

Bartelsen, Jan et al.

Working Paper

Erfahrungsbericht eines Blended-learning-Kurses zur Programmiergrundausbildung: Objektorientierte Entwicklung von Web-Anwendungen mit Smalltalkt/Seaside

Arbeitspapiere der Nordakademie, No. 2012-08

Provided in Cooperation with:

Nordakademie - Hochschule der Wirtschaft, Elmshorn

Suggested Citation: Bartelsen, Jan et al. (2012) : Erfahrungsbericht eines Blended-learning-Kurses zur Programmiergrundausbildung: Objektorientierte Entwicklung von Web-Anwendungen mit Smalltalkt/Seaside, Arbeitspapiere der Nordakademie, No. 2012-08, Nordakademie - Hochschule der Wirtschaft, Elmshorn

This Version is available at:

<https://hdl.handle.net/10419/67097>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT

ARBEITSPAPIERE DER NORDAKADEMIE

ISSN 1860-0360

Nr. 2012-08

Erfahrungsbericht eines Blended-learning-Kurses zur Programmiergrundausbildung

Objektorientierte Entwicklung von Web-Anwendungen mit Smalltalk/Seaside

**Jan Bartelsen, Johannes Brauer, Sandra Gebert, André Kowalk,
Andreas Krey, Christina Krey, Heiko Rehder, Sophie Schumann**

November 2012

Eine elektronische Version dieses Arbeitspapiers ist verfügbar unter
<http://www.nordakademie.de/arbeitspapier.html>



NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT

Köllner Chaussee 11
25337 Elmshorn
<http://www.nordakademie.de>

Erfahrungsbericht eines Blended-learning-Kurses zur Programmiergrundausbildung

Objektorientierte Entwicklung von Web-Anwendungen mit
Smalltalk/Seaside

Jan Bartelsen, Johannes Brauer, Sandra Gebert, André Kowalk,
Andreas Krey, Christina Krey, Heiko Rehder, Sophie Schumann

12. November 2012

Inhaltsverzeichnis

1	Ausgangssituation	2
2	Der Westpol-Kurs	4
2.1	Das Konzept	4
2.2	Die Entwicklungsumgebung	5
3	Programmierung 1 im Wintersemester 2011/2012	5
4	Die gewonnenen Erfahrungen	6
4.1	Blended-learning im Allgemeinen	6
4.2	Inhaltliche Themen	8
4.3	VisualWorks/Seaside	9
4.4	Kritik an einzelnen Themenblöcken	10
4.5	Prüfungsvorbereitung	11
5	Schlussfolgerungen	12
6	Ausblick	13
7	Danksagung	14

Abstract

Im Wintersemester 2011/2012 wurde erstmals das Modul *Programmierung 1* im Bachelor-Studiengang *Wirtschaftsinformatik* der NORDAKADEMIE nach einem im Rahmen des Projekts *Westpol* entwickelten Konzeptes unterrichtet. Die Wurzeln dieses Konzeptes und die Rahmenbedingungen der Kursdurchführung werden in diesem Arbeitspapier erläutert. Ebenfalls wird über die dabei gemachten Erfahrungen berichtet, die auf wöchentlich festgehaltenen Feedbacks der Studierenden basieren. Dazu wurde von den Autoren die Vielzahl der Feedbacks gesichtet und versucht, nach verschiedenen Kriterien zu gruppieren, um damit eine ausreichende Grundlage für eine spätere Überarbeitung des Kurses zu erhalten.

1 Ausgangssituation

Seit dem Wintersemester 1999/2000 steht die Programmiergrundausbildung im Studiengang Wirtschaftsinformatik der NORDAKADEMIE unter dem Motto „Objektorientierung von Anfang an“¹ und wird unter Verwendung der Programmiersprache Smalltalk in einer *Programmierung 1* genannten Lehrveranstaltung im ersten und zweiten Semester durchgeführt. Im dritten und vierten Semester schloss sich die Lehrveranstaltung *Programmierung 2* an, in der die Übertragung erlernter Grundbegriffe der Objektorientierung auf die Programmiersprache Java im Vordergrund stand.

Die Hochschule folgte mit diesem Konzept damals dem Trend, die in den neunziger Jahren noch überwiegend „prozedural“ geprägte Ausbildung stärker auf die Objektorientierung hin auszurichten. Vielerorts setzte sich die Ansicht durch, dass sich die objektorientierte Denkweise in den Köpfen der Studierenden am Besten verankern lässt, wenn man objektorientierte Programmierung von Beginn an unterrichtet. Dieser Ansatz wurde und wird unter dem Stichwort „Objects first“ auf vielen Konferenzen, die sich mit der Didaktik der Programmierausbildung beschäftigen, diskutiert.

DIE ERFAHRUNGEN MIT DEM OBJECTS-FIRST-ANSATZ der ersten Jahre zeigten aber, dass einige Studierende nach den ersten beiden Semestern Probleme mit der Objektorientierung erkennen ließen. Dazu gehörten z. B.

- Prozedurale Denkweise „bricht immer wieder durch“.
- Schwierigkeiten beim Klassenentwurf
- „Missbrauch“ von Vererbung
- Fallunterscheidungen anstelle der Nutzung von Polymorphie
- Verwechslung von Exemplar- und Unterklassenbeziehung

Bis zu diesem Zeitpunkt war die Lehrveranstaltung *Programmierung 1* klassisch aufgeteilt in Vorlesung und Übungen, in denen von Woche zu Woche wechselnde Aufgaben zu bearbeiten waren, die das Verständnis der in der Vorlesung vermittelten Konzepte der Objektorientierung vertiefen sollten. Hierbei stellte sich heraus, dass ein tieferes Verständnis durch die Bearbeitung von kleinen Aufgaben nur schwer zu erzielen war. Erst wenn eine „Population“ von Objekten verschiedener Klassen für die Erzielung eines Anwendungszwecks untereinander Nachrichten austauschen muss, werden die Vorteile, aber auch die Schwierigkeiten mit der Objektorientierung deutlich.

