



Marco Hartmann

Pädagogische Hochschule Schwyz & ETH Zürich
marco.hartmann@phsz.ch



Michael Hielscher

Pädagogische Hochschule Schwyz
michael.hielscher@phsz.ch



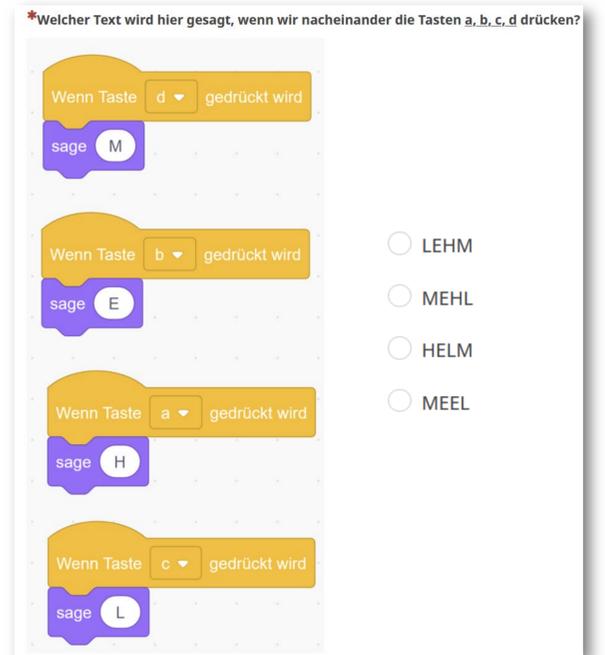
Eva Marinus

Pädagogische Hochschule Schwyz
eva.marinus@phsz.ch



Mögliche Programmierfehlvorstellungen bei Primarschulkindern

Erhebungen mit dem ProMAT-Instrument



Beispiel eines Items aus dem ProMAT. Die dritte Antwort ist korrekt. Die zweite Antwort könnte auf die Fehlvorstellung S2 hindeuten.

Diskussion

- Ziel der Untersuchung: Hinweise auf die Häufigkeit von möglichen Programmierfehlvorstellungen bei Primarschulkindern erfassen.
- Zwei Fehlvorstellungen zu Schleifen traten häufig auf: Schleifen erzeugen immer dieselben Ausgaben (L6) und jeder Befehl wird separat wiederholt (L5).
- Auch in anderen Kategorien wurden einige Fehlvorstellungen-Antworten häufig gewählt.
- Vergleich mit Swidan et al. (2018) zeigt allgemeine Übereinstimmungen, aber auch einzelne Unterschiede.
 - Häufigste Fehlvorstellung bei Swidan et al.: Verständnis der Reihenfolge von Anweisungen, was den Ergebnissen für S2 und S3 in unserer Studie entspricht.
 - Fehlvorstellung zu Kontrollstrukturen: L1 war bei Swidan et al. häufiger als in unserer Studie; C2 hatte in beiden Studien eine geringe Häufigkeit.
- Mit dem ProMAT lassen sich Programmierfehlvorstellungen bei Primarschülern systematisch untersuchen. Dies ermöglicht in der Praxis, relevante Fehlvorstellungen gezielt zu thematisieren.

Literatur

du Boulay, B. (1986). Some Difficulties of Learning to Program. *Journal of Educational Computing Research*, 2(1), 57–73. <https://doi.org/10.2190/3LFX-9RRF-67T8-UVK9>

Grover, S., & Basu, S. (2017). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and Boolean logic. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITICSE*, 267–272. <https://doi.org/10.1145/3017680.3017723>

Hermans, F., Swidan, A., Aivaloglou, E., & Smit, M. (2018). Thinking out of the Box: Comparing metaphors for variables in programming education. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3265757.3265765>

Mladenović, M., Boljat, I., & Žanko, Ž. (2017). Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education and Information Technologies*, 23(4), 1483–1500. <https://doi.org/10.1007/s10639-017-9673-3>

