



CodeAbility Austria – Digital gestützte Programmierausbildung an österreichischen Universitäten

**Extended Management Summary
Dezember 2024**

**Universität Innsbruck (UIBK), Johannes Kepler Universität Linz (JKU),
Paris Lodron Universität Salzburg (PLUS), TU Graz, TU Wien,
Universität Klagenfurt (AAU), Universität für Weiterbildung Krems**

Angesichts der stetig wachsenden Bedeutung digitaler Methoden und künstlicher Intelligenz in allen Wissenschaftsdisziplinen ist der **Zugang zu einer qualitativ hochwertigen Programmierausbildung für Studierende aller Studienrichtungen** ein entscheidender Faktor für die Weiterentwicklung von Lehre und Forschung an österreichischen Universitäten. Auf curricularer Ebene gewinnen Programmier- und Datenanalysekompetenzen zunehmend an Bedeutung. Dies spiegelt sich auch in der Entwicklung neuer Bildungsangebote mit interdisziplinärer oder modularer Struktur wider, wie etwa Erweiterungsstudien oder Wahlpakete.

Diese notwendige und zu begrüßende Entwicklung geht jedoch einher mit einem entstehenden Spannungsfeld zwischen einer wachsenden Zahl zu servicerenden Studierenden, einer zu geringen Zahl an Lehrkräften mit entsprechendem fachlichem und didaktischem Know-How, sowie knappen Budgets. Hinzu kommen allgemeine Herausforderungen an die Qualität der Programmierausbildung, um Studierende zu befähigen, Problemstellungen zu verstehen und eigenständig Lösungen zu entwickeln. Von zentraler Bedeutung ist selbständiges Üben und individuelles Feedback des Lehrenden. Dabei ist der Einsatz moderner Lehrmethoden, wie zum Beispiel projektbasiertes Lernen und der Einsatz digitaler Lernplattformen, von Vorteil. Die kontinuierliche Anpassung der Lehrinhalte an aktuelle technologische Entwicklungen trägt ebenfalls zur Sicherung der Ausbildungsqualität bei.

Im Projekt *CodeAbility Austria*¹ haben sieben österreichische Universitäten in fünf Jahren Projektlaufzeit Strukturen, Methoden und digitale Plattformen aufgebaut, die diesen Rahmenbedingungen gerecht werden und dazu geeignet sind, die Programmierausbildung in Österreich in eine digitale Zukunft zu führen, insbesondere auch unter Berücksichtigung der Herausforderungen und Chancen der Entwicklungen im Bereich Künstlicher Intelligenz.

¹ *CodeAbility Austria* wurde im Rahmen der Initiative „Digitale und soziale Transformation in der Hochschulbildung“ vom Ministerium für Bildung, Wissenschaft und Forschung von 2020-2024 gefördert. Partner: Universität Innsbruck (Konsortialleitung), Johannes Kepler Universität Linz, Paris-Lodron-Universität Salzburg, TU Graz, TU Wien, Universität Klagenfurt, Universität für Weiterbildung Krems

Die Ergebnisse von *CodeAbility Austria* sind den folgenden vier Bereichen zuzuordnen:

- Programmierlernplattformen und Learning Analytics
- Digital gestützte didaktische Methoden
- Open Educational Resources (OER) und Lehrendenvernetzung
- Chancen und Herausforderungen Künstlicher Intelligenz und deren Einsatz in der Lehre

Die entwickelten Ergebnisse wurden an mehreren Standorten evaluiert und in die universitäre Lehre integriert, mit dem Ziel die Ausbildung der Studierenden zu verbessern. Ein besonderer Fokus lag hierbei auf Lehrveranstaltungen zur Einführung in die Programmierung, sowohl für Informatik-Studierende als auch für Studierende anderer Fachrichtungen. Angesichts der substanziellen Veränderungen, die damit für Studierende und Lehrende einhergingen, wurden mehrere empirische Fallstudien durchgeführt, deren Ergebnisse in hochrangigen wissenschaftlichen Publikationen veröffentlicht wurden und wertvolle Impulse zur weiteren Verbesserung der Lehrinhalte geliefert haben.