Um die geschilderten Probleme anzugehen, wurde die Lehrveranstaltung *Programmierung 1* beginnend mit dem Wintersemester 2006/2007 überarbeitet:

- Die klassische Vorlesung-Übung-„Bauform“ wurde zugunsten eines Blended-learning-Konzepts² aufgegeben.
- Die während des Semesters zu bearbeitenden Einzelaufgaben wurden durch eine einzige, über das gesamte Semester zu bear-

¹ BRAUER, Johannes: *Grundkurs Smalltalk - Objektorientierung von Anfang an*. 3. Auflage. Vieweg+Teubner Verlag, 2009

Zum Beispiel:

- SIGCSE Technical Symposium on Computer Science Education
 - ITiCSE Innovation and Technology in Computer Science Education
- Einen guten Überblick vermittelt auch

BENNEDSEN, Jens ; CASPERSEN, Michael E. ; KLLING, Michael: *Reflections on the Teaching of Programming: Methods and Implementations*. 1. Springer Publishing Company, Incorporated, 2008. – ISBN 3540779337. 9783540779339

² unter Einsatz der Lernplattform moodle (<http://www.moodle.org>); der Einsatz von moodle war zu Beginn eine „Ad-hoc-Aktivität“, die im Wesentlichen darin bestand, dort die Programmieraufgabe in Teilschritten zu beschreiben und Verknüpfungen auf Teile des existierenden Vorlesungsskripts zu hinterlegen.

beitende, größere Programmieraufgabe ersetzt.

Die Umstellung auf einen Blended-learning-Kurs hatte in erster Linie zum Ziel, dem Dozenten die Rolle eines Betreuers zu ermöglichen, in der er mehr Zeit für Hilfestellungen bei individuellen Schwierigkeiten der Studierenden mit der Bewältigung der gestellten Aufgabe hat. Das ist angesichts der heterogenen Vorkenntnisse der Studienanfänger in der Programmierung von Bedeutung. Zum anderen wurde damit den Studierenden die Möglichkeit gegeben, den Kurs weitgehend selbständig zu bearbeiten.

DEN EINSATZ DER LERNPLATTFORM ZU SYSTEMATISIEREN, war Gegenstand der Master-Thesis von Jan Bartelsen³. Im Rahmen dieser Arbeit sowie im Projekt NORDPOL⁴ ist ein von den Studierenden gut akzeptierter⁵ Blended-learning-Kurs *Programmierung 1* entstanden.

DIE WAHL DER PROGRAMMIERSPRACHE hat wahrscheinlich auch einen Einfluss darauf, wie schnell insbesondere Programmieranfänger aber auch „Umsteiger“ mit den Konzepten der objektorientierten Programmierung vertraut gemacht werden können. Die „herkömmlichen“ objektorientierten Programmiersprachen⁶ erlauben tatsächlich nur einen „Classes-first“-Ansatz. In ihnen kann man eigentlich nicht mit Objekten beginnen, weil man zuvor Klassen haben muss, von denen Objekte als ihre Exemplare angelegt werden müssen. Bevor die Studierenden mit Objekten arbeiten können, müssen sie das abstraktere Konzept der *Klasse* verstanden haben.⁷ Dafür, dass es Klassen geben muss, gibt es gerade zu Beginn des Unterrichts keine wirklich überzeugende Begründung.

Um diesem Problem aus dem Wege zu gehen, könnte man natürlich auch eine objektorientierte Programmiersprache verwenden, die schon das Grundkonzept des Objekts konstruktiv anbietet und ggf. ganz ohne Klassen auskommt.⁸ Derartige Sprachen gelten aber unter dem Blickwinkel einer praxisorientierten Ausbildung als zu „exotisch“, zumal man in diesen Sprachen den Klassenbegriff wiederum nicht einführen kann.

In Smalltalk wird das Problem ein wenig dadurch gemildert, dass es für einige Basisklassen LiteralDarstellungen für ihre Exemplare gibt. Dazu gehören die Zahlenklassen⁹ sowie die Klassen String und Array. Dadurch ist es möglich, erste Grundbegriffe der objektorientierten Programmierung¹⁰ ohne Rückgriff auf den Klassenbegriff zu erläutern. Sehr weit kommt man damit aber auch nicht. Weder lässt sich erklären, weshalb manche Nachrichten von manchen Objekten verstanden werden, andere hingegen nicht, noch ist es möglich, eigene Objekte mit einer selbst definierten Struktur zu bauen.

Die Tatsache, dass heute viele Studienanfänger schon umfangreiche Programmierkenntnisse mitbringen, lässt jeden „first“-Ansatz fraglich erscheinen, wie Marc Guzdial in seinem Blogbeitrag „We are too late for ‘First’ in CS1“ (in <http://cacm.acm.org/blogs/blog-cacm/102624>) bemerkt.

³ BARTELTSEN, Jan-Ivar: *Entwurf einer kooperativen Lehrveranstaltung in moodle für die Grundausbildung in der Programmierung*. Master-Thesis an der FernUniversität Hagen, 2008

⁴ BARTELTSEN, Jan ; BRAUER, Johannes: *Kooperatives Lernen mit moodle in der Programmierausbildung / NORDAKADEMIE Hochschule der Wirtschaft*. 2008 (2008-6). – Arbeitspapier

⁵ BARTELTSEN, Jan ; BRAUER, Johannes ; REHDER, Heiko: *Abschlussbericht für das Projekt NORDPOL / NORDAKADEMIE Hochschule der Wirtschaft*. 2010 (2010-02). – Arbeitspapier

⁶ wie z. B. Java, C++ oder Smalltalk

⁷ BRAUER, Johannes ; KRASEMANN, Hartmut ; KRASEMANN, Christoph: *Objekte statt Klassen zuerst*. In: *Forschung für die Wirtschaft*. NORDAKADEMIE Hochschule der Wirtschaft, 2011

⁸ wie z. B. Self (<http://selflanguage.org/>) oder JavaScript (<https://developer.mozilla.org/en/JavaScript>)

⁹ Alle Zahlen in Smalltalk sind Exemplare von Klassen.

¹⁰ z. B. das Senden von Nachrichten an Objekte

2 Der Westpol-Kurs

Parallel zur Lehrveranstaltung *Programmierung 1* fand bis zum Studienbeginn Wintersemester 2011/2012 im ersten und zweiten Semester des Studiengangs Wirtschaftsinformatik die Lehrveranstaltung *Internet-Grundlagen* statt, in der die Studierenden grundlegende Inhalte zu HTML, XML, CSS und JavaScript sowie zur Gestaltung von Web-Seiten erlernten und als Prüfungsleistung (Hausarbeit) einen einfachen Web-Auftritt gestalteten.