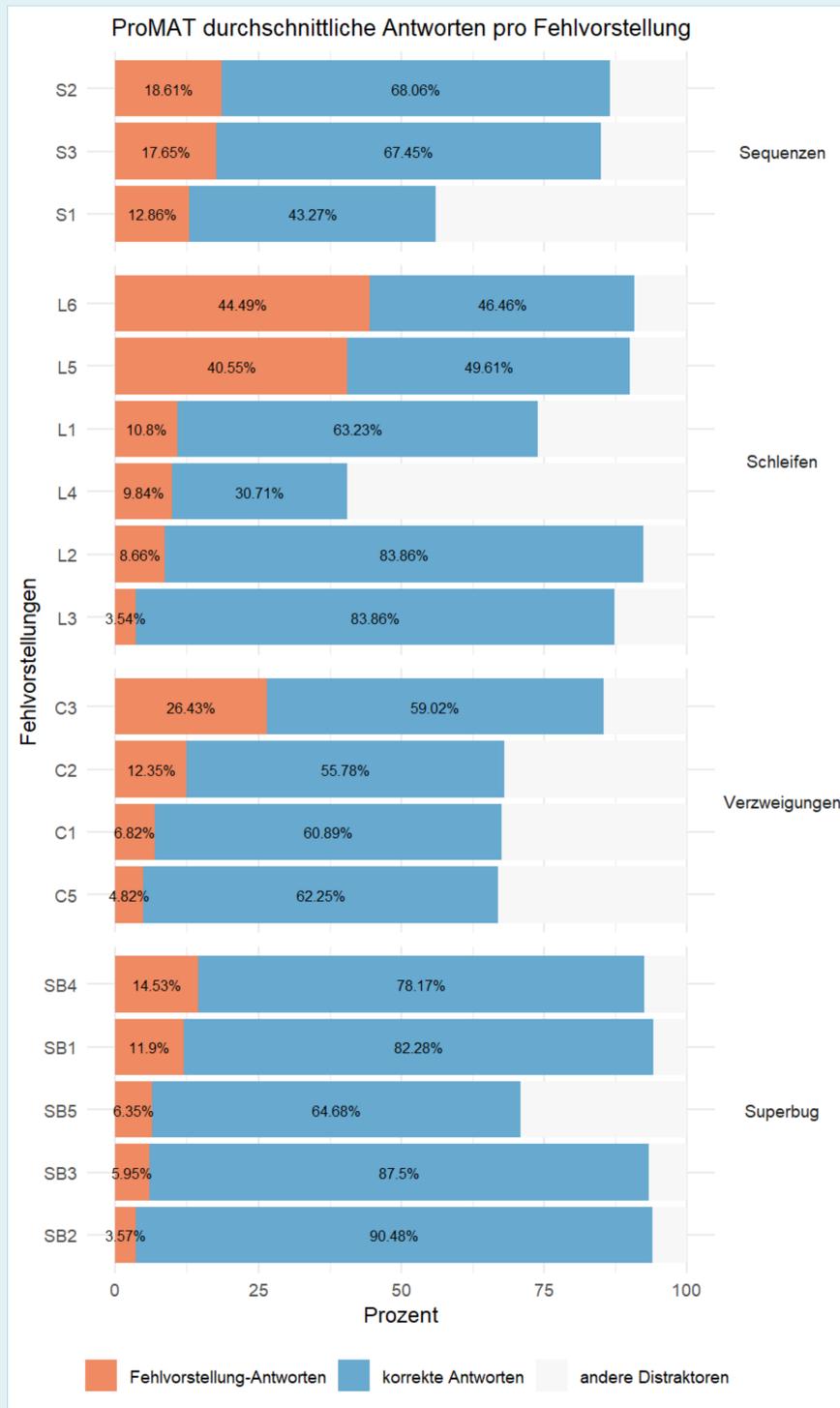
Swidan, A., Hermans, F., & Smit, M. (2018). Programming misconceptions for school students. *ICER 2018 - Proceedings of the 2018 ACM Conference on International Computing Education Research*, August, 151–159. <https://doi.org/10.1145/3230977.3230995>

Žanko, Ž., Mladenović, M., & Boljat, I. (2019). Misconceptions about variables at the K-12 level. *Education and Information Technologies*, 24(2), 1251–1268. <https://doi.org/10.1007/s10639-018-9824-1>

Žanko, Ž., Mladenović, M., & Krpan, D. (2022). Mediated transfer: Impact on programming misconceptions. *Journal of Computers in Education*, 0123456789. <https://doi.org/10.1007/s40692-022-00225-z>