KONTEXT: ERLERNEN EINER PROGRAMMIERSPRACHE

Es ist heute sowohl auf europäischer als auch auf nationaler Ebene akzeptiert, dass Programmierfähigkeiten als essenzieller Bestandteil moderner Allgemeinbildung gelten. Auf universitärer Ebene bilden sie die Basis für die Anwendung digitaler Methoden und künstlicher Intelligenz, die als Treiber für Erkenntnisgewinn in allen Wissenschaftsdisziplinen wirken.

Beim Erlernen einer Programmiersprache stellen Übungen ein zentrales Element dar. Durch das Erstellen eigener Programme werden Studierende schrittweise in die Lage gebracht, für gegebene Problemstellungen selbständig adäquate Abstraktionen zu finden und Lösungen zu entwickeln. Lehrende haben die Aufgabe, die Leistung einer Gruppe von Studierenden insgesamt einzuschätzen und zu bewerten und Studierenden Rückmeldung zu ihren Lösungen zu geben.

Die universitäre Praxis ist dabei typischerweise mit einer Reihe von Herausforderungen konfrontiert:

- Hohe Studierendenzahlen und die zur Verfügung stehenden Kontaktstunden erlauben zu wenig individuelles Feedback.
- Es besteht ein Mangel an erfahrenen Lehrveranstaltungsleiter:innen, die entsprechend qualitativ hochwertiges Feedback an die Studierenden liefern können.
- Die Studierenden haben unterschiedliche Vorbildung (z.B. verschiedene schulische Hintergründe), und dadurch ein heterogenes Vorwissen. Der Bedarf an Rückmeldung ist deswegen sehr unterschiedlich und zum Teil auch sehr hoch.
- Die faire Bewertung und Notengebung von Studierendenleistungen sind sehr herausfordernd, das Problem von Plagiaten und neuerdings die Generierung von Lösungen durch Künstliche Intelligenz (Artificial Intelligence, kurz AI)-Tools ist allgegenwärtig.

Durch innovative didaktische Konzepte und Bereitstellen digitaler Lehr- und Lernplattformen hat das Projekt *CodeAbility Austria* die Voraussetzungen für eine moderne Programmierlehre geschaffen,

- die den Übungsprozess von den Kontaktstunden weitgehend entkoppelt und auf die individuellen Kompetenzen der Studierenden ausgerichtet ist,
- die allen teilnehmenden Studierenden eine transparente, faire und kompetenzorientierte Notengebung garantiert,

- die die Lehrende von Routinetätigkeiten entlastet, ihnen datenbasierte Analysen über das aktuelle Kompetenzlevel der Gruppe zur Verfügung stellt und sie damit in die Lage versetzt, die Kontaktstunden auf die Gruppe ausrichten zu können.

PROGRAMMIERLERNPLATTFORMEN UND LEARNING ANALYTICS

Das Kernstück eines zukunftsfähigen Ansatzes sind **Programmierlernplattformen**. Sie unterstützen den Workflow von Aufgabenerstellung, Einreichung von Lösungen durch die Studierenden bis zur Rückmeldung und (halb-)automatisierten Bewertung durch den Lehrenden, sowie der Planung der Kontaktstunden. Programmierlernplattformen sind aktuell ein überaus dynamisches Gebiet, hinter dem zum einen ein globaler Milliardenmarkt steht, getrieben z.B. durch Assessment Zentren und dem privaten Bildungsmarkt, zum anderen aber auch die Möglichkeiten der Künstlichen Intelligenz, z.B. in Richtung Learning Analytics oder persönlichen „AI-Tutoren“.

