

Reflexionsberichte von ZHL-Lehrwerkstatt Teilnehmenden

aus dem Wintersemester 2022/2023



UNIVERSITÄT
BAYREUTH



ZENTRUM FÜR
HOCHSCHULLEHRE

Zentrum für Hochschullehre

Die Reflexionsberichte von Lehrwerkstatt Teilnehmenden werden vom Zentrum für Hochschullehre an der Universität Bayreuth veröffentlicht.

Das Zentrum für Hochschullehre bietet hochschuldidaktische Serviceleistungen für alle Lehrenden an. Die Aktivitäten und Angebote umfassen neben den klassischen Fortbildungsangeboten, Projekte zur Weiterentwicklung von Lehre mit Lehrstühlen, Lehrberatungen und Coachings von Einrichtungen und Einzelpersonen sowie Austauschformate im weiten Feld der Hochschullehre. Durch den Erwerb von Zertifikaten können alle Lehrenden ihr Engagement und ihre Kompetenzen im Bereich der Lehre sichtbar machen. Somit versteht sich das ZHL als Servicestelle, die sich um jegliche Lehrthemen an der Universität Bayreuth kümmert. Alle Aktivitäten drehen sich ums Entwickeln, Unterstützen und Austauschen von Lehre nach den Wünschen und Anforderungen der Lehrenden.

„Praktisches Programmieren lehren“

Ein Reflexionsbericht im Rahmen
der Veranstaltung ZHLxChange aus
dem Wintersemester 2022/23

Der nachfolgende Reflexionsbericht im Rahmen der Veranstaltung ZHLxChange wird vom Zentrum für Hochschullehre an der Universität Bayreuth veröffentlicht. **Autor des Reflexionsberichts ist Nico Höllerich, wissenschaftlicher Mitarbeiter am Lehrstuhl Angewandte Informatik 3 (Robotik und eingebettete Systeme).**

Über die Veranstaltung ZHLxChange

Seit 2022 bietet das ZHL der Universität Bayreuth Lehrwerkstätten im Rahmen des Projekts QUADIS an. QUADIS steht für "Qualität digital gestützter Lehre an bayerischen Hochschulen steigern." Das Projekt stärkt folglich den bayernweiten Fortbildungsraum für Lehrende an Universitäten und Hochschulen für Angewandte Wissenschaften. Es befördert den Wissenstransfer, Methodenaustausch sowie Synergien unter den Standorten. Dabei werden an den Standorten Schwerpunktbildungen in der Hochschullehre sowie die Evaluation und Beforschung der digitalen Lehrpraxis ermöglicht. Ziel ist es, die hochschuldidaktische Weiterbildung bayernweit auf einem hohen Qualitätsniveau zu digitalisieren und zu flexibilisieren. Für Dozierende wird im Rahmen des Projektes ein umfassendes Fort- und Weiterbildungsangebot zu digital gestützter Lehre in Form von Blended-Learning-Formaten erstellt. Dabei entstehende Lehr-/Lernmaterialien werden allen Weiterbildungsinteressierten als Open Educational Resources (OER) zugänglich gemacht. Weitere Aufgaben sind, die bisherigen Austausch- und Diskursformate weiterzuentwickeln sowie den Transfer von Lehrinnovationen in die Praxis zu unterstützen. Das Verbundprojekt QUADIS wird von der [Stiftung Innovation in der Hochschullehre](#) in der Linie "[Hochschullehre durch Digitalisierung stärken. Präsenzlehre, Blended Learning und Online-Lehre innovativ weiterdenken, erproben und strukturell verankern.](#)" seit dem 01. August 2021 gefördert. Die Projektlaufzeit endet zum 31. Juli 2024.