Auf die Probleme des Nebeneinanders dieser beiden Lehrveranstaltungen soll an dieser Stelle nicht näher eingegangen werden. Es sollte jedoch erwähnt werden, dass die Lehrveranstaltung *Internet-Grundlagen* ab etwa der Mitte des ersten Jahrzehnts des 21. Jahrhunderts darunter litt, dass der Umfang der Vorkenntnisse der Studienanfänger sehr heterogen ist. So erschienen auf der einen Seite Studierende, für die der Themenkomplex HTML vollkommen neu ist und die noch nie eine Webseite erstellt haben. Auf der anderen Seite gab es Studierende, die bereits seit mehreren Jahren, teils sogar professionell, mit den Themen der Lehrveranstaltung *Internet-Grundlagen* arbeiteten. Für den Dozenten war es daher oft schwierig, einen Unterricht zu gestalten, der niemanden über-, aber auch niemanden unterforderte¹¹. Im Zuge der Neugestaltung des Bachelor-Studiengangs Wirtschaftsinformatik¹² kam der Gedanke auf, *Internet Grundlagen* und *Programmierung 1* zu einer neuen Lehrveranstaltung unter dem Motto „Objektorientierte Entwicklung von Web-Anwendungen“ zu verschmelzen.

Um dieses Vorhaben zu verwirklichen, wurde das Projekt WESTPOL¹³ ins Leben gerufen.

2.1 Das Konzept

Wichtige Grundgedanken des neuen Konzepts sind:

- integrative Behandlung von Lerninhalten „benachbarter“ Lehrveranstaltungen: Die Lehrveranstaltung *Internet-Grundlagen* wird als eigenständiges Modul obsolet. Die bisher dort vermittelten Inhalte werden – sozusagen nebenbei – im Zuge der Entwicklung einer Web-Anwendung mit gelehrt und gelernt.
- Die Lehrveranstaltung *Programmierung 1* wird attraktiver:
 - Bisher wurden in *Programmierung 1* nur Programme ohne jede graphische Benutzungsoberfläche geschrieben. Das wird von den Studierenden teilweise als unbefriedigend empfunden. Nach dem neuen Konzept entwickeln die Studierenden von Beginn an eine Anwendung mit einer Web-Oberfläche.
 - Der zunehmenden Bedeutung von Web-Anwendungen wird Rechnung getragen¹⁴.
- Durch die Zusammenfassung der Unterrichtsstunden beider Module steht mehr Zeit für die Entwicklung interessanterer Anwendungen zur Verfügung.
- Projekt-/problemorientiertes Lernen wird gefördert.

¹¹ Quelle: Seite 3 in:

BARTELTSEN, Jan ; BRAUER, Johannes ; REHDER, Heiko: Ideen für die Neugestaltung der Programmierausbildung durch Entwicklung von Webanwendungen / NORDAKADEMIE Hochschule der Wirtschaft. 2010 (2010-03). – Arbeitspapier

¹² auch veranlasst durch die inzwischen notwendig gewordene Reakkreditierung

¹³ Das Akronym steht für WEborientierter SmallTalk Programmierunterricht OnLine.

Zwar lenkt aus didaktischer Sicht die Beschäftigung mit graphischen Benutzungsoberflächen vom eigentlichen Lernziel eher ab und erhöht auch die „Umgebungskomplexität“, da der Umgang mit entsprechenden Frameworks zusätzlich erlernt werden muss. Aber möglicherweise kann die Motivation der Studierenden für ihre Programmierprojekte durch eine sichtbare Bedienoberfläche gesteigert werden, was wiederum dem Lernerfolg zugute kommen könnte.

¹⁴ WRIGHT, Alex: Ready for a Web OS? In: *Commun. ACM* 52 (2009), Nr. 12, S. 16–17. <http://dx.doi.org/http://doi.acm.org/10.1145/1610252.1610260>. – DOI <http://doi.acm.org/10.1145/1610252.1610260>. – ISSN 0001-0782

Die Ausgangslage und die Ziele dieses Projekts *WESTPOL* werden in einem Arbeitspapier¹⁵ eingehend erläutert.

2.2 Die Entwicklungsumgebung

Als Entwicklungsumgebung wurde in *Programmierung 1* von Beginn an *VisualWorks*¹⁶ von *Cincom* verwendet. Dabei handelt es sich um ein professionelles Werkzeug, das in keiner Weise auf die Belange des Lehrens und Lernens von Smalltalk ausgerichtet ist. So hat sich durch wiederholte Befragungen von Studierenden gezeigt, dass die „Reibungsverluste“ durch die Komplexität von *VisualWorks* doch recht hoch sind. Eine auf die Bedürfnisse von Anfängern zugeschnittene Entwicklungs- bzw. Lernumgebung¹⁷ steht für Smalltalk leider nicht zur Verfügung.

Zu den ursprünglichen Ideen von *WESTPOL* gehörte auch, ein Problem in Angriff zu nehmen, über das die Studierenden der *NORDPOL*-Kurse geklagt haben: das „Zuviele-offene-Fenster-Syndrom“. Es scheint das Hauptproblem beim Umgang mit *VisualWorks* zu sein. Es wird sicherlich noch dadurch verschärft, dass neben der Entwicklungsumgebung in einer Blended-learning-Umgebung mindestens ein Browser-Fenster geöffnet sein muss, um die aktuelle Kurslektion bzw. Programmieraufgabe lesen zu können.

ABHILFE sollte hier eine zeitweise von der Firma *Cincom* unter dem Namen *WebVelocity* angebotene Entwicklungsumgebung für *Seaside*-Smalltalk-Anwendungen schaffen, die selbst vollständig im Web-Browser läuft und mithilfe von *Seaside* realisiert wurde.

Leider erlaubte der Entwicklungsfortschritt bei *WebVelocity* den Einsatz in der Lehre vorerst nicht. Für die Zukunft ergeben sich möglicherweise neue Alternativen. Mit *tODE*¹⁸ und *Amber*¹⁹ sind weitere web-gestützte Smalltalk-Entwicklungsumgebungen im Entstehen. Daher wurde für den ersten Durchgang des *WESTPOL*-Kurses im Wintersemester 2011/2012 *VisualWorks* in Verbindung mit *Seaside* genutzt.

3 Programmierung 1 im Wintersemester 2011/2012

Der neue Kurs fand erstmals zwischen Januar und März 2012 mit drei Zenturien²⁰ des Manipels²¹ I1²² statt. Dabei erstreckte sich die Bearbeitungszeit des Blended-learning-Kurses über acht Wochen mit jeweils sechs Unterrichtsstunden pro Zenturie. Während des Unterrichts wurde jede Zenturie²³ in zwei Teilgruppen aufgeteilt, die zeitgleich in zwei benachbarten Rechnerräumen arbeiteten. Für jede Zenturie standen dabei drei Lehrpersonen zur Verfügung: zwei Dozenten und ein(e) Tutor(in).