Nach einem sorgfältigen Auswahlprozess entschied sich das Konsortium für die Plattform Artemis, die als open-source Projekt von der TU München entwickelt wurde, dort seit 2019 im Einsatz ist, und eine stabile User Community besitzt. Im Rahmen von *CodeAbility Austria* wurde Artemis für den Betrieb an österreichischen Universitäten angepasst, eine Kooperation mit der TU München aufgebaut und die Plattform den österreichischen Projektpartnern seit dem Wintersemester 2021 für den Pilotbetrieb zur Verfügung gestellt.

Artemis bietet eine vielseitige, interaktive Plattform, auf der eine Reihe verschiedener Übungstypen wie Programmier-, Quiz-, Modellierungs- und Textaufgaben mit direktem Feedback eingesetzt werden können. Die Übungen sind flexibel konfigurierbar, und auch die Plattform selbst ist flexibel einsetzbar, indem sie z.B. weitgehend unabhängig von der eingesetzten Programmiersprache ist. Ein wesentlicher Vorteil von Artemis ist die Möglichkeit, die abgegebenen Lösungen von Studierenden automatisiert zu prüfen, z.B. durch Ausführen von Testfällen oder anderen Bewertungskriterien. Dies ermöglicht nicht nur sofortiges und individuelles Feedback für Studierende, sondern reduziert auch den manuellen Korrekturaufwand für Lehrende erheblich. Somit ist auch bei Lehrveranstaltungen mit einer großen Anzahl an Studierenden Skalierbarkeit gegeben, und individuelles und qualitativ hochwertiges Feedback möglich. Weiters in Artemis integriert sind Plagiatschecks, sowie Learning-Analytics-Funktionen, die den Lehrenden umfassende Einblicke über die Leistungsentwicklung der Studierenden geben.

Neben dem „klassischen“ Übungsbetrieb bietet Artemis auch die Möglichkeit Online-Tests bzw. Klausuren durchzuführen. Der Prüfungsmodus erlaubt die Erstellung verschiedener Prüfungsvarianten, sowie zusätzlich zur automatisierten Bewertung einen Doppelblindmodus, welcher die Objektivität der Bewertung erhöht. Über den LTI-Standard kann Artemis in verschiedene Lern-Management-Systeme, wie z.B. Moodle oder OLAT, integriert werden.

Seit der Pilot-Inbetriebnahme von Artemis am Institut für Informatik an der Universität Innsbruck 2021 wurden über 30 Lehrveranstaltungen an vier Standorten (UIBK, PLUS, JKU, AAU) durchgeführt, in den Programmiersprachen C/C++, Python und Java. Darüber hinaus wurde Artemis auch fallweise in weiteren Kursen für einzelne Aufgaben eingesetzt. Da das Vertrauen in die Stabilität des Systems ständig wuchs, wurde es ab WS 22/23 auch vermehrt für Klausuren mit bis zu 180 Teilnehmer:innen eingesetzt. Seit dem produktiven Einsatz wurden mehr als 3600 Studierende mittels Artemis geschult.

Neben Artemis sammelte das Konsortium auch mit anderen Programmierlernplattformen Erfahrungen und arbeitete an der Neuentwicklung eines seit vielen Jahren an der TU Graz verwendeten Tutor-Systems für MATLAB.

Für Lehrende stellt der Wechsel vom bisher üblichen und erprobten Übungsbetrieb zur digital gestützten Programmierausbildung mit einer Übungsplattform einen großen Schritt dar, der sowohl Chancen als auch Herausforderungen mit sich bringt. Zum einen muss der Betrieb der Plattform organisiert werden und eine Finanzierung gefunden werden. Zum anderen müssen bestehende Übungsmaterialien mit teilweise erheblichem Arbeitsaufwand überarbeitet, erweitert und angepasst werden. Weiters muss die gesamte Organisation des Übungsbetriebs umgestellt und an die neuen Gegebenheiten angepasst werden.

