

Peter LUDES-ADAMY, Hamburg

Erste Programmiererfahrungen im Mathematikunterricht der Grundschule

Einleitung

Als Voraussetzung für die Teilhabe am gesellschaftlichen Leben unter den Bedingungen von Digitalität werden informatische Kompetenzen gerade im Hinblick auf das Lernen *über* digitale Medien als unerlässlich erachtet (Brandhofer et al., 2018; Viermann & Ludes-Adamy, 2022). Dementsprechend sollten bereits Schüler*innen der Primarstufe beim Aufbau informatischer Kompetenzen (bspw. Programmieren) unterstützt werden (Gesellschaft für Informatik e.V., 2018; Kultusministerkonferenz (KMK), 2016). Da Informatik als Schulfach nicht im Fächerkanon der Primarstufe verankert ist, sind Lerngelegenheiten zum Aufbau informatorischer Kompetenzen in der Primarstufe fächerübergreifend zu realisieren. Insbesondere der Mathematikunterricht bietet sich für den Aufbau informatorischer Kompetenzen an, da eine Vielzahl mathematischer Konzepte (wie die Funktionsweise eines Algorithmus) eine tragende Rolle in der Informatik im Allgemeinen und beim Programmieren im Besonderen spielt. Programmieren stellt hierbei eine grundlegende Kompetenz aus dem Bereich Informatik dar, bei der sich durch ihre logische Struktur und deduktive Natur große Parallelen zu vielen inhaltsbezogenen mathematischen Kompetenzen aufzeigen lassen (Ludes-Adamy & Schütte, 2018, 2019). Im Beitrag wird ein Einblick in ein Dissertationsprojekt gegeben, in dem die Vorstellungen von Schüler*innen von dem Begriff *Programmieren* untersucht werden.

Theoretische Überlegungen

Im Rahmen der allgemeinen mathematischen Kompetenzen spielt das Problemlösen im Primarbereich eine große Rolle. Programmieraufgaben können als Modellierungsaufgaben bzw. als algorithmische Problemlöseaufgaben angesehen werden, die algorithmisches Denken oder „*computational thinking*“ erfordern (Gesellschaft für Informatik e.V., 2018; Tedre & Denning, 2016). Algorithmisches Denken stellt nach Schwill (1993) eine fundamentale Idee der Informatik bzw. nach Winter (1995) eine fundamentale Idee der Mathematik dar. Kortenkamp et al. (2019) differenzieren algorithmisches Denken in die Teilbereiche „*Algorithmen analysieren*“ und „*Algorithmen produzieren*“. Um im Zuge mathematisch-informatischer Bedeutungsaushandlungen rekonstruieren zu können, welche Vorstellungen Schüler*innen

mit dem Begriff *Programmieren* verbinden, wurden Schüler*innen im Rahmen eines Dissertationsprojektes dazu angeregt, sich über den Begriff auszutauschen.

Forschungsdesign

Im Rahmen des Dissertationsprojektes wurden Lernumgebungen für die Grundschule mit mathematisch-informatischen Inhalten entwickelt, die mit 47 Schüler*innen der 3. und 4. Klasse durchgeführt worden sind. Die Lernumgebungen wurden kooperativ gestaltet (Borsch, 2010; Krummheuer, 2007; Naujok, 2002), um bei den Schüler*innen kollektive Lernprozesse unterstützen zu können (Miller, 1986). Die Durchführungen wurde videografiert, transkribiert und mit Hilfe der Interaktionsanalyse (Krummheuer, 2011; Schütte et al., 2019) ausgewertet. Ziel der Auswertungen war es, bei den Schüler*innen Merkmale mathematisch-informatischer Bedeutungsaushandlungen (u. a. zu dem Begriff des Programmierens) zu rekonstruieren. Nachfolgend wird ein Beispiel vorgestellt, in dem zwei Schüler den Begriff *Programmieren* diskutieren.

Beispiel Nino & Christoph – Was ist Programmieren?

Als Einstiegsaufgabe zur Lernumgebung *Programmieren* diskutieren die Schüler Nino und Christoph darüber, was Programmieren ist und wofür man es braucht. Im Anschluss sollen sie eine gemeinsame Antwort verschriftlichen.