ZHLxChange wird experimentell seit 2022 am Standort Bayreuth angeboten. Die Veranstaltung richtet sich an Dozierende, die gerne „etwas Neues“ in Ihrer Lehrveranstaltung ausprobieren möchten, z.B. eine innovative Methode, um Diskussionen zu fördern oder die Einführung eines neuen didaktischen Konzepts oder die vor einer Lehr-Herausforderung stehen und Unterstützung bei der Lösungsfindung benötigen. ZHLxChange hilft dabei, die Lehridee weiter auszubauen und von einer kollegialen Beratung zu profitieren. Angelehnt an einer Kollegialen Beratung „Plus“ gliedert sich das Vorgehen in sechs Phasen:

1. Zunächst stellen die Teilnehmenden ihre Lehridee/Herausforderung anderen Kolleg:innen vor und erhalten sowohl von ihnen als auch von dem ZHL Team als Didaktikexpert:innen Feedback dazu.
2. Anschließend werden sie dabei begleitet, ihre überarbeitete Lehridee von den Studierenden im Rahmen der Lehrveranstaltung vorab zu besprechen, bevor diese tatsächlich umgesetzt wird.
3. Im dritten Schritt wird die Idee in der eigenen Lehrveranstaltung umgesetzt. Hierbei ist eine kollegiale Hospitation auf Wunsch möglich, d.h. die anderen Seminarteilnehmenden kommen mit in die Lehrveranstaltung, beobachten das Geschehen und schildern danach ihre Beobachtungen.
4. Die ZHLxChange Teilnehmenden holen Feedback der Studierenden ein.

5. Die Teilnehmenden reflektieren gemeinsam mit den Didaktikexpert:innen und Kolleg:innen über die gemachten Erfahrungen zur Realisierung der Lehridee. Daraus entwickeln Sie neue Impulse für die Lehre.
6. Optional schreiben die Teilnehmenden am Ende einen Reflexionsbericht über ihr Projekt.

Der Mehrwert dieses Veranstaltungstyps ist, dass die Dozierenden bei der Entwicklung neuer Ideen für ihre Lehre nicht alleine gelassen werden, sondern von drei zusätzlichen Perspektiven (Kolleg:innen, Didaktikexpert:innen, Studierende) profitieren können. So können sie das neue Vorgehen auf den Prüfstand stellen, sodass sie mit einem sichereren Gefühl in die eigene Veranstaltung gehen können.

Reflexionsbericht „Praktisches Programmieren lehren“ von Nico Höllerich

Aufbau und Lernziele der Veranstaltung

Programmieren zu lernen stellt für einen Teil der Studierenden der Informatik eine große Herausforderung dar. Ein anderer Teil hat bereits umfassend Erfahrung durch Programmieren in der Freizeit gesammelt. In diesem Spannungsfeld bewege ich mich als Betreuer für das Bachelorpraktikum der Informatik. Ziel des Praktikums ist es Studierenden weiterführende Kenntnisse beim Programmieren und Modellieren von Programmen zu vermitteln, indem sie eine eigene, kleine Software implementieren. Das Praktikum ist eine Pflichtveranstaltung des Studiengangs. Es baut einerseits auf den Einführungsvorlesungen zum Programmieren auf, welche sich mit grundlegenden Strukturen beschäftigen und im Rahmen der Übungen kleine Methoden implementieren lassen. Andererseits dient es als Vorbereitung auf ein Projekt, in dem Studierende selbstorganisiert in Gruppenarbeit ein größeres Programm erstellen sollen. Es kann und soll nicht die Rolle eines reinen Einführungskurses in das Programmieren erfüllen, in dem Schritt für Schritt die Bausteine der Programmiersprache erklärt werden. Als Betreuer für das Praktikum stellt mich das vor die Herausforderung, mit den sehr individuellen Vorkenntnissen und Betreuungsanforderungen umzugehen.



Abbildung 1: Skizze des zu implementierenden Spiels. Per Textfolge muss ein Roboter so programmiert werden (rechte Spalte), dass er ein vorgegebenes Ziel in der Spielwelt erreicht.