Für die Studierenden scheint ein Umstieg auf Übungsplattformen viele Vorteile zu bieten, z.B. für einen schnellen Einstieg in die Programmierung und um sofortiges und individuelles Feedback zu erhalten. Zudem eröffnet die Nutzung von Übungsplattformen Potenziale für personalisierte Lernbegleitung. Um belastbare Aussagen über die Wirkung von Programmierlernplattformen treffen zu können, entschied das Konsortium, eine Reihe von Begleitstudien mit Befragungen von Lehrenden und Studierenden durchzuführen.

Rückmeldung der Lehrenden: Dadurch, dass die Plattform alle Abgaben mit den gleichen Kontrollen und Auswertungen überprüft, ist, verglichen mit herkömmlichen manuellen Beurteilungsmethoden, eine objektivere Leistungsbeurteilung möglich. Die Beobachtung des Lernfortschritts der Studierenden durch erzeugte Auswertungen und Visualisierungen wird von den Lehrenden als sehr hilfreich empfunden. Sie können Wissenslücken von Studierenden in Gruppen frühzeitig erkennen und die Präsenzeinheiten dahingehend anpassen. Die Lehrenden beobachten auf der anderen Seite, dass Studierende ihre Lösungen auf die automatisierten Checks der Plattform anpassen, und sich somit häufig ein Versuch-und-Irrtum-Verhalten einstellt. Der Mehraufwand zur Erstellung automatisiert überprüfbarer Programmierübungen wird häufig als Nachteil empfunden. Viele Lehrende sehen jedoch darin eine sinnvolle Investition, insbesondere bei großen Kursen, da die Tests einmalig erstellt werden und anschließend eine effiziente Bewertung einer großen Anzahl von Studierenden ermöglichen.

Rückmeldung der Studierenden: Die Studierenden schätzen das sofortige Feedback beim Einreichen der Lösungen. Sie können damit fehlerhafte Lösungen auf Basis des Feedbacks korrigieren, noch vor der nächsten Präsenzeinheit mit dem Lehrenden. Sie empfinden die Bewertung der Lösungen außerdem als fair und transparent, insbesondere bei großen Studierendenzahlen mit vielen Übungsgruppen und Lehrenden, wo bei "traditionellen" Beurteilungen typischerweise nicht immer eine konsistente Beurteilung garantiert werden kann.

Eine Übersicht über die Studien, die im Rahmen der Begleitforschung von *CodeAbility Austria* durchgeführt und publiziert wurden, ist am Ende dieses Artikels zu finden.

OPEN EDUCATIONAL RESOURCES (OER) UND LEHRENDENVERNETZUNG

Der signifikante Mehraufwand zur Erstellung von Programmierübungen für die Plattform erhöht ganz wesentlich die Bereitschaft zum Austausch von Lehrmaterialien. Im Rahmen von *CodeAbility Austria* wurden mit Bereitstellung einer Sharing Plattform² die technologischen und organisatorischen Voraussetzungen geschaffen, Lehrmaterialien für die Programmierlehre zwischen österreichischen Universitäten auszutauschen. Die Sharing Plattform unterstützt den gesamten Workflow der Lehrenden von der Ideenfindung bis zum Erstellen einer Aufgabe. Um Materialien aufzufinden, wurde

² <https://search.sharing-codeability.uibk.ac.at>

vom Konsortium ein Kompetenzmodell für die Programmierausbildung entwickelt. Zusätzlich kann in der Sharing Plattform nach Programmiersprachen, Formaten, Autoren und Lizenzen gefiltert werden.

Die Sharing Plattform ist für alle Lehrenden und auch Studierende offen. Ziel war eine Bereitstellung der Inhalte unter einer offenen Lizenz (OER). Im Laufe des Projekts stellte sich allerdings der Wunsch heraus, z.B. Musterlösungen nur unter Lehrenden auszutauschen. Dem wurde Rechnung getragen, indem auch private (bzw. teil-öffentliche) Aufgaben unterstützt werden. Zudem wurde ein Reviewing Prozess integriert, der eine Qualitätssicherung der Aufgaben ermöglicht und dies bei der Trefferpriorität berücksichtigt.