Da die Lehrveranstaltung, wie eingangs geschildert, als Blended-learning-Kurs gestaltet war, bearbeiteten die Studierenden die Lerneinheiten des *moodle*-Kurses weitgehend selbständig. Währenddes-

¹⁵ BARTELSEN, Jan ; BRAUER, Johannes ; REHDER, Heiko: Ideen für die Neugestaltung der Programmierausbildung durch Entwicklung von Webanwendungen / NORDAKADEMIE Hochschule der Wirtschaft. 2010 (2010-03). – Arbeitspapier

¹⁶ <http://www.cincomsmalltalk.com>
Weitere Hinweise zur Auswahl der Entwicklungsumgebung sind zu finden in

BARTELSEN, Jan ; BRAUER, Johannes ; REHDER, Heiko: Ideen für die Neugestaltung der Programmierausbildung durch Entwicklung von Webanwendungen / NORDAKADEMIE Hochschule der Wirtschaft. 2010 (2010-03). – Arbeitspapier

¹⁷ vergleichbar mit *BlueJ* für Java:

BARNES, David J. ; KÖLLING, Michael: *Objects First with Java*. Pearson Education, 2008

Seaside ist ein Framework für die Entwicklung von Web-Anwendungen in Smalltalk (<http://seaside.st>)

DUCASSE, Stéphane ; RENGGLI, Lukas ; SHAFFER, David C. ; ZACCONE, Rick: *Dynamic Web Development with Seaside*. Square Bracket Associates, 2010

¹⁸ <http://code.google.com/p/tode/>

¹⁹ <http://amber-lang.net/>

²⁰ Eine *Zenturie* bezeichnet an der NORDAKADEMIE eine Gruppe mit max. 36 Studierenden, die ähnlich einem Klassenverbund in einer Schule gemeinsam lernen.

²¹ Ein *Manipel* bezeichnet alle Zenturien eines Studiengangs im selben Jahrgang, bspw. alle Wirtschaftsinformatikzenturien, die zum WS 2011/12 ihr Studium begonnen haben.

²² I1 bezeichnet das Manipel der Wirtschaftsinformatik-Studierenden mit Studienbeginn im WS 2011/12.

²³ Jede der drei Zenturien bestand aus 33 bis 36 Studierenden.

sen standen ihnen die drei Lehrpersonen bei Fragen und Problemen hilfreich zur Seite.

DER INHALT DER SEMESTERAUFGABE besteht in der Entwicklung eines Programms für die Simulation eines an Monopoly²⁴ angelehnten Spiels. Die Entwicklung des Programms wird dabei durch im Kurs vorgegebene, durch die Studierenden zunächst nur nachzuvollziehende Implementierungen und durch von den Studierenden selbständig zu lösende Teilaufgaben gesteuert. Um den Studierenden die Möglichkeit zu geben, ihre Lösungen zu überprüfen, werden im Kurs Unit-Tests bereitgestellt. Daneben stehen *moodle*-Tests²⁵ für die Selbstkontrolle von eher theoretischen Lehrinhalten bereit. Diese werden im Kursverlauf sporadisch gemeinsam erörtert.²⁶

Als zusätzliche Hilfestellung – insbesondere für den Umgang mit der Entwicklungsumgebung – stehen eine Reihe von Screenshots zur Verfügung, da das ein- oder auch mehrmalige Vorführen bestimmter Arbeitsvorgänge von den Studierenden zwar nachvollzogen, aber nur schwer behalten werden kann.

AM ENDE JEDES KURSTAGS wurden die Studierenden der Zenturie zu einer viertelstündigen Abschlussdiskussion versammelt. Dabei wurde z. B. über

- Schwierigkeiten beim Verständnis von Lerneinheiten,
- Schwierigkeiten beim Umgang mit der Entwicklungsumgebung,
- Ungereimtheiten im *moodle*-Kurs oder
- den erreichten Arbeitsfortschritt

gesprochen. Die Ergebnisse der Diskussion wurden von den Studierenden in einem *moodle*-Wiki selbst protokolliert. Die Zusammenfassung dieser Protokolle wird in Abschnitt 4 diskutiert.

4 Die gewonnenen Erfahrungen

In diesem Abschnitt wird versucht, die in den Wikis festgehaltenen Äußerungen der Studierenden zu gruppieren und geeignet zusammenzufassen. Dabei kommt es bei einigen wenigen Punkten zu Wiederholungen, da sich manche Kritikpunkte nicht nur einer Überschrift zuordnen lassen.

4.1 Blended-learning im Allgemeinen

In diesem Abschnitt werden zunächst einige grundlegende Kritikpunkte erörtert, die mit der Durchführung des Kurses in der hybriden Unterrichtsform zusammenhängen. Sich daraus ergebende Verbesserungsvorschläge für einzelne Themenblöcke werden in Abschnitt 4.4 erneut aufgegriffen.

SELBSTSTÄNDIGKEIT

Zu Beginn des *moodle*-Kurses wird sehr detailliert beschrieben,

²⁴ <http://de.wikipedia.org/wiki/Monopoly>

²⁵ in der Regel Multiple-choice-Aufgaben

²⁶ Dazu wurden die beiden Teilgruppen in der Regel wieder zusammengeführt, was durch die Nachbarschaft der Unterrichtsräume auch spontan problemlos durchführbar war.

wie die Studierenden bei der Programmierung vorgehen sollen, es gleicht schon fast einer „Schritt-für-Schritt-Anleitung“. Einige Studierende bemängelten dies und hatten den Eindruck, dass hier unter anderem nach der Methode „Copy & Paste“ vorgegangen werden konnte oder, aufgrund des Unterrichtstempos, vielleicht auch vorgegangen werden musste. Als die erste Aufgabenstellung kam, bei der die Studierenden weitestgehend alleine vorgehen sollten, kam die Rückmeldung, dass dieser Übergang zu abrupt war. Zusätzlich waren die Aufgabenstellungen zu ungenau. Die Studierenden bemängelten das Fehlen von Beispiellösungen, um die Ergebnisse bzw. das Vorgehen kontrollieren zu können. Um ein weiteres „Copy & Paste“ vermeiden zu können, wurde vorgeschlagen, mehrere Besprechungszeiten zu integrieren, in denen über Lösungen gesprochen wird, u. a. auch um Missverständnisse zu vermeiden.

Es wurde der Wunsch geäußert, das „Ziel“²⁷ deutlich aufzuzeigen.