Von den vorherigen Betreuenden des Praktikums habe ich die Aufgabenstellungen übernommen. Die Aufgabenstellung umfasst die Implementierung eines kleinen Spiels mit graphischer Oberfläche in der Programmiersprache C++ (s. Abbildung 1). Die Studierenden müssen die Software von der Benutzeroberfläche und dem Anzeigen der Spielwelt über die Steuerung des Roboters bis hin zum Verhalten der Gegner selbst schreiben. Die Aufgabenstellung unterteilt die Software hierfür in sieben Arbeitspakete mit

Anforderungen und Hinweisen zur Implementierung. Zu jedem der Arbeitspakete müssen die Studierenden ein Testat ablegen. Im Testat zeigen sie, dass alle Funktionen umgesetzt sind und die Software fehlerfrei läuft. Den größeren Teil des Testats nimmt die Durchsicht des Codes ein. Dabei gebe ich Feedback zur Struktur des Codes, der Verständlichkeit und Effizienz der Algorithmen. Das bildet einen wichtigen Baustein, damit Studierende besser und schneller programmieren. Außerdem kann ich Studierende so frühzeitig auf die Auswirkungen ihrer Strukturierung hinweisen. Die Testate werden nach einem objektiven Bewertungsschema bewertet. Die Gesamtpunkte aus den Testaten ergeben die Endnote der Veranstaltung. Die Anforderungen stehen am Anfang der Veranstaltung zur Verfügung, sodass sich die Studierenden darauf vorbereiten können.

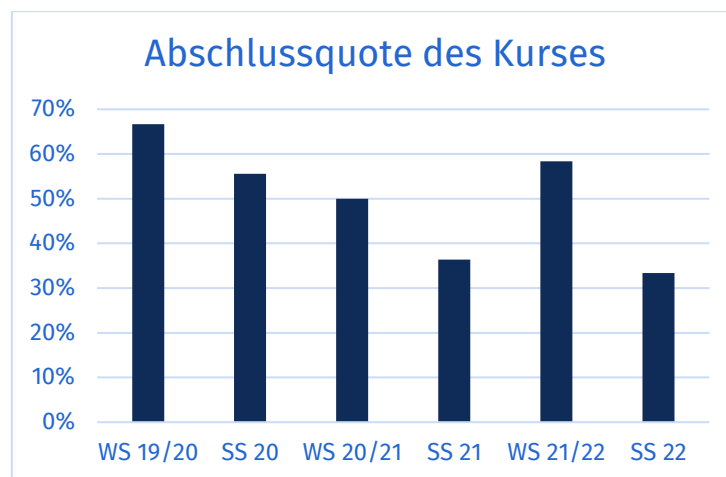


Abbildung 2: Alle Studierende werden zu Beginn des Kurses zur Prüfung angemeldet und können sich nicht selbstständig abmelden. Die Grafik zeigt den Prozentsatz der Teilnehmenden, der die Prüfung bestanden, also alle Testate abgelegt haben.

Unterstützung der Studierenden

Gegenseitige Hilfestellung bei Kompilierfehlern und beim Einarbeiten in die Sprache stellte einen wichtigen informellen Teil des Kompetenzerwerbs dar. Vor der Corona-Pandemie geschah dies vor allem durch Begegnungen und Gruppenarbeiten in den CIP-Pools. Mit der Corona-Pandemie ließ die Vernetzung der Studierenden merklich nach. Nicht nur fiel die gemeinsame Arbeit in den CIP-Pools weg, sondern auch die Bildung von Arbeitsgruppen und das Kennenlernen älterer Studierender wurde schwieriger. Für mich als Betreuer äußerte sich dies vor allem in den Abschlusszahlen, welche – trotz gleichbleibender Schwierigkeit der Aufgabenstellung – kontinuierlich sanken (s. Abbildung 2). Um die Einstiegshürden des Praktikums in Erfahrung zu bringen und die Betreuung zu verbessern, führe ich am Ende des letzten Testats ein strukturiertes Interview mit jedem und jeder Teilnehmenden. Die Studierenden können so rückblickend einschätzen, aus welchen Betreuungsmodalitäten sie den meisten Nutzen gezogen haben und Anregungen für Verbesserungen vorschlagen. Aus dem Interview am Ende sowie den Testaten haben sich zwei Kernprobleme herauskristallisiert: Das Fehlen von Einstiegshilfen sowie die fehlende Praxis beim Strukturieren von Programmen. Für beides