Mittlerweile sind mehr als 350 Aufgaben (34 C-Aufgaben, 200 Java-Aufgaben, 109 Python-Aufgaben und 20 nicht programmierspezifische Aufgaben) in der Sharing-Plattform verfügbar. Um den Austausch unter Lehrenden zu optimieren, wurden Schnittstellen zu Artemis und zum OERHub realisiert.

HERAUSFORDERUNGEN UND CHANCEN VON KÜNSTLICHER INTELLIGENZ

Die zunehmende Verfügbarkeit und einfache Einsetzbarkeit generativer AI-Werkzeuge stellen eine zentrale Herausforderung in der Lehre dar. Dies hat sich insbesondere ab dem Studienjahr 2023 mit der breiten Verfügbarkeit von ChatGPT auf den Ablauf, die Organisation und die Bewertungsmodalitäten von Lehrveranstaltungen durchgeschlagen. Studierende nutzen AI-Tools häufig, um direkt vollständige Lösungen für Programmieraufgaben zu erhalten, ohne dabei die eigentlichen Konzepte zu erlernen. Aktuell verfügbare Versionen der Tools (wie ChatGPT4) sind dabei in der Lage, speziell bei einführenden Lehrveranstaltungen, durchwegs gute bis sehr gute Lösungen zu produzieren. Um diesem fundamentalen Problem entgegenzuwirken, wurde im Rahmen von *CodeAbility Austria* ein AI-basierter Tutor entwickelt, der direkt in die Artemis Plattform integriert ist.

Dies soll dazu beitragen, dass Studierende AI-Tools zwar einsetzen, jedoch nicht direkt komplette Lösungen zu den Aufgabenstellungen erhalten. Stattdessen ist die Idee, dass dieser "Tutor" die Rolle eines menschlichen Lernbegleiters übernimmt, und Studierende Schritt für Schritt zu einer Lösung führt. Der AI-Tutor soll dabei gezielte Gegenfragen stellen und die Studierenden auf zentrale Aspekte des Problems lenken, und dadurch eigenständiges Denken, Reflexion sowie ein tieferes Verständnis der Materie fördern, ohne direkt den Lösungsweg vorwegzunehmen. Dadurch können Vorteile AI-gestützter Tools gezielt genutzt werden, ohne den Lerneffekt zu beeinträchtigen.

Eine Übersicht über die Studien, die im Rahmen der Begleitforschung von *CodeAbility Austria* in diesem Kontext durchgeführt und publiziert wurden, ist im nächsten Abschnitt zu finden.

BEGLEITFORSCHUNG

Die Begleitforschung im Rahmen des *CodeAbility Austria* Projektes hat sich im Wesentlichen auf zwei Hauptbereiche fokussiert. Der erste Teil der Begleitforschung beschäftigte sich mit dem Einsatz von Programmierlernplattformen im Allgemeinen, sowie Anforderungen von Studierenden und Lehrenden an eine solche Plattform. Durch eine Reihe von Studien und Interviews mit Studierenden und Lehrenden konnten kritische Funktionalitäten identifiziert werden, die dazu beigetragen haben, die Artemis-Plattform zu verbessern und zu erweitern und an die Bedürfnisse der Studierenden anzupassen.