²⁷ vermutlich das Lernziel oder das Ziel einer Aktivität

moodle-KURS

Bezogen auf den *moodle*-Kurs gab es zu Beginn einige Probleme mit Fehlern in den Texten und Dateien. Dazu gehörten sowohl Rechtschreibfehler, als auch einige Fehler in der Funktionsbeschreibung. Diese wurden jedoch im Laufe des Kurses behoben. Von den Studierenden kamen unterschiedliche Rückmeldungen bezüglich des *moodle*-Kurses. Einige waren mit dem Aufbau des Kurses sehr zufrieden und fanden den Aufbau logisch. Lediglich die Textdateien sollten gekürzt werden. Die Erfahrung der Dozenten zeigt jedoch, dass für späteres Nachschlagen ausführlichere Texte und häufige Überschriften praktischer sind. So kam die Idee auf, dass am Ende des Kurses erneut beleuchtet werden sollte, ob lange Texte weiterhin stören. In einem Nachgespräch zum Kurs wurde auch von Seiten ausgewählter Studierender die Erfahrung der Dozenten bestätigt.

Da der Blended-learning-Kurs von mehreren Autoren bearbeitet wurde, gibt es unterschiedliche Verfahren, wie Aufgaben beschrieben werden. In einem Verfahren werden bspw. alle möglichen Wege, eine Aufgabe zu lösen erörtert, ohne dass den Studenten aufgezeigt wird, welche die zu wählende ist. Hier sollten zuerst nur die unterschiedlichen Varianten genannt und danach direkt klargestellt werden, welche der Möglichkeiten im weiteren Verlauf des Kurses benötigt wird. Erst danach sollten die unterschiedlichen Varianten im einzelnen vorgestellt werden.

Grundsätzlich waren mehr Beispiele oder zur Selbstkontrolle je eine kurze Zusammenfassung am Ende eines Themenblocks erwünscht.

TEMPO

Zum Tempo des Kurses gab es widersprüchliche Aussagen. Die einen empfanden das Tempo als gut, andere als zu hoch, um „selbst

alles zu hinterfragen“. Es wurde darüber hinaus angemerkt, dass ein „hohes Lernlevel“ existiert, bei dem mit Smalltalk und HTML viel Neues gelernt werden muss.

UNTERSTÜTZUNG

Die Studenten waren mit der Unterstützung durch die Dozenten zufrieden und lobten die Kommunikation. Im Laufe des Kurses wurden in einem Teil der Lehrveranstaltungen zu Beginn und in der Mitte des Tages Theorieblöcke durchgeführt. Diese wurden von den Studierenden sehr positiv bewertet. Es wurde gewünscht, dass solche Theorieblöcke bereits seit der ersten Vorlesung stattgefunden hätten. Es kam zusätzlich der Wunsch, Theorieblöcke zu Beginn des Kurses vermehrt stattfinden zu lassen, um die Grundlagen von Smalltalk besser kennenzulernen, bevor die aktive Programmierung beginnt.

Eine weitere Hilfestellung durch die Dozenten war die Bereitstellung von Screencasts. Diese wurden ebenfalls durch die Studierenden positiv bewertet, da zu einem späteren Zeitpunkt immer noch einmal darauf zurückgegriffen werden kann.

4.2 Inhaltliche Themen

In diesem Abschnitt werden Anmerkungen aus den Wikis zusammengefasst, die sich auf die Lerninhalte *Smalltalk* und *CSS* beziehen.

DER EINSTIEG IN DIE PROGRAMMIERSPRACHE SMALLTALK fiel offenbar einigen Studierenden aufgrund der für die objektorientierte Programmierung typischen Aufteilung eines Programms in viele kleine Methoden schwer. Dadurch ging für einige am Anfang der Überblick über die Anwendung verloren.

Verschiedentlich wurde auf Probleme mit der Syntax von Smalltalk hingewiesen, die teilweise von der Syntax anderer, manchen Studierenden bereits bekannten Programmiersprachen, abweicht. In diesem Zusammenhang ist der Vorschlag bemerkenswert, im Kurs vorgegebene Smalltalk-Methoden als „grammatikalisch richtigen Text“ aufzuschreiben.²⁸

Der frühe Einstieg in die Verwendung von *Seaside* bringt die für Anfänger relativ komplexen `renderContentOn:`-Methoden mit sich. Diese verwenden fast alle Bestandteile der Smalltalk-Syntax, was es offenbar schwierig macht, diese von *Seaside*-spezifischen Nachrichten zu unterscheiden. Hier ist offenbar eine „behutsamere“ Heranführung notwendig.

Darüber hinaus wird von den Studierenden eine Übersicht der wichtigsten Nachrichten der Smalltalk-Klassenbibliothek im Allgemeinen und des *Seaside*-Frameworks im Besonderen gefordert.

Häufig beklagten Studierende, dass im Web angeblich kaum Informationen über Smalltalk zu finden wären. Diesem Eindruck muss frühzeitig durch Hinweis auf einschlägige Webseiten entge-

²⁸ Hierfür eignet sich insbesondere die Syntax der Schlüsselwort-Nachrichten gut.

Insbesondere die für Smalltalk essentielle Unterscheidung von runden und eckigen Klammern bereitet Schwierigkeiten.

gen getreten werden.

Das Zusammenspiel mehrerer *Seaside*-Komponenten kann schnell zu einem Verlust an Übersicht führen. Eine im Verlauf des Kurses bereitgestellte Grafik zur Verdeutlichung der Zusammenhänge der Komponenten wurde von den Studierenden sehr begrüßt. Es wurde aber angeregt, dieses zu einem früheren Zeitpunkt einzubringen.

MIT CSS wurde auf der Plattform *Seaside* das Weblayout gestaltet. Die Studierenden des Studiengangs Wirtschaftsinformatik wurden im Vorfeld informiert, dass für das Modul Programmierung grundlegende HTML-Erfahrungen vorausgesetzt werden. Wer sich diese bisher nicht aneignen konnte, hatte die Möglichkeit, vor der ersten Theoriephase ein HTML-Seminar an der NORDAKADEMIE zu besuchen. Dort wurden die erwarteten Grundlagen in zwei Tagen kompakt vermittelt.

CSS war somit ein neues Thema im Kurs, obgleich auch hier bei einigen Studierenden bereits Vorkenntnisse vorhanden waren. Im Blended-learning-Kurs Westpol wurde daher ein CSS-Tutorial angeboten, mit dessen Hilfe die Studierenden gute Lernerfolge erzielen konnten, denn damit konnten die gelernten Inhalte eigenständig und ohne große Hilfe angewandt werden. Nebenbei wurde regelmäßig darauf hingewiesen, dass die eigengeschriebenen HTML- und CSS-Codes W3C-valide sein mussten. Hierfür wurde den Studierenden gezeigt, wie man seinen eigenen CSS-/HTML-Code auf der Homepage des W3C überprüfen und validieren lassen kann.