habe ich mit Unterstützungsangeboten begonnen, welche ich im Laufe der Semester weiterentwickelt habe.

Obwohl im Netz einige Tutorials zu C++ zu finden sind, ergibt sich das Problem, dass die Sprache einen großen Wandel durchlebt hat. Viele Dinge lassen sich eleganter lösen als es ältere Tutorials erklären. Dies veranlasste mich einen zweimal zweistündigen Crashkurs in C++ zu geben. Hauptaugenmerk lag dabei auf den Unterschieden zu Java – der Programmiersprache, die die Studierenden in den Einführungsveranstaltungen kennen gelernt hatten. Auch hier zeigten sich wieder die unterschiedlichen Anforderungen der Studierende. Während es für manche – wie von mir beabsichtigt – ein Update ihrer Programmierkenntnisse auf die neue Sprache war, taten sich viele schwer dem Crashkurs in Echtzeit zu folgen. Dies veranlasste mich, den Crashkurs in ein reines Online-Angebot zu überführen. Die Inhalte sind in einzelne Kapitel unterteilt, welche im eigenen Tempo durchgearbeitet und wiederholt werden können. Am Ende des Tutorials steht ein Demo-Projekt, welches alle essentiellen Bausteine enthält und zum Download zur Verfügung steht. Das Angebot wird durch weitere Links und externe Tutorial-Videos ergänzt, welche vertieftes Wissen zu einzelnen Aspekten vermitteln. Viele Studierende bestätigen rückblickend, dass das Online-Angebot hilfreich war. Weisen manchmal aber darauf hin, dass bei Problemen die Möglichkeit zur direkten Rückfrage fehlt.

Die zweite Hürde stellt die Strukturierung des Programms dar: Wie können die Anforderungen jeder Aufgabe in sinnvolle logische Einheiten zerlegt werden, die sich gut wiederverwenden und warten lassen? Gerade hier ist es notwendig frühzeitig steuernd einzugreifen. Andernfalls wird die Bearbeitung nachfolgender Aufgaben erheblich komplizierter oder aufwändiger, weil größere Teile des vorhandenen Programms angepasst werden müssen. Hauptinstrument sind hierfür die Testate, bei denen Tipps und Hilfestellungen ein wichtiger Baustein sind. Von mir als Betreuer verlangt das Erfahrung, mich schnell in den Code einzufinden, die Design-Absicht zu verstehen und abzuschätzen, inwiefern diese kompatibel mit den Anforderungen zukünftiger Aufgaben sind. Gerade Studierende mit wenig Programmiererfahrung neigen dazu, ungünstigere Strukturierungen zu wählen und sich so die Aufgaben noch schwerer zu machen.

Einerseits sind die Hilfestellungen hochgradig individuell (an welchen Stellen kann der Programm-Code verbessert werden?) – andererseits ist es zielführend und effizient gewisse Klassen und Methodensignaturen zu verwenden. Die Klassenstruktur im Hinblick auf die anstehenden Aufgaben zu planen stellt ein wichtiges Lernziel des Praktikums dar. Um die Studierenden gleichermaßen zu unterstützen und zum Austausch anzuregen, habe ich Gruppentermine eingeführt. Der erste Gruppentermin (Einstiegs-Gruppentermin) dient der Vorstellung der Aufgabenstellung mit Fokus darauf, welche Aspekte für mich bei der Bewertung wichtig sind und wie die Aufgaben zusammenspielen. In zwei weiteren Gruppenterminen (Strukturierungs-Gruppentermin) soll eine potentielle Klassenstruktur und Schnittstellendefinition für das gesamte Projekt erarbeitet werden. Hierfür habe ich die Aufgaben weiter in Teil-Anforderungen zerlegt, sodass es insgesamt so viele Anforderungen wie Teilnehmende am Kurs gibt. Jedem und jeder wird eine Teilanforderung zugewiesen. Die Studierenden überlegen sich in der ersten Hälfte des Termins, welche Klassen und Methodensignaturen notwendig sind. In der zweiten Hälfte