Ein besonderes Augenmerk lag besonders in der zweiten Hälfte des Projektes im Einsatz von AI-gestützten Technologien. Von Herbst 2023 bis Frühjahr 2024 wurde dazu eine umfassende Studie mit über 300 Studierenden, an drei Standorten, Linz (JKU), Salzburg (PLUS) und Innsbruck (UIBK), durchgeführt, um den Einsatz unterschiedlicher Feedbackformen in der Programmierlernplattform Artemis zu untersuchen. Speziell wurde dabei "traditionelles" Feedback, wie zum Beispiel automatisierte Testergebnisse, mit AI-gestütztem Feedback verglichen. Für die Einschätzung der unterschiedlichen Feedbackarten durch die Studierenden wurde das Technology Acceptance Model (TAM) herangezogen. Im Rahmen dieser und noch weiterer Studien konnten wir aufzeigen, dass der gezielte Einsatz von AI-Feedback und AI-Tutoring in der Artemis Plattform zu guten, und teilweise besseren Lernergebnissen bei Studierenden führen kann.

Der zweite große Forschungsschwerpunkt lag auf der Erarbeitung und Validierung neuer und moderner didaktischer Konzepte, etwa der Entwicklung von Kompetenzmodellen oder Werkzeugen zur Unterstützung von Learning Analytics Techniken. Ein wichtiger Aspekt war dabei die Analyse ob und inwieweit durch die Sammlung und Auswertung von Studierendendaten Rückschlüsse auf das Lernverhalten bzw. auf Lernschwierigkeiten von Studierenden gezogen werden können. Eine Auswertung von Studierendendaten im Rahmen einer durchgeführten Studie an der JKU hat belegt, dass der Einsatz solcher Methoden und Werkzeuge zu Verbesserungen im Lehralltag beitragen kann. In diesem Kontext wurden Kompetenzmodelle für die Programmiergrundausbildung in Java und Python entwickelt inklusive einer Methode zur Erstellung kompetenzbasierter Übungsaufgaben. Diese Methode unterstützt Lehrende bei der Entwicklung gezielter Programmieraufgaben für ausgewählte Kompetenzen und die entsprechenden Tests.

AUSBLICK: ZUKUNFT DER PROGRAMMIERAUSBILDUNG IN ÖSTERREICH

Im Rahmen des Projekts *CodeAbility Austria* haben sieben österreichische Universitäten in den Jahren 2020 bis 2024 die methodischen, technologischen und organisatorischen Grundlagen geschaffen, um Lehrveranstaltungen zur Einführung in die Programmierung für Studierende aller Disziplinen qualitativ hochwertig, effizient und skalierbar anzubieten.

Die Ergebnisse der begleitenden Forschung zeigen, dass der Einsatz von Übungsplattformen sowohl Lehrenden als auch Studierenden gleichermaßen zugutekommt. Lehrenden wird ermöglicht, die Leistungen von Studierendengruppen objektiver zu beurteilen und ihren Lernfortschritt gezielt zu beobachten. Studierende profitieren von einer höheren Flexibilität hinsichtlich Lernzeit und -ort sowie von zeitnahen, individuellen Rückmeldungen zu eingereichten Lösungen. Diese Rückmeldungen fördern das Verständnis und steigern den Lernerfolg.

Aktuelle Entwicklungen im Bereich Generative AI deuten auf tiefgreifende Veränderungen sowohl in der Programmierung als auch in der Programmierausbildung hin. Es wird in den kommenden Jahren entscheidend sein, die Programmierkompetenzen neu zu definieren, die Studierende der Informatik sowie anderer Fachrichtungen benötigen. Ebenso gilt es, innovative didaktische Ansätze zu entwickeln, die den neuen Anforderungen gerecht werden. Das *CodeAbility Austria*-Konsortium hat sich seit 2023 intensiv mit diesen Herausforderungen auseinandergesetzt und erste Schritte unternommen, um diese Themen in die Lehre zu integrieren und wissenschaftlich zu analysieren.

Das Ergebnis des *CodeAbility Austria*-Projekts zeigt, dass universitätsübergreifende Zusammenarbeit in der Lehre nicht nur deren Qualität und Skalierbarkeit erheblich verbessert, sondern auch die internationale Sichtbarkeit der österreichischen Hochschulen stärkt. Gleichzeitig bietet diese

Kooperation eine ideale Basis, um die Programmierausbildung an die sich rapide wandelnden Anforderungen der digitalen Welt anzupassen.