Eine geringe Anzahl der Studierenden empfahl im Feedback, dass der Themenblock zu CSS auch hätte früher im Kurs behandelt werden können. Und mehrere Personen waren der Ansicht, dass die Reihenfolge einiger CSS-Themenblöcke geändert werden könnte. Darüber hinaus wurden einige Aufgabenstellungen teilweise als zu undeutlich und missverständlich empfunden. Wenn im Vorhinein bekannt ist, dass es zu bestimmten Problemen kommen kann, sollte im Text darauf hingewiesen werden. Hier werden für den nächsten Durchgang sicherlich noch Überarbeitungen stattfinden.

Jedoch war sich die Mehrheit darüber einig, dass im Grunde die Anordnung dieses Themenblockes für das Bearbeiten der Übungen nicht so relevant wie andere Themenblöcke war.

Im Allgemeinen lässt sich also festhalten, dass das CSS-Tutorial auf eigene HTML-Kenntnisse (und somit auch eigene CSS-Kenntnisse) aufbaut, beziehungsweise diese auch Voraussetzung für den Kurs Westpol sind. Das CSS-Tutorial frischt das eigene Wissen auf.

4.3 *VisualWorks/Seaside*

Die Nutzung der Smalltalk-Entwicklungsumgebung VisualWorks führt zu einer Steigerung der Komplexität, die bei Programmieranfängern durch die hohe Anzahl separater Fenster zu Tage tritt. Bei Studenten mit Programmiererfahrung sind es insbesondere

die ungewöhnlichen Tastaturkürzel (<Strg>-<S> übersetzt zwar eine Methode, speichert diese aber nicht auf der Festplatte). Weiterhin unterscheidet sich die Organisation des Quellcodes sehr stark von bekannteren Programmierumgebungen. Ein Wunsch der Studierenden war hier, eine bessere Einführung in die Entwicklungsumgebung, welche allerdings nur durch entsprechende Übungsaufgaben sinnvoll umgesetzt werden kann. Dabei sollte auch erklärt werden, wie das Fenster aufgebaut ist und wo man welche Informationen findet.

Ein Kritikpunkt bei den VisualWorks betreffenden Anleitungen im *moodle*-Kurs war, dass diese z.T. nicht aktuell bzw. genau genug waren. Dies ist zum einen auf ältere Screencasts zurück zu führen, die aufgrund des erheblichen Produktionsaufwands nicht für jede neue VisualWorks-Version erneuert werden können.

4.4 Kritik an einzelnen Themenblöcken

Dieser Abschnitt beschäftigt sich mit konkreten Verbesserungsvorschlägen der Studierenden an einzelnen Themenblöcken. Dabei kommen Kritikpunkte zur Sprache, die teilweise bereits in Abschnitt 4.1 erörtert wurden.

Es wurde oft angemerkt, dass bei mehreren Themenblöcken die Aufgaben ungenau gestellt wurden und daher nicht erkannt wurde, wie man die Aufgabe bearbeiten soll. Als Beispiel wurden Themenblock 3 („Einstieg in *Seaside*“) und 5 („Cascading Style Sheets (CSS)“) genannt. Bei Themenblock 5 wurde auch gesagt, dass man sich „regelrecht alleingelassen“ gefühlt hat, da im Gegensatz zu den vorherigen Themenblöcken zu wenig Instruktionen vorhanden waren.²⁹ Ähnliches wurde auch zu den Themenblöcken 10 („Eingabeprüfung“) und 11 („Fileupload“) angemerkt. Es gab dennoch einige Gegenstimmen, die das selbständige Herausfinden der Lösung als Herausforderung sahen. Doch allgemein wurde darin übereingestimmt, dass Zwischenlösungen sehr sinnvoll wären, falls man mal nicht weiter kommt. Dazu kommt, dass vielen Kursteilnehmern das Ziel einiger Themenblöcke nicht klar ersichtlich war und daher einige Probleme auftraten. Diese bestanden z. B. darin, dass man vorarbeitete und dann feststellte, dass eine andere Lösung als die eigene gefordert war. Dafür wäre eine klarere Aufgabendefinition sehr wünschenswert.

Es kam der Vorschlag, die Themenblöcke 4 („Einstieg in NoNopoly“) und 5 („Cascading Style Sheets (CSS)“) in der Reihenfolge zu tauschen, damit man den in der Übungsaufgabe zu Themenblock 3 („Einstieg in *Seaside*“) erstellten „Steckbrief“ gleich mit CSS aufbereiten kann.

Im Verlauf der Themenblöcke 4 und 5 – bzw. schon bei Themenblock 3 – wäre es hilfreich, den Unterschied zwischen der `canBeRoot`- und der `initialize`-Methode zu verdeutlichen. Genauso wäre eine frühere Vorstellung der *Seaside*-Nachrichten `call:`, `callback:` und `answer:` besser für das Gesamtverständnis.

Ein weiteres Problem im Rahmen des Unterrichts war die Auswahl des Ports für den *Seaside*-Server. Zuerst war dieser 7777, in einem Rechnerraum musste er aber aufgrund konkurrierender Software in 7778 geändert werden. Dies wird allerdings in folgenden Kursen nicht der Fall sein, da die konkurrierende Software mittlerweile vom System entfernt wurde.

²⁹ vgl. hierzu auch den CSS-Teil in Abschnitt 4.2

Viele Studierende fanden die anfänglichen Themenblöcke zu umfangreich an Text. Man wurde von der „Masse an Informationen regelrecht erschlagen“. Für die spätere Arbeit hat sich aber herausgestellt, dass sie manchmal sehr gut zum Nachschlagen geeignet waren. Dazu wurde auch eine Zusammenfassung von Befehlen vorgeschlagen, damit man eine bessere Übersicht für das spätere Programmieren hat. Des Weiteren wäre für den Anfang und für das spätere Nachschlagen eine kurze Zusammenfassung am Ende der meisten Kapitel sehr hilfreich. Außerdem wäre bei manchen Programmtexten der ersten Themenblöcke eine Ausformulierung in „lesbaren Text“ sehr hilfreich, um ihn besser zu verstehen. Bei den Formulierungen wurde auch gewünscht, dass es einheitliche Klassen-/Methodennamen³⁰ gibt.

Im Abschnitt *Refactoring der renderContentOn:-Methode* des Themenblocks 6 („Zusammenspiel mehrerer Komponenten in *Seaside* (call-answer)“) stehen Programmausschnitte zur Verfügung, die allerdings einige Fehler enthalten, die noch korrigiert werden müssen.³¹

Es fielen in den meisten Themenblöcken mehrere Rechtschreibfehler auf.

