10.20378/irb-108298

# Introducing a self-study course for learning textual programming at highschool with APFEL and ALeA

Dominic Lohr [1], Lars Wechsler[1], and Marc Berges [1]

Starting with block-based programming languages is a popular approach to introduce students to programming, as it removes the hurdle of dealing with syntax errors [WW17]. The transition to textual programming often results in e.g. syntax errors, leading to cryptic compiler messages that can frustrate beginners highlighting the need for enhanced support and adaptive feedback, especially when working on initial programming problems. The issue is particularly pronounced in self-study courses, where immediate guidance may be lacking. In this demo, we present a self-study course for beginners in textual programming with Java who already have experience with block-based programming languages and are now facing the transition to text-based programming. It is based on the 12th grade curriculum at Bavarian high schools [2] and is currently being piloted. In this course, the students work on programming problems using the educational IDE APFEL (Adaptive Programming Feedback for E-Learning) [Lo24].

APFEL combines static and dynamic code analysis to categorize program code into so-called answer classes – objectively observable states of an answer [Lo23]. This classification enables the generation of personalized feedback. In addition to standard functionality like executing code and unit tests, learners can select the desired feedback type in APFEL themselves. Using rich contexts such as answer classes, a model solution, and the task description are dynamically loaded into prompts to generate the desired feedback type using a state-of-the-art Large Language Model.

To further support the learners when working independently on the course content, the course material is semantically annotated and embedded in the adaptive learning assistant ALeA [Kr23]. In ALeA, learners can, for instance, add personal notes, discuss the course content collaboratively with fellow students in the script, or generate Guided Tours for concepts tailored to their individual learner model.

## Bibliography

[Kr23] Kruse, Theresa; Berges, Marc; Betzendahl, Jonas; Kohlhase, Michael; Lohr, Dominic; Müller, Dennis: Learning with ALeA: Tailored Experiences through Annotated Course Material. In: INFORMATIK 2023 - Designing Futures: Zukünfte Gestalten. Gesellschaft für Informatik e.V., Bonn, pp. 395–398, 2023.

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstraße 3, 91058 Erlangen, Deutschland, dominic.lohr@fau.de, https://orcid.org/0000-0002-6330-2327; lars.wechsler@fau.de; marc.berges@fau.de, https://orcid.org/0000-0002-9982-547X

[2] https://www.lehrplanplus.bayern.de/fachlehrplan/gymnasium/12/informatik/grundlegend-spaet

[Lo23] Lohr, Dominic; Berges, Marc; Kohlhase, Michael; Rabe, Florian: The Potential of Answer Classes in Large-scale Written Computer-Science Exams. In: Hochschuldidaktik Informatik HDI 2023. Jörg Desel, Simone Opel, Dortmund, pp. 179–189, 2023.

[Lo24] Lohr, Dominic; Berges, Marc; Chugh, Abhishek; Striewe, Michael: Adaptive Learning Systems in Programming Education: A Prototype for Enhanced Formative Feedback. In: Proceedings of DELFI 2024. Gesellschaft für Informatik e.V., 2024.

[WW17] Weintrop, David; Wilensky, Uri: Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. ACM Transactions on Computing Education, 18(1):1–25, 2017.