Nummer	Beschreibung der Fehlvorstellung	Kategorie
L6	Manche Schüler:innen erklären, dass eine Schleife jeweils dieselben Aktionen wiederholt und erwarten, dass Schleifen in jeder Iteration genau die gleiche Ausgabe erzeugen.	Schleifen
L5	Wenn es mehrere Aktionen innerhalb einer Schleife gibt, neigen einige Schüler:innen dazu, jede Aktion separat zu wiederholen, bevor die nachfolgende Aktion wiederholt wird, anstatt eine Sequenz von Aktionen in einer Schleife auszuführen und dann die gesamte Sequenz zu wiederholen. Sprich, die Aktionen werden innerhalb der Schleife gruppiert.	Schleifen
C3	Die THEN-Verzweigung wird immer ausgeführt.	Verzweigungen
S2	Unterprogrammcode wird in der Reihenfolge ausgeführt, in der die Unterprogramme definiert sind.	Sequenzen
S3	Die Reihenfolge der Befehle spielt keine Rolle.	Sequenzen
SB4	Die Maschine versteht Englisch [oder Sprache des Benutzers/der Umgebung].	Superbug
S1	Magischer Parallelismus: Mehrere Zeilen eines (einfachen, nicht simultanen) Programms können gleichzeitig aktiv oder bekannt sein.	Sequenzen
C2	Eine falsche Bedingung beendet das Programm, wenn keine ELSE-Verzweigung vorhanden ist.	Verzweigungen
SB1	Der Computer kennt die Absicht des Programms oder eines Stücks Code (oder des Programmierers) und handelt entsprechend.	Superbug
L1	Angrenzender Code wird innerhalb der Schleife ausgeführt.	Schleifen
L4	WHILE-Schleifen enden, sobald die Bedingung "falsch" wird.	Schleifen
L2	Code nach der Schleife wird innerhalb der Schleife ausgeführt.	Schleifen
C1	Code nach der IF-Anweisung wird nicht ausgeführt, wenn die THEN-Klausel ausgeführt wird.	Verzweigungen
SB5	Die Semantik von Variablenamen in der natürlichen Sprache beeinflusst, welcher Wert welcher Variable zugewiesen wird.	Superbug
SB3	Das System lässt unvernünftige Operationen nicht zu.	Superbug
C5	Die Werte von bedingten Anweisungen werden ausgegeben.	Verzweigungen
SB2	Werte werden automatisch entsprechend dem logischen Kontext aktualisiert.	Superbug
L3	Code vor und nach der Schleife wird innerhalb der Schleife ausgeführt.	Schleifen



Ergebnisse

Die Tabelle links listet alle Fehlvorstellungen und ihre Konzeptkategorien, die in den Antwortoptionen des ProMAT abgebildet sind, in absteigender Häufigkeit auf.

Die Grafik rechts zeigt dieselben Fehlvorstellungen (nummeriert) nach Kategorien gruppiert und entsprechend der Häufigkeit der gewählten Antworten sortiert. Jeder Balken stellt den Prozentsatz der Antworten dar, die einer Fehlvorstellung (orange), der richtigen Antwort (blau) oder anderen Distraktoren (grau) entsprechen.

Sämtliche erhobenen Daten und aktuelle Versionen des ProMAT sind auf unserer OSF-Projektseite (QR-Code rechts) verfügbar.



Theoretischer Hintergrund

- Schon seit den 80-er Jahren Forschung zu Fehlvorstellungen im Programmieren (du Boulay, 1986).
- In letzten Jahren vermehrt untersucht, welche Fehlvorstellungen jüngere Lernende entwickeln, (Grover & Basu, 2017; Hermans et al., 2018; Mladenović et al., 2017; Žanko et al., 2019, 2022).
- Swidan et al. (2018) führten die erste umfassende Untersuchung von Fehlvorstellungen bei Kindern durch.
- Wir haben ein Instrument entwickelt, um Programmierfehlvorstellungen bei Schülerinnen und Schülern der 5. und 6. Klasse zu erfassen: Das Programming Misconception Assessment Tool – ProMAT.
- Konzeptkategorien Sequenzen, Schleifen, Verzweigungen und Superbug
- Die Programmierumgebung Scratch wird häufig im schulischen Kontext eingesetzt und wurde daher auch als Umgebung für die Implementierung unserer Testitems ausgewählt.

Forschungsfrage

Welche Programmier-Fehlvorstellungen finden wir bei Primarschulkindern, die mit Scratch gelernt haben?

Methode

- Alle Programmierfehlvorstellungen basieren auf der Literatur, hauptsächlich dem Inventar von Sorva (2013).
- Aus über 160 Fehlvorstellungen wählten wir jene aus, die für die 5. und 6. Klasse relevant sind.
- ProMAT enthält 20 Multiple-Choice-Fragen, verteilt auf vier Kategorien (Sequenzen, Schleifen, Verzweigungen und Superbug), basierend auf kurzen Programmbeispielen.
- Nur eine Antwort pro Frage ist korrekt, während andere Antworten spezifische Fehlvorstellungen widerspiegeln.
- Daten wurden zwischen 2022 und 2023 im regulären Unterricht mit Schulklassen erhoben, die Scratch zum Programmieren nutzten.
- Kinder bearbeiteten vor den Hauptfragen einfachere Aufgaben, um ihre Vertrautheit mit Scratch zu testen.
- Kinder, die weniger als 50 % der anfänglichen Aufgaben korrekt lösten, wurden ausgeschlossen; es blieben 255 Kinder (127 Mädchen) für die Auswertung.