- | | | |
|----|-----------|--|
| 32 | Nino | Also programmieren braucht man eigentlich fast für alles\ |
| 33 | Christoph | Wirklich/ |
| 34 | Nino | Ja der Computer ist programmiert die Kamera ist programmiert |
| 35 | Christoph | Oke [schaut in die Kamera] jajaja ok |
| 36 | Nino | Ist fast alles programmiert |
| 37 | Nino | Öh, man programmiert jeden einzelnen Ablauf von einer Maschine |
| 38 | Christoph | Ja [lächelt] |
| 39 | Christoph | Ja so . wenn das passiert . dann mache das . . wenn dies passiert . dann mache das . immer halt so das klar ist was passiert sonst gehts kaputt. |
| 40 | Nino | Ja gut gut gut. Das schreiben wir auf. |

Nino stellt fest, dass Programmieren etwas ist, dass „*fast immer*“ gebraucht wird. Nino verweist dabei auf verschiedene technische Geräte. Er scheint mit Programmieren etwas zu verbinden, dass es erlaubt, technische Geräte nutzbar zu machen. Dies wird in Zeile 37 deutlich, als Nino zusammenfassend formuliert, dass einzelne Abläufe einer Maschine programmiert sind. Nino

führt das Konzept eines Computers auf eine allgemeinere Ebene, indem er von einer Maschine spricht (Zeile 37). Des Weiteren benutzt er die Formulierung „*jeden einzelnen Ablauf*“. Diese Formulierung lässt darauf schließen, dass Nino eine Vorstellung von der algorithmischen Funktionsweise eines Computers hat und ihm ebenso bewusst ist, dass ein Computer ohne Programmierung nichts leisten kann. Christoph scheint zunächst weniger an der allgemeinen Aushandlung beteiligt zu sein, da er eher passiv zustimmend auf Ninos Äußerungen reagiert (Zeile 6). Später formuliert Christoph aber selbst auch Antwortvorschläge, die Nino akzeptiert (Zeile 39/40). Er spricht in Zeile 39 auch schon die algorithmischen Strukturen an „*wenn (...) dann*“ und geht auf erste Merkmale eines Algorithmus ein. Indirekt scheint Christoph die Abgeschlossenheit von Algorithmen klar zu sein, was durch seine Erläuterung deutlich wird, dass auf eine Eingabe ein Ergebnis folgt „*das klar ist was passiert sonst gehts kaputt*“ (Ludes-Adamy & Schütte, 2022).

Die Vorstellungen vom Programmieren, die Nino und Christoph hier erkennen lassen, zeigen bereits ein relativ ausgeprägtes Begriffsverständnis. Auch die Vorstellung einer schrittweisen Ausführung im Sinne eines Algorithmus sollte positiv hervorgehoben werden. Speziell zum Begriff des Algorithmus und seiner Struktur ist im Primarbereich nicht unbedingt von mathematischer Vorerfahrung auszugehen, da dies im Unterricht nicht explizit thematisiert wird. Christoph tritt zunächst zurückhaltend auf, profitiert aber wohl von der kooperativen Situation und ist so an der kollektiven Aushandlung beteiligt.

Nino und Christoph lassen bereits ein tieferes Verständnis für die algorithmische Struktur von Programmierungen erkennen, auch wenn Christoph zunächst zurückhaltend auftritt. Mit Blick auf das individuelle Begriffsverständnis scheint insbesondere Christoph von der kooperativen Situation zu profitieren.

Mit Blick auf das Anfangs formulierte Ziel, Merkmale mathematisch-informatischer Bedeutungsaushandlungen zu rekonstruieren, wird anhand der untersuchten Sequenzen ersichtlich, dass die Schüler*innen bereits über anschlussfähige Konzepte zum Programmieren verfügen, die geeignet sind, um ein solides Begriffsverständnis zum Programmieren entwickeln zu können. Wie sich anhand des ausgewählten Beispiels erkennen lässt, kann der Aufbau eines solches Begriffsverständnis durch die Teilhabe an kollektiven Aushandