

Objektorientierte Programmierung - Kompetenzerwerb im Mastery Model des Inverted Classroom

Aufbau und Akzeptanz der Kursumsetzung

Annett Thüring¹ und Kathrin Jäger²

Abstract: Das Modul Objektorientierte Programmierung wird an der Martin-Luther-Universität Halle-Wittenberg für ca. 300 Studierende im ersten Semester klassisch als Präsenzvorlesung mit Übung angeboten. Vor dem Hintergrund steigender und hoher Abbruchquoten wurde für diese Lehrveranstaltung ein lernerzentriertes Blended-Learning-Konzept entwickelt, das auf dem Inverted Classroom Mastery Modell (ICMM) basiert. Das didaktische Konzept setzt an der Praxis an, fokussiert auf den Kompetenzerwerb und orientiert auf eine Förderung selbstregulierten Lernens unter Einbeziehung der Rolle des Lehrenden in diesem Ansatz. Nach einer ersten Testphase im WS 2019/20 können Ergebnisse des Einsatzes des ICMMs und praktische Erfahrungen bei der Nutzung der Lernumgebung positiv diskutiert werden.

Keywords: Programmierkurs, Inverted Classroom Mastery Modell, Kompetenzmessung

1 Hintergrund

Objektorientierte Programmierung (OOP) ist eine Lehrveranstaltung für Studierende des ersten Semesters, die an der Martin-Luther-Universität Halle-Wittenberg klassisch in Präsenz sowohl Vorlesung als auch Übung umfasst. Studierende unterschiedlicher Studienrichtungen wie z.B. Informatik, Mathematik, Wirtschaftsinformatik, Lehramt Informatik und Physik können das Modul obligatorisch oder wahlobligatorisch in ihr Studium einbringen. In den letzten Jahren wurden sinkende Lernerfolge und in der Fortführung hohe Kursabbrüche seitens der Studierenden verzeichnet. Diese werden vor allem auf die zunehmende Heterogenität der Zielgruppe bzgl. des individuellen Vorkenntnisstandes aber auch auf die Bedürfnisse und Wünsche der Studierenden nach individualisierten Studienwegen und flexiblen Studienmodellen zurückgeführt. Eine vorangestellte Analyse des Modells Vorlesung und Übung in Präsenz ergab, dass eine passive Vorlesung zur Vermittlung von Fähigkeiten zudem ungeeignet ist. Vor diesem Hintergrund fokussiert eine Umstrukturierung des Moduls OOP die Einbeziehung und Passung zeitgemäßer Lehr-/Lernformen.

¹ Martin-Luther-Universität Halle-Wittenberg, Institut für Informatik, Von-Seckendorff-Platz 1, 06120 Halle (Saale), annett.thuering@informatik.uni-halle.de

² Martin-Luther-Universität Halle-Wittenberg, Zentrum für multimediales Lehren und Lernen (LLZ), Hoher Weg 8, 06120 Halle (Saale), kathrin.jaeger@llz.uni-halle.de

2 Ziele

Hinsichtlich des individuellen Vorkenntnisstandes der Studierenden können angebotene Lehrveranstaltungen nur effizient durchgeführt werden, wenn ein möglichst homogenes Wissensniveau aller Lernenden erreicht werden kann. In der Praxis haben sich hierzu Konzepte mit einer Integration von E-Learning-Anteilen in Form von Blended-Learning bewährt. Zur Individualisierung des Modulangebotes basiert die vorgestellte Umstrukturierung der Lehrveranstaltung auf einem lernerzentrierten dreiphasigen Inverted Classroom Mastery Modell (ICMM) [Ha15], das neben onlinebasierter Wissensvermittlung und problembasierten Austausch- und Übungsphasen in Präsenz, formative Assessments integriert und Lernerpräferenzen berücksichtigt. Unterstützend werden universelle Ansätze aus Motivationsmodellen³ herangezogen sowie Planungs- und Regulationshilfen für selbstreguliertes Lernen⁴ implementiert. Umsetzungen stützen sich auf didaktische Konzepte mit einer Balance zwischen Instruktion und Konstruktion sowie auf eine Passung der Lehr-/Lernformen und Interventionen zu einer stufenweisen Vermittlung der Fachkompetenz. Dabei werden die technologiegestützten Elemente auf das Verstehen immer komplexer werdender Problemstellungen ausgerichtet. Ein ähnlicher Ansatz wird auch von Fässler verfolgt [Fä07], wobei hier Studierende nur am Ende Einblick in den Kompetenzstand erhalten. Ziel dieses Kurses ist es, den Lernerfolg schrittweise zu ermöglichen, zu messen und zudem transparent den Studierenden den Lernstand sichtbar zu machen. Rückmeldungen zum Lernstand fokussieren die Selbstregulation, sodass das Kursziel, die Fähigkeit des Programmierens, erreicht werden kann.