werden die Ergebnisse in der Gruppe zusammengetragen. Es entsteht ein gemeinsamer Vorschlag, welche von den Teilnehmenden als Richtschnur verwendet werden kann und nach der Sitzung im E-Learning zur Verfügung steht. Rückblickend wird das Ergebnis von vielen als hilfreich bewertet. Allerdings zeigt sich, dass sich manche Studierende schwertun: Die Aufgabenstellung wird nicht verstanden, es wird zu sehr oder zu wenig ins Detail gegangen oder die Teilbausteine passen am Ende nicht zusammen. Für mich als Betreuer bedeutet dies, häufig steuernd eingreifen zu müssen und das Ergebnis nachzujustieren, damit sich die Studierenden sinnvoll daran orientieren können. Andererseits ist es schwierig die Veranstaltung in einem Zeitrahmen zu halten, bei dem einerseits die Konzentration gewährleistet und andererseits den Studierenden genügend Zeit gegeben wird, sich einzuarbeiten.

Trotz dieser Bemühungen ist die Abschlussquote meines Kurses weiter gesunken und es fiel mir schwer, gerade die Studierenden mit wenig Programmiererfahrung gezielt zu fördern und über die Hürde der ersten Aufgaben zu bringen. Als neues didaktisches Konzept sollte die Bearbeitung einer Präsenzaufgabe im Rahmen des Einstiegs-Gruppentermins dienen. Im Gegensatz zum Onlinematerial stehe ich als Betreuer für Rückfragen direkt zur Verfügung und kann so helfen, wenn Dinge nicht funktionieren.

Weiterentwicklung des Lehrkonzepts

Während der Entwicklung des Konzepts nahm ich an der Lehrwerkstatt des Zentrums für Hochschullehre (ZHL) teil, um das Lehrkonzept des Kurses mit Fokus auf der Präsenzaufgabe zu verfeinern. Die Lehrwerkstatt bestand aus drei Veranstaltungen: Vorstellung des (Kurs-)Konzepts und Feedback unter den Teilnehmenden der Lehrwerkstatt, Hospitation des Einstiegs-Gruppentermins durch Mitarbeitende des ZHL sowie einer Nachbesprechung mit Reflexionen und weiteren Anregungen. Aus der ersten Veranstaltung ergaben sich vier Vorschläge: Lernziele mehr betonen, direktere Kommunikationswege anbieten, unbewertete Übungsmöglichkeiten bieten und Schritt-für-Schritt-Anleitungen anbieten. Die ersten beiden setzte ich im Rahmen der Besprechung der Aufgabenstellung sowie durch die Forum-Aktivität im E-Learning um. Während des Semesters richteten die Studierenden noch einen privaten Discord-Server ein.