Die weitere Unterstützung dieser Initiative ist von zentraler Bedeutung. Nur so können die erzielten Fortschritte nachhaltig in der Breite verankert, die Qualität der Programmierausbildung national gesichert und Österreichs Rolle als Vorreiter in der Programmierausbildung weiter ausgebaut werden.

Projektbeteiligte

Universität Innsbruck
Univ.-Prof. Dr. Ruth Breu (ruth.breu@uibk.ac.at) Dr. Michael Breu Ass. Prof. Clemens Sauerwein PhD Ass. Prof. Dr. Michael Vierhauser Dr. Michael Tschuggnall Eduard Frankford M.Sc. Lukas Kaltenbrunner M.Sc. Simon Priller M.Sc.
Paris Lodron Universität Salzburg
Univ.-Prof. Dr. Alexander Meschtscherjakov (Alexander.Meschtscherjakov@plus.ac.at) Vivien Wallner M.Sc. Dr. Heinz Hofbauer Assoc. Prof. Dr. Andreas Naderlinger Ao. Univ.-Prof. Dr. Helge Hagenauer Univ.-Prof. Dr. BSc. Sebastian Forster Dr. Daniel Kocher Assoz. Prof. Dr. Stefan Resmerita
Universität für Weiterbildung Krems
Martin Dobiasch PhD (martin.dobiasch@donau-uni.ac.at) Fabian Schober MA Univ.-Prof. Dr. Stefan Oppl
Universität Klagenfurt
Assoc. Prof. Dr. Klaus Schöffmann (klaus.schoeffmann@aau.at) Prof. Dr. Radu Prodan
TU Graz
Ass. Prof. iR. Dipl.-Ing. Dr.techn. Winfried Kernbichler (winfried.kernbichler@tugraz.at) Univ.-Prof. Dipl.-Ing. Dr.techn. Daniel Gruss (daniel.gruss@tugraz.at) David Camhy Maximilian Mandlez B.Sc.
TU Wien
o. Univ.-Prof. Dr. Gerti Kappel (gertrude.kappel@tuwien.ac.at) Dr. Stefan Podlipnig (stefan.podlipnig@tuwien.ac.at)
Johannes Kepler Universität Linz
Assoz. Univ.-Prof. Dr. (iris.groher@jku.at) A. Univ.-Prof. Mag. Dr. Reinhold Plösch (reinhold.ploesch@jku.at)

