

IDE-integrierte Assistenz- und Lernsysteme zur Unterstützung von ProgrammieranfängerInnen beim Erlernen von Debugging-Strategien

Thema:

IDE-integrierte Assistenz- und Lernsysteme zur Unterstützung von ProgrammieranfängerInnen beim Erlernen von Debugging-Strategien

Art:

[BA](#)

BetreuerIn:

[Alexander Bazo](#)

BearbeiterIn:

Tim Eichinger

ErstgutachterIn:

[Christian Wolff](#)

ZweitgutachterIn:

[Niels Henze](#)

Status:

[abgeschlossen](#)

Stichworte:

[Software Engineering](#), [Softwaretechnik](#), [Lernsysteme](#), [Assistentssysteme für die Programmierausbildung](#)

angelegt:

2021-02-25

Anmeldung:

2021-06-01

Antrittsvortrag:

2021-04-26

Hintergrund

Der Begriff Debugging beschreibt das Suchen und Beheben schadhafter Stellen (*Bugs*) in Computerprogrammen. Debugging gehört dabei zu den wichtigsten Aufgaben von Programmieren und Programmiererinnen die zu diesem Zweck unterschiedliche Methoden einsetzen. Die Fähigkeit zur systematischen Fehleranalyse wird dabei bereits während der Ausbildung, z.B. im Rahmen klassischer Einführungskurse (CS 101) oder konkreten, studentischen Softwareprojekten benötigt. Die Notwendigkeit einer entsprechenden Vermittlung dieser Fähigkeiten im Rahmen der schulischen oder akademischen Ausbildung wurde bereits in zahlreichen Publikationen nachgewiesen [1]. Dabei wurden sowohl grundlegende Untersuchungen studentischen Debugging-Verhaltens vorgenommen [2][3] als auch konkrete Frameworks [4] oder Vorgehensmodelle [5] entworfen. Diese Arbeit kombiniert diese Ansätze mit der Idee, professionelle Entwicklungsumgebungen durch entsprechende Modifikationen für Lernende zugänglicher zu machen und zur konkreten Vermittlung von

Methodenkompetenz einzusetzen.

Zielsetzung der Arbeit

Diese Arbeit untersucht die Effektivität unterschiedlich interaktiv-gestalteter IDE-Integration zur Vermittlung von Grundlagenwissen im Bereich systematischer Debugging-Strategien. Zu diesem Zweck werden zwei Varianten eines Plugin für die Entwicklungsumgebung *Android Studio* erstellt. Das Plugin leitet die Studierenden bei der Verwendung von *printlining*-basierten Debugging-Ansätzen im eigenen Quellcode an. Neben einer passiven, auf die kontext-nahe Informationsvermittlung beschränkten Variante wird zusätzlich eine interaktive, stärker assistierende Version umgesetzt. Beide Varianten werden kursbegleitend unter möglichst realistischen Bedingungen im Rahmen einer Langzeitstudie untersucht. Im Rahmen einer Laborstudie wird im Anschluss an die Feldstudie die Auswirkung beider Apparate auf einen kurzfristig messbaren Lerneffekt untersucht.

Forschungsfragen

- Wirkt sich eine stärker assistierende Unterstützung von Studierenden positive auf das Erlernen und selbständige Anwenden systematischer Debugging-Strategien aus?

Konkrete Aufgaben

- Zentrale Literaturthemen sammeln und relevante Arbeiten/Quellen suchen.
- Konkreten Vorschlag für die didaktisch sinnvolle Vermittlung einer Debugging-Strategie erarbeiten.
- Zweistufige Implementierung des Plugins (wenig u. viel Interaktivität)
- Studiendesign planen
- Durchführung der Feldstudie und Erheben des kurzfristigen Lerneffekts (ggf. extra Studie)
- Auswertung und Ergebnispräsentation (Schriftliche Ausarbeitung)

Erwartete Vorkenntnisse

- Java (OOP)
- Android (Anwendungsentwicklung)
- Fortgeschrittene Erfahrung mit verschiedenen Debugging-Strategien

Weiterführende Quellen

[1] Renée McCauley, Sue Fitzgerald, Gary Lewandowski, Laurie Murphy, Beth Simon, Lynda Thomas & Carol Zander (2008) Debugging: a review of the literature from an educational perspective, *Computer Science Education*, 18:2, 67-92, DOI: [10.1080/08993400802114581](https://doi.org/10.1080/08993400802114581)

[2] Laurie Murphy, Gary Lewandowski, Renée McCauley, Beth Simon, Lynda Thomas, and Carol Zander. 2008. Debugging: the good, the bad, and the quirky – a qualitative analysis of novices' strategies. *SIGCSE Bull.* 40, 1 (March 2008), 163–167. DOI:<https://doi.org/10.1145/1352322.1352191>

[3] S. Fitzgerald, R. McCauley, B. Hanks, L. Murphy, B. Simon and C. Zander, „Debugging From the

Student Perspective,“ in IEEE Transactions on Education, vol. 53, no. 3, pp. 390-396, Aug. 2010, doi: 10.1109/TE.2009.2025266.

[4] Chen Li, Emily Chan, Paul Denny, Andrew Luxton-Reilly, and Ewan Tempero. 2019. Towards a Framework for Teaching Debugging. In Proceedings of the Twenty-First Australasian Computing Education Conference (ACE '19). Association for Computing Machinery, New York, NY, USA, 79–86. DOI:<https://doi.org/10.1145/3286960.3286970>

[5] Paulina Haduong and Karen Brennan. 2018. Getting Unstuck: New Resources for Teaching Debugging Strategies in Scratch (Abstract Only). In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). Association for Computing Machinery, New York, NY, USA, 1092. DOI:<https://doi.org/10.1145/3159450.3162248>

From:

<https://wiki.mi.ur.de/> - **MI Wiki**

Permanent link:

<https://wiki.mi.ur.de/arbeiten/debugging-assistent>

Last update: **14.02.2022 08:35**