Rückblickend betrachtet, stellte sich auf keiner der beiden Plattformen der erhoffte rege Austausch ein. Da die E-Learning-Aktivität aber kaum Aufwand mit sich bringt und von einigen Studierenden doch genutzt wurde, werde ich es für kommende Semester beibehalten. Ähnliches gilt für die unbewertete Übungsmöglichkeit. Ich bot den Studierenden die Möglichkeit, die Präsenzaufgabe als unbewertetes Testat abzulegen. Sie hätten so erfahren können, welche Kritik sie erwartet, ohne dass es Einfluss auf die Note nimmt. Von dem Angebot hat allerdings niemand Gebrauch gemacht. Sehr positiv verlief dagegen die Bearbeitung der Präsenzaufgabe, was auch von Seiten des ZHL-Mitarbeiters bestätigt wurde, der der Veranstaltung beiwohnte. Studierende hatten beim Einrichten des Projekts zum Teil Probleme mit denen ich vorher nicht gerechnet hatte. Ein typischer Fehler war z.B., dass die bereitgestellten Dateien an beliebigen Orten abgelegt wurden.

Beim Kompilieren des Programms konnten dann referenzierte Dateien nicht gefunden werden. Derartige Probleme konnte ich direkt beheben und auf die zu Grunde liegende Funktionsweise eingehen. Wie Abbildung 3 zeigt, schätzten auch die Studierenden die Gruppenarbeit als sehr hilfreich ein.

Wie hilfreich war ...



Abbildung 3: Feedback der Studierenden nach dem Einstiegs-Gruppentermin mit Präsenzaufgabe

Nicht ganz so erfolgreich erwiesen sich die Strukturierungs-Gruppentermine (s. Abbildung 4). Studierende fehlte oftmals eine Idee, wie der Teilaspekt angegangen werden kann. Entsprechend war es für sie nicht möglich, über benötigte Methoden nachzudenken. Weitere Schwierigkeiten zeigten sich beim Verständnis, was Ziel der Aufgabe ist, und beim Zusammentragen der Lösungen. Trotz massiver Verlängerung der Bearbeitungszeit ergab sich keine Lösung, mit der die Studierenden sinnvoll weiterarbeiten konnten. Zu viel Code wurde von den Gruppen nebeneinanderher entwickelt (da z.B. als Abhängigkeit benötigt) ohne zusammenzupassen. Ebenfalls hinderlich waren die technischen Probleme mit Etherpad Lite, der E-Learning-Aktivität zum gemeinsamen Arbeiten an Textdokumenten. Bei gleichzeitiger Nutzung sprang der Cursor oder Teile wurden ungewollt überschrieben. Aus diesem Grund bot ich den Studierenden für den zweiten Strukturierungs-Gruppentermin an, die alte Gruppenlösung zu überarbeiten und zusammenzuführen statt die Anforderungen weiterer Aufgaben zu integrieren. Die Studierende überarbeiteten anschließend in einer neuen Gruppenzusammensetzung die alte Lösung, sodass alle Teile zusammenpassen und eine verwendbare Klassenstruktur mit einigen Schnittstellen-Methoden bilden.

Wie hilfreich war ...



Abbildung 4: Feedback der Studierenden nach dem ersten Strukturierungs-Gruppentermin

Die Rückmeldung der Teilnehmenden nach dem zweiten Strukturierungs-Gruppentermin zeigt, dass sie sich detailliertere Aufgabenbeschreibung, Spezifikationen, Feedback zum Verständnis und mehr Zeit wünschen.

Aufbauend auf dem Feedback der Studierenden und den Anregungen aus der Nachbesprechung der Lehrwerkstatt habe ich zwei Maßnahmen ergriffen: Regelmäßige Treffen der Studierenden und Umstellung des Gruppentermins zur Projektstruktur. Mit der ersten Maßnahme habe ich etwa zur Hälfte des Semesters begonnen. Die Studierenden trafen sich mit mir an der Uni und waren aufgefordert, einen tragbaren Rechner mit funktionierender Entwicklungsumgebung und ihrem Projektstand mitzubringen. Zu Beginn der Veranstaltung habe ich sie in Gruppen aufgeteilt, sodass weiter fortgeschrittene Studierende den anderen helfen konnten. Die Veranstaltung zeigte auch, dass sich einige Fragen angesammelt hatten, die trotz der angebotenen Kanäle nicht gestellt wurden. Insgesamt bewerteten die Studierenden das Angebot als positiv und wünschten sich eine Fortsetzung im Zwei-Wochen-Rhythmus, was ich übernahm. Aus den Interviews mit den Studierenden am Ende der Veranstaltung ergibt sich ein zweigeteiltes Bild: Leistungsstarke Studierende profitierten von neuen Anregungen durch gegenseitigen Austausch, bewerteten dagegen den Projektstruktur-Gruppentermin als weniger hilfreich.