Publikationsliste

2024
M. Vierhauser; I. Groher; T. Antensteiner; and C. Sauerwein: <i>"Towards Integrating Emerging AI Applications in SE Education"</i> , In Proc. of the 2024 Conference on Software Engineering Education and Training, 2024.
E. Frankford, C. Sauerwein, P. Bassner, S. Krusche, and R. Breu: <i>"AI-Tutoring in Software Engineering Education"</i> , In Proc. of the International Conference on Software Engineering - Software Engineering Education and Training Track (ICSE SEET), 2024.
P. Bassner, E. Frankford, and S. Krusche: <i>"IRIS: An AI-driven virtual tutor for computer science education"</i> , In Proc. of the 2024 Innovation and Technology in Computer Science Education Conference, 2024.
I. Groher, M. Vierhauser, and E. Hartl: <i>"A Learning Analytics Dashboard for Improved Learning Outcomes and Diversity in Programming Classes"</i> , In Proc. of the 16th International Conference on Computer Supported Education (CSEDU), 2024.
M. Vierhauser, I. Groher, C. Sauerwein, T. Antensteiner, and S. Hatmanstorfer: <i>"Learning Analytics Support in Higher-Education: Towards a Multi-Level Shared Learning Analytics Framework"</i> , In Proc. of the 16th International Conference on Computer Supported Education (CSEDU), 2024.
E. Frankford, D. Crazzolaro, C. Sauerwein, M. Vierhauser, and R. Breu: <i>"Requirements for an Online Integrated Development Environment for Automated Programming Assessment Systems"</i> , In Proc. of the 16th International Conference on Computer Supported Education (CSEDU), 2024.
C. Sauerwein, T. Antensteiner, P. Salhofer, and R. Breu: <i>"Success Factors of Automated Programming Assessment Systems in Different Learning Environments"</i> , In Proc. of the 56th Hawaii International Conference on System Sciences (HICSS), 2024.
R. Plösch, I. Groher, and A. Hofer: <i>"Competence-based Assessment of Programming Assignments"</i> , In Proc. of the 36th Conference on Software Engineering Education and Training (CSEE&T), 2024.
V. Wallner, J. Santiago, M. Gaertner, and A. Meschtscherjakov, A.: <i>"From Entry to Empowerment: Motivations and Barriers for Female Students at the Beginning of their Programming Education"</i> , (under submission), 2024
M. Leopold and K. Schöffmann: <i>"An Open-Source Presentation Video Retrieval System Using Transformers"</i> , In Proc. of the 2nd International Conference on Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications (FAIEMA), 2024.
2023
C. Sauerwein, S. Priller, M. Dobiasch, S. Oppl, M. Felderer, and R. Breu: <i>"Lecturers' and Students' Experiences with an Automated Programming Assessment System"</i> , In Proc. of the 56th Hawaii International Conference on System Sciences (HICSS), 2023.
M. Leopold and K. Schöffmann: <i>"An Open-Source Presentation Video Retrieval System Using Transformers"</i> , In Proc. of the 2nd International Conference on Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications (FAIEMA), 2023.
C. Sauerwein, R. Breu, S. Oppl, I. Groher, T. Antensteiner, S. Podlipnig, and R. Prodan: <i>"CodeAbility Austria – Digitally Supported Programming Education at Austrian Universities"</i> , In Zeitschrift für Hochschulentwicklung, 18, 2023.
P. Grasserbauer, and R. Plösch: <i>"Value Based Prioritization of Requirements in Software Engineering Education"</i> , In Proc. of the 35th IEEE International Conference on Software Engineering Education and Training (CSEE&T), 2023.
C. Sauerwein, T. Antensteiner, S. Oppl, I. Groher, A. Meschtscherjakov, P. Zech, and R. Breu: <i>"Towards a Success Model for Automated Programming Assessment Systems"</i> , In Proc. of the 2023 Conference on Innovation and Technology in Computer Science Education (ITiCSE), 2023.
C. Kohlbacher, M. Vierhauser, and I. Groher: <i>"Common Code Quality Issues of Novice Java Programmers: A Comprehensive Analysis of Student Assignments"</i> , In Proc. of the 15th International Conference on Computer Supported Education (CSEDU), 2023.
M. Wieser, K. Schöffmann, D. Stefanics, A. Bollin, and S. Pasterk: <i>"Investigating the Role of ChatGPT in Supporting Text-Based Programming Education for Students and Teachers"</i> , In Proc. of the 16th International Conference on Informatics in Schools (ISSEP), 2023.
2022
I. Groher, M. Vierhauser, B. Sabitzer, L. Kuka, A. Hofer, and D. Muster: <i>"Exploring Diversity in Introductory Programming Classes: An Experience Report"</i> , In Proc. of the 44th International Conference on Software Engineering - Software Engineering Education and Training (ICSE-SEET), 2022.
2021
A. Hofer and I. Groher: <i>"Introducing Gamification in Introductory Programming Courses"</i> , In Proc. of the 15th European Conference on Games Based Learning (ECGBL), 2021.
A. Münzner, N. Bruckmoser, and A. Meschtscherjakov, A.: <i>"Can I Code? User Experience of an Assessment Platform for Programming Assignments"</i> . In Proc. of the 2nd International Computer Programming Education Conference (ICPEC), 2021.