Bezüglich des Tests Nummer 3 in Themenblock 10 („Eingabeprüfung“) wurden die Studierenden zwar darauf hingewiesen, dass die Ausführung des Tests einen Fehler hervorrufen würde, der selbständig behoben werden sollte. Aber dazu waren einige Studierende nicht in der Lage, da noch nicht bekannt war, wie man die Fehler eines Tests auswertet. Dies sollte vorher gezeigt werden. Allgemein wurde überlegt, dass dieser Themenblock schon früher behandelt werden sollte, vor allem wegen der Variablen und deren Gültigkeitsbereiche.

Allgemein sind die Screencasts und Videos als sehr hilfreich angesehen worden.

4.5 Prüfungsvorbereitung

Allgemein kann gesagt werden, dass sich der *moodle*-Kurs gut eignet, um bestimmte Inhalte für die Prüfung zu wiederholen und nachzuschlagen. Der Kurs allein war vielen Studierenden aber als einziges Vorbereitungsmittel nicht ausreichend. Das empfohlene Buch³² fanden viele Studierende als geeigneter, um für die Klausur zu lernen und Inhalte noch einmal nachzuschlagen. Ebenfalls positiv bewerteten die Studierenden die „Theorieblöcke“, in denen gemeinsam Smalltalk-Grundlagen besprochen und wiederholt wurden. Diese fanden ab Mitte des ersten Semesters zu Beginn jeder Unterrichtseinheit (morgens und mittags) statt. Diese Theorieblöcke wären aber strikt von Anfang an gewünscht gewesen, lautet ein Resultat aus den Feedbackgesprächen. Auf diese Weise werde die Theorie besser vertieft aber bezüglich der Vorbereitung sei zu spät damit angefangen worden. Dadurch fühlten sich viele Studierende ebenfalls nicht gut vorbereitet.

Aus diesen Anmerkungen könnte die Schlussfolgerung gezogen werden, dass die Nutzungsform eines Themenblocks (erstmaliges Durcharbeiten während der Lehrveranstaltung vs. späteres Nachschlagen bzw. Nacharbeiten) Einfluss auf seine Gestaltung haben müsste. Dies dürfte aber aus Aufwandsgründen kaum umsetzbar sein.

³⁰ z. B. `NoNopolyFiles` vs. `NoNopolyFileLibrary`

³¹ `renderUeberschrift`: html sollte `renderUeberschrift0n`: html, `renderEingabeSpielername`: html `renderEingabeSpielername0n`: html heißen.

³² BRAUER, Johannes: *Grundkurs Smalltalk - Objektorientierung von Anfang an*. 3. Auflage. Vieweg+Teubner Verlag, 2009

Durch das Umsetzen der Vorlesung als Blended-Learning-Kurs war im Laufe des ersten Semesters den Studierenden nicht klar, wie die Prüfungsfragen aussehen könnten. Als Lösung wurde angeboten, Beispielfragen mitzubringen. Die Verteilung der Klausur-Beispielaufgaben fand den meisten Kursteilnehmern allerdings zu kurzfristig vor der Klausur statt. Das hatte zur Folge, dass es keine Zeit mehr zur Besprechung im Plenum gab. Man hatte somit auch keine Chance, Fragen, die man zu den Aufgaben hatte, zu besprechen und man musste somit versuchen, die Lösungsvorschläge³³ alleine zu verstehen. Fragen, die gestellt wurden, wurden vielen manchmal unzureichend beantwortet, was vermutlich auch an der mangelnden Zeit dazu gelegen haben könnte. Andere waren hingegen der Meinung, dass die Klausuraufgaben mit den gelernten Inhalten aus dem Kurs gut machbar waren. Einig waren sich aber alle Studenten darin, dass man für die letzte Aufgabe, bei der ein kleines Programm³⁴ auf Papier geschrieben werden sollte, nicht gut vorbereitet war. Vorschläge zur Verbesserung in zukünftigen Kursen liefen darauf hinaus, dass man den Kurs nicht von Anfang an am Rechner macht, sondern mit Papier und Stift seine ersten Schritte macht.

Allgemein kann gesagt werden, dass viele Studierende mit der Vorbereitung auf die Prüfung zufrieden waren. Allerdings fiel es gerade den Studierenden, die vorher noch nie eine Programmiersprache gelernt hatten, schwer, in dem Kurs mitzukommen. Ein Ziel und Vorteil des Blended-Learning-Kurses ist zwar, dass jeder sein eigenes Tempo wählen kann, aber vorausblickend auf die bevorstehende Prüfung waren es gerade diese Studierenden, die sich gewünscht hätten, viel früher und öfter abgeholt worden zu sein.

5 Schlussfolgerungen

Die Analyse der Feedback-Wikis ergab eine große Anzahl von Vorschlägen für Detailverbesserungen an einzelnen Lektionen des Westpol-Kurses bzw. an deren Abfolge. Daraus kann man die Schlussfolgerung ziehen, dass grundsätzlich das Abhalten des Kurses in einer hybriden Form durchaus Zustimmung findet. Auch die in Abschnitt 4.1 eher grundsätzlichen Kritikpunkte lassen nicht darauf schließen, dass ein Blended-Learning-Kurs für den Programmierunterricht abgelehnt würde.

Eine wichtige Erkenntnis besteht darin, dass der Wunsch der Studierenden die „Plenumsanteile“³⁵ eines Hybrid-Kurses auszuweiten, deutlich geworden ist. Das ist von den Dozenten des Kurses so nicht vorhergesehen worden. Dieser Wunsch erscheint aber durchaus plausibel, wenn man bedenkt, dass es sich zwar um einen Online-Kurs handelt, der weitgehend selbständig bearbeitet werden kann und soll, die Studierenden aber aufgrund der für die NORDAKADEMIE geltenden Studienbedingungen in der Lehrveranstaltung Anwesenheitspflicht haben.

Die Ausführungen in Abschnitt 4.2 lassen darauf schließen,

³³ Ein Tutor hatte sich die Mühe gemacht, Lösungsvorschläge auszuarbeiten und allen zukommen zu lassen. Diese waren jedoch zum Teil fehlerhaft.

³⁴ eine Art Auftragsbearbeitung mit verschiedenen Methoden

³⁵ also die Zeiten, in denen die Studierenden vom Dozenten in der Gruppe unterrichtet werden

dass der Kurs bezüglich der Einführung der Programmiersprache Smalltalk vermutlich zu „steil“ war. Hier sind Korrekturen bzw. Ergänzungen angebracht. An dieser Stelle gibt es auch einen Berührungspunkt zu der in Abschnitt 4.5 erörterten, von einigen Studierenden als unzulänglich betrachteten Prüfungsvorbereitung. Smalltalk-Methoden einmal mit Papier und Bleistift aufzuschreiben, könnte sich als zielführend erweisen.