Fazit und Ausblick

Zwar bestand auch bisher die Möglichkeit statt eines Testats eine Fragestunde abzuhalten, um Probleme bei Strukturierung oder Implementierung mit mir zu besprechen. Diese wurde leider nur in wenigen Fällen wahrgenommen. Alternative, elektronische Kommunikationswege scheinen hier nur einen unzureichenden Ersatz zu bieten. Ähnliches gilt für Online-Material (Tutorials, Einführungen, etc.). Zwar bietet es eine gute Möglichkeit, tiefer in die Materie einzusteigen. Allerdings kann es gerade für Einsteiger Probleme bereiten: die eigene Entwicklungsumgebung stimmt nicht exakt mit der des Tutorials überein, das Programm verhält sich anders oder die Studierenden sind überfordert. Direktes Eingreifen oder Anpassen des Lehrinhalts ist in dem Fall nicht möglich, was zu Frustration, Demotivation und Aufschieben der Aufgaben führt bei einigen führt.

Abhilfe für die Nachteile soll ein verpflichtendes, wöchentliches Treffen in der Gruppe bieten. Studierende sollen im Rahmen dieser Treffen sich gegenseitig unterstützen und Probleme und offene Fragen direkt ansprechen können. Die Besprechung der Projektstruktur wird in diesen Rahmen eingebettet. Der Einführungs-Gruppentermin soll ebenfalls in gleicher Form beibehalten werden. Dagegen soll der Strukturierungs-Gruppentermin überarbeitet werden. Wichtigste Änderungen sollen sein, die Studierenden in größeren Gruppen an der Gesamtstruktur arbeiten zu lassen, sodass die Konsistenz einer Lösung gewährleistet ist. Außerdem werden die Teilfunktionalitäten auf das Wesentliche reduziert und mehr Klassen bzw. Methodensignaturen vorgegeben, sodass klarer wird, worauf es hinauslaufen soll.

Ein neuer experimenteller Baustein soll die Verwendung von ChatGPT darstellen. Das KI-basierte Sprachmodell ermöglicht es, Code-Beispiele zu generieren und diese durch gezielte Anweisungen den eigenen Anforderungen anzupassen. Ferner antwortete es zielgerichtet auf Rückfragen zum Code und zu Fehlermeldungen. Die Studierenden sollen es optional im Rahmen der Präsenzaufgabe, Gruppentermine und zur Bearbeitung der Testate nutzen können. Ziel ist es, das Sprachmodell einerseits praxisnah ausprobieren können, andererseits aber auch die Grenzen im Hinblick auf Fehler und Größe des erzeugten Programms kennen zu lernen. Für mich als Betreuer stellt sich die Frage, ob es Studierenden einen Mehrwert für den Einstieg in die Programmiersprache bietet, wenn sie natürlichsprachliche Anweisungen in Quellcode überführen können.

Eine Abschlussquote von 64% im aktuellen Semester spricht dafür, dass die umgesetzten Konzepte die weiterhin anhaltenden Auswirkungen der Corona-Pandemie adäquat adressieren zu können. Der Einstieg in das eigenständige Designen und Implementieren wird hierdurch auch Studierende mit weniger Erfahrung ermöglicht. Die Impulse und konkreten Vorschläge aus der Lehrwerkstatt, detailliertere Einführungsaufgaben zu geben und den Austausch in Gruppen stärker zu erzwingen, hat hierfür einen wichtigen Beitrag geleistet.