3 Das didaktische Konzept

Das didaktische Konzept orientiert auf eine verstärkte Einbeziehung onlinebasierter digitaler Elemente, wobei ein intensiver Analyseprozess für die Auswahl der Inhalte und Aktivitäten vorangeht. Die Grundstruktur der 15 Kurseinheiten im Semesterzyklus wird beibehalten und zur Planung die umfassende didaktische Designplanung nach Reinmann [Re15] herangezogen. Im ersten Schritt erfolgte eine Formulierung wissens- und kompetenzorientierter Lernziele⁵ sowohl der gesamten Lehrveranstaltung als auch jeder Kurseinheit. Die Zuordnung der Lern- und Kompetenzziele basiert auf der Bloom'schen Lernzieltaxonomie [AK01] sowie den von Weicker et al. [WWD06] formulierten Schlüsselkompetenzen für Informatiker.

³ Grundlegende Ansätze im Modulablauf integrieren Aspekte aus dem ARSCS-Modell von Keller [KK87], dem Motivationsprogramm von Fries [Fr02] und dem ECOLE-Modell (Emotionale und kognitive Aspekte des Lernens) von Gläser und Zikuda [Gl05].

⁴ Studierende benötigen bei selbständiger Wissensaneignung in Online-Phasen Selbstregulationsstrategien. [ZS01]

⁵ Empfehlungen der Gesellschaft für Informatik für Bachelor- und Masterprogramme im Studienfach Informatik in denen Lernziele, Kompetenzen und Empfehlungen im Studienfach Informatik verankert sind. [Zu16]

Die Lern- und Kompetenzziele geben dem Lehrenden eine Grundlage zur Strukturierung, Selektion und Reduktion der Inhalte. Alle Inhaltselemente und Aufgaben des Kurses werden danach ausgewählt und zugeordnet und die Prüfungsaufgaben abgeleitet⁶. Sichtbare Lern- und Kompetenzziele im Modulablauf, sowie eine klare Struktur bieten dem Studierenden zudem Transparenz, Verbindlichkeit und Orientierungshilfe.

Für die eigentliche Inhaltserstellung und den Ablauf der verschiedenen Lernphasen werden unterschiedliche Modelle und Methoden herangezogen, um Aufmerksamkeit und Selektion sowie Verarbeitung und Integration des Wissens in die individuelle Wissensstruktur zu ermöglichen und zu fördern. Als Hauptlehrmethode über den gesamten Kurs wird das Cognitive Apprenticeship-Modell (CAM) [CBN89] genutzt. Dieses Modell wurde an die traditionelle Handwerkslehre angelehnt und als schulisches Unterrichtsmodell entwickelt. Es geht davon aus, dass der Lehrende für ein Problem oder eine Aufgabe die Lösung vorführt und gleichzeitig die kognitiven aber auch metakognitiven Schritte artikuliert. Lernende versuchen im Anschluss selbst Probleme gleicher Klasse auf die gleiche Art zu lösen und der Lehrende zieht sich zurück. Im Vergleich weist das CAM eine sehr viel höhere und auch konkretere Intensität an instruktionaler Unterstützung auf als andere Methoden wie z. B. Anchored Instruction. Dieses Ausmaß an Unterstützung steht im universitären Lehrkontext zumeist nicht im Fokus. Für Studierende im Erstsemester, die noch wenig Erfahrung mit selbstreguliertem Lernen haben, kann diese Methode gewinnbringend sein. Eine Diskussion über die Handlungsschritte zwischen Lehrenden und Studierenden aber auch zwischen Studierenden untereinander ermöglicht zudem in die jeweilige Fachkultur hineinzuwachsen und zukünftiges funktionales Verhalten bezüglich der Programmieraufgaben zu erlernen. Abb. 1 zeigt die instruktionale Sequenz der didaktischen Bausteine *modeling*, *scaffolding*, *fading*, *coaching*, *exploration*, *articulation* und *reflection* über den Kursverlauf (Online und in Präsenz).

Umsetzung im Lernmanagementsystem ILIAS⁷

Alle Online-Kursmaterialien⁸ werden im ILIAS zur Verfügung gestellt. Die Kurseinheiten sind identisch aufgebaut (Abb. 1) und werden tutoriell und technisch begleitet. Formative Tests sind direkt in den Kurseinheiten implementiert und setzen am Mastery Learning an. Die Studierenden erhalten sofort Rückmeldung zu ihrem Lernstand. Zusätzlich werden sie durch fachlich betreute Foren, ein ausgeprägtes Feedback und ein Rückkopplungssystem unterstützt. Zum Lösen der Programmieraufgabe steht eine webbasierte Arbeitsumgebung (YAPEX)⁹ zur Verfügung. Abgeschlossen wird jede Kurseinheit mit einem Wissenstest, fortschreitend mit einem Kompetenztest. Nach jeder Kurseinheit ist eine Umfrage implementiert, die der Evaluation dient und zudem Aspekte zur Metakognition enthält, um bei den Studierenden eine aktive Überwachung, Bewertung und

⁶ Passung von Lernzielen, Inhalten und Prüfungsaufgaben (constructive Alignment) [Bi96]

⁷ ILIAS - Integriertes Lern-, Informations- und Arbeitskooperations-System ist ein Open-Source Learning Management System. [<https://www.ilias.de/>]

⁸ Demo-Kurs abrufbar unter: <https://bit.ly/2UJfSud>. (Nutzer: ICMMDemo, Passwort: ICMM_ilias2020)

⁹ YAPEX – (Yet Another Practical EXercise Platform) ist eine von der Uni Halle am Institut für Informatik entwickelte webbasierte Arbeitsumgebung zum Bearbeiten von Programmieraufgaben. [Dä16]

Regulation des eigenen Lernprozesses anzuregen. Eine elektronische Prüfung lässt abschließend zu, aus Lernerfolgen Rückschlüsse auf erworbene Kompetenzen zu ziehen.

Komponenten / inhaltliche Elemente Online/Präsenz	Umsetzung Elemente Cognitive Apprenticeship	Lernziele Taxonomie
Online LMS ILIAS (1.-15. Kurseinheit) didaktisch und unterschiedlich medial aufbereitete Inhalte (Videoinput, Text, Bild), Codebeispiele, eingebettete Übungen etc. mit Kohärenzhilfen zum selbstregulierten Lernen, Integration von Feedback- und Kommunikationstools	Content Vermittlung von Fakten, Konzepten und prozeduralem Wissen Situated Learning Lernende arbeiten mit realistischen und authentischen Problemen	Wissen aneignen verstehen abrufen
Präsenz (1.-15. Kurseinheit) kleinere Übungsaufgaben zur Förderung verstehenden Lernens, Möglichkeit zum intensiven Diskurs mit Mitstudierenden, dem Lehrenden und erfahrenen Studierenden	Modeling and Explaining Lehrender demonstriert eine Problemlösung und kommentiert die Strategie Scaffolding Lehrender stellt eine Orientierungsgrundlage bereit, Studierende lösen ein ähnliches Problem	Wissen aneignen verstehen abrufen anwenden
Online Programmieraufgabe LMS ILIAS u. YAPEX (1.-11. Kurseinheit) komplexe Anwendungsaufgabe zum aktuellen Kompetenzstand, Bearbeitung in einer webbasierten Entwicklungsumgebung (YAPEX) mit sofortiger Rückmeldung	Fading Lehrender zieht sich bei steigender Kompetenz der Lernenden zurück Coaching Problemlösung der Lernenden, Lernende erhalten Hinweise und Feedback vom Lehrenden	Wissen verstehen abrufen anwenden analysieren
Online Praxisprojekt LMS ILIAS u. YAPEX (12.-15. Kurseinheit) Entwickeln einer Software im realen Kontext (Programmieren kleiner lauffähiger Programme und Vorstellen des Gesamtprojektes)	Exploration Problemlösungskompetenz wird auf andere Bereiche übertragen bzw. Variationen vom Lernenden ausprobiert Articulation Lernende müssen ihre Strategien kommentieren und Entscheidungen begründen	Wissen abrufen anwenden analysieren synthetisieren reflektieren
Online Formative Tests LMS ILIAS u. YAPEX (1.-15. Kurseinheit) Selbsttest zur Lernstandsüberprüfung mit sofortigem Feedback Online Evaluation LMS ILIAS (1.-15. Kurseinheit) Umfrage (geschlossene und offene Fragen)	Reflection Lernende haben Einsicht in ihren Entwicklungsfortschritt und können ihn analysieren und vergleichen Lernende reflektieren den Lernprozess und treffen ggf. Maßnahmen zur Selbstregulation	Wissen abrufen anwenden analysieren reflektieren

