .....	119
Tabelle 17: Hilfestellung Trainer/innen .....	120
Tabelle 18: Übungsbeispiele .....	120
Tabelle 19: Menge Frontalunterricht.....	121
Tabelle 20: Hilfe Frontalunterricht.....	122
Tabelle 21: Menge Einzelarbeit .....	122
Tabelle 22: Hilfe Einzelarbeit.....	123
Tabelle 23: Menge Gruppenarbeit .....	124
Tabelle 24: Hilfe Gruppenarbeit.....	124
Tabelle 25: Menge Partnerarbeit .....	125
Tabelle 26: Hilfe Partnerarbeit.....	126
Tabelle 27: Menge Referat.....	126
Tabelle 28: Hilfe Referat.....	127
Tabelle 29: Medien prozedurale Gruppe .....	128
Tabelle 30: Medien objektorientierte Gruppe .....	128
Tabelle 31: Programmierinteresse.....	129
Tabelle 32: Zeitaufwand Freizeit.....	130
Tabelle 33: LAP-Projektunterstützung .....	131
Tabelle 34: Programmierkenntnisse nach Ausbildung .....	131
Tabelle 35: OO-Grundlagen nach Ausbildung .....	132
Tabelle 36: Berufschancen nach Ausbildung.....	133