Probleme, die die Studierenden im Umgang mit der Entwicklungsumgebung *VisualWorks*³⁶ haben, werden sich nicht vollständig beseitigen lassen, solange man bei ihr bleibt. Es handelt sich um ein Werkzeug für professionelle Software-Entwickler, das auf die Bedürfnisse von Programmieranfängern oder „Umsteigern“ keine Rücksicht nimmt. Ob es Alternativen gibt, die weniger Schwierigkeiten bereiten, bliebe erst noch zu untersuchen. Bis dahin wird es notwendig sein, die Unterstützung der Studierenden z. B. durch aktualisierte und neue Screencasts weiter zu entwickeln.

Die im Abschnitt 4.4 zusammengefassten Kritikpunkte an einzelnen Themenblöcken werden sich größtenteils bei einer Überarbeitung des Kurses berücksichtigen lassen. Die Kritikpunkte geben auch Anlass, die Balance zwischen im Kurs vorgegebenen Lösungsvorschlägen³⁷ und von den Studierenden selbst zu entwickelnden Lösungen zu überdenken. Die diesbezüglichen Äußerungen der Studierenden zeigen aber auch, dass man hier wohl nicht immer allen gerecht werden kann, was angesichts der Heterogenität der Vorkenntnisse aber auch nicht überraschen kann.

Die Diskrepanz zwischen den in einem Programmierkurs zu vermittelnden Fähigkeiten und Fertigkeiten einerseits und dem in einer Klausur Prüfbareren andererseits wird sich kaum beseitigen lassen. Gleichwohl ist es sinnvoll³⁸, die Studierenden in Zukunft zu einem früheren Zeitpunkt mit der Prüfungsform bzw. den darin vorkommenden Aufgabentypen³⁹ vertraut zu machen.

6 Ausblick

Die in diesem Arbeitspapier zusammengetragenen Erfahrungen werden von den Dozenten des Kurses für eine Überarbeitung des Kurses genutzt. Einige Hinweise konnten bereits im Sommersemester 2012 im Modul *Programmierung 2*⁴⁰ umgesetzt werden. Auch für diesen Kurs liegen inzwischen einige Erfahrungen vor, die wieder in Wikis protokolliert wurden. Hier zeigt sich, dass die Zahl der geäußerten Kritikpunkte vergleichsweise gering ist. Das mag daran liegen, dass Vorschläge der Studierenden aus *Programmierung 1* teilweise bereits umgesetzt wurden, die Studierenden sich an die Kursform gewöhnt haben, die Heterogenität der Vorkenntnisse nach dem ersten Semester etwas „eingeebnet“ werden konnte oder die Studierenden einfach den Eindruck hatten, dass inzwischen „alles gesagt ist“.

Trotzdem scheint es sinnvoll, bei der Durchführung des Kurses *Programmierung 1* im Wintersemester 2012/2013 bei der Evaluation

Hier kann man natürlich auch die Frage stellen, ob die Prüfungsform Klausur für einen derartigen Kurs adäquat ist.

³⁶ vgl. Abschnitt 4.3

³⁷ für Programmieraufgaben

³⁸ wie in Abschnitt 4.5 angesprochen

³⁹ „Programmieren ohne Rechner“

⁴⁰ Dieses Modul setzt das Modul *Programmierung 1* inhaltlich fort und ist ebenfalls als Blended-learning-Kurs aufgebaut.

in ähnlicher Weise vorzugehen, um dann in einer vergleichenden Studie zu untersuchen, inwieweit die Steigerung der Zufriedenheit von Studierenden mit einem derartigen Kurs auf diese Weise erfolgreich sein kann.

7 *Danksagung*

Die Autoren danken den Studierenden des Manipels I11 für die engagierten Diskussionen und die bewiesene Geduld beim Umgang mit einem neuen und daher alles andere als perfekten Blended-learning-Kurs.

Literatur

- [1] BARNES, David J. ; KÖLLING, Michael: *Objects First with Java*. Pearson Education, 2008
- [2] BARTELTSEN, Jan ; BRAUER, Johannes: Kooperatives Lernen mit moodle in der Programmierausbildung / NORDAKADEMIE Hochschule der Wirtschaft. 2008 (2008-6). – Arbeitspapier
- [3] BARTELTSEN, Jan ; BRAUER, Johannes ; REHDER, Heiko: Abschlussbericht für das Projekt NORDPOL / NORDAKADEMIE Hochschule der Wirtschaft. 2010 (2010-02). – Arbeitspapier
- [4] BARTELTSEN, Jan ; BRAUER, Johannes ; REHDER, Heiko: Ideen für die Neugestaltung der Programmierausbildung durch Entwicklung von Webanwendungen / NORDAKADEMIE Hochschule der Wirtschaft. 2010 (2010-03). – Arbeitspapier
- [5] BARTELTSEN, Jan-Ivar: *Entwurf einer kooperativen Lehrveranstaltung in moodle für die Grundausbildung in der Programmierung*. Master-Thesis an der FernUniversität Hagen, 2008
- [6] BENNEDSEN, Jens ; CASPERSEN, Michael E. ; KLLING, Michael: *Reflections on the Teaching of Programming: Methods and Implementations*. 1. Springer Publishing Company, Incorporated, 2008. – ISBN 3540779337, 9783540779339
- [7] BRAUER, Johannes: *Grundkurs Smalltalk - Objektorientierung von Anfang an*. 3. Auflage. Vieweg+Teubner Verlag, 2009
- [8] BRAUER, Johannes ; KRASEMANN, Hartmut ; CRASEMANN, Christoph: Objekte statt Klassen zuerst. In: *Forschung für die Wirtschaft*. NORDAKADEMIE Hochschule der Wirtschaft, 2011
- [9] DUCASSE, Stéphane ; RENGGLI, Lukas ; SHAFFER, David C. ; ZACCONE, Rick: *Dynamic Web Development with Seaside*. Square Bracket Associates, 2010
- [10] WRIGHT, Alex: Ready for a Web OS? In: *Commun. ACM* 52 (2009), Nr. 12, S. 16–17. <http://dx.doi.org/http://doi.acm.org/10.1145/1610252.1610260>. – DOI <http://doi.acm.org/10.1145/1610252.1610260>. – ISSN 0001-0782