Abb. 1: Umsetzung des Moduls OOP im ICMM – Elemente CAM

4 Ergebnisse der Testphase des ICMM

Die Veranstaltung im ICMM wurde erstmalig im Wintersemester 2019/20 durchgeführt und evaluiert. Die Studierenden konnten selbst wählen, ob sie an der klassischen Veranstaltung teilnehmen (217 Studierende) oder das ICMM-Format (20 Studierende) wählen. Im Semesterverlauf wurden über Beobachtungen, anonyme Online-Umfragen je Kurseinheit und eine abschließende Evaluation unterschiedliche lernrelevante Aspekte untersucht. Nachfolgende Betrachtungen richten sich vor allem auf die Motivation der Studierenden und den Lernerfolg (Zuwachs an Kompetenzen) über den Kursverlauf. Kein Studierender hat zur klassischen Lehrveranstaltung gewechselt. Zwei Studierende (OOP als wahlobligatorisches Modul) brachen den Kurs nach Erreichen des ersten Kompetenzlevels ab. Alle Online-Kurseinheiten wurden intensiv bearbeitet (gemessen über Lernfortschrittskontrolle im ILIAS, anonymisierte Zeit- und Seitenerfassung, sowie über die Ergebnisse aus formativen Assessments). Gruppenarbeiten zeigten eine hohe Aktivität in Form der Beteiligung aller Studierenden am Endergebnis. Die Mehrzahl der Studierenden war an einer partizipativen Mitgestaltung und Weiterentwicklung der Lernumge-

bung interessiert. Dies äußerte sich darin, dass die Studierenden freiwillig und eigenständig ergänzende Testfälle für YAPEX-Aufgaben erstellten und die Kursgestaltung durch Erweiterung der Online-Kurseinheiten mit praxisnahen Beispielen bereicherten. Zur Bewertung des Kompetenzzuwachses wurden die Ergebnisse der Online-Tests und Online-Programmieraufgaben herangezogen. Eine quantitative Analyse ergab, dass 39% der Studierenden den klassischen Kurs abgebrochen haben, aber nur 10% der Studierenden den Kurs im ICMM. Die elektronische Abschlussprüfung misst inwieweit die Studierenden objektorientierte Systeme modellieren und Algorithmen in eine Programmiersprache umsetzen, sowie Testfälle formulieren können. Abb. 2 zeigt im Vergleich den Lernerfolg für den Kurs im ICMM und den klassischen Kurs für Klausur 1. Es meldeten sich anteilig mehr Studierende aus dem Kurs im ICMM zur ersten Klausur an. Alle Studierenden haben die Klausur bestanden, die schlechteste Note war 3,3 (ein Studierender). Aus dem klassischen OOP-Kurs haben dagegen 10,68% der Studierenden die erste Klausur nicht bestanden, sieben Studierende erhielten zudem die Note 3,7 und schlechter.

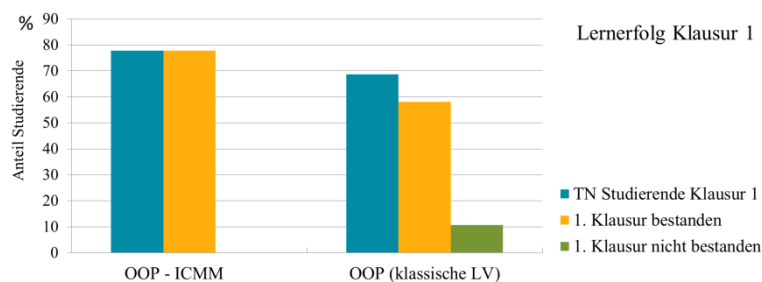


Abb. 2: Klausurteilnahme und summativer Lernerfolg (Klausur 1)

5 Fazit

Die Ergebnisse des Einsatzes des ICMMs und praktische Erfahrungen bei der Nutzung der Lernumgebung OOP zeigen, dass Studierende die Kursinhalte mit hoher Motivation ergebnisorientiert bearbeiteten. Positiv hervorzuheben sind die geringe Abbruchquote und der höhere Lernerfolg im Vergleich zur klassischen Veranstaltung. Der Lernerfolg, formativ und summativ, lässt zudem die Schlussfolgerung zu, dass die Cognitive Apprenticeship-Methode für den Erwerb der Kompetenz der Programmierfähigkeit einen Mehrwert zu bieten scheint. Die didaktische und mediale Gestaltung der Lernumgebung des ICMM wird zusätzlich durch die Einstellungsakzeptanz der Studierenden unterstützt. Lehrende müssen jedoch bei der Umsetzung des CAMs eine sehr viel höhere und auch konkretere Intensität an instruktionaler Unterstützung anbieten. Nachhaltig ist die Wiederverwendung und unkomplizierte Modifizierung sowie Ergänzung des Materials. Aufgrund der positiven Bilanz der Testphase ist vorgesehen, die Veranstaltung weiter auszubauen und für alle Studierenden anzubieten. Hierbei sind eine ATI-Studie und eine statistische Auswertung basierend auf dem unterschiedlichen Vorwissensniveau der Studierenden hinsichtlich der Wirksamkeit gegen Abbruch und zum Lernerfolg geplant.

Literaturverzeichnis

- [AK01] Anderson, L. W., Krathwohl, D. R.: A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Boston: Pearson Education Group, 2001.
- [Bi96] Biggs, J.: Enhancing teaching through constructive alignment, in: Higher Education 32, 347–364, 1996.
- [CBN89] Collins, A., Brown, J. S., Newman, S. E.: Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics. In L.B. Resnick (Hrsg.), Knowing, learning and instruction: Essays in honor of Robert Glaser. Hillsdale, NJ: Erlbaum, S. 453-494, 1989.
- [Dä16] Dähne, J.: YAPEX, 2016. <https://yapex.informatik.uni-halle.de/>, Stand: 28.03.2020.
- [Fä07] Fässler, L. E.: Das 4-Schritte-Modell – Grundlage für ein kompetenzorientiertes E-Learning (Dissertation Nr. 17521), ETH Zürich, 2007.
- [Fr02] Fries, S.: Wollen und Können: Ein Training zur gleichzeitigen Förderung des Leistungsmotivs und des induktiven Denkens, Münster, Waxmann 2002.
- [Gl05] Gläser-Zikuda, M. et.al.: Promoting students' emotions and achievement – Instructional design and evaluation of the ECOLE-approach. Learning and Instruction, Volume 15, Issue 5, S. 481-495, 2005.
- [Ha15] Handke, J.: Digitalisierung der Hochschullehre - Welche Rolle spielt das Inverted Classroom Model dabei?, In Freisleben-Teutscher C. F. / Gruber W. / Haag J. / Weißböck J. (Hrsg.) Neue Technologien – Kollaboration – Personalisierung, Tagungsband zum 3. Tag der Lehre, 2015.
- [KK87] Keller, J. M., Kopp, T. W.: An application of the ARCS model of motivational design. In Reigeluth, C. M. (Hg.): Instructional theories in action. Lessons illustrating selected theories and models, Hillsdale, NJ: Erlbaum, S. 289–320, 1987.
- [Re15] Reinmann, G.: Studententext Didaktisches Design. Hamburg, 2015. Abruf unter: https://gabi-reinmann.de/wp-content/uploads/2013/05/Studententext_DD_Sept2015.pdf, Stand: 28.03.2020.
- [WWD06] Weicker, N., Draskoczy, B., Weicker, K.: Fachintegrierte Vermittlung von Schlüsselkompetenzen der Informatik. In: Forbrig, P., Siegel, G. & Schneider, M. (Hrsg.), HDI 2006: Hochschuldidaktik der Informatik – Organisation, Curricula, Erfahrungen. Bonn: Gesellschaft für Informatik e. V., S. 51-62, 2006.
- [ZS01] Zimmerman, B. J. & Schunk, D. H.: Self-regulated learning and academic achievement: Theoretical perspectives. Routledge, 2001.
- [Zu16] Zukunft, O.: Empfehlungen für Bachelor- und Masterprogramme im Studienfach Informatik an Hochschulen, Bonn: Gesellschaft für Informatik e.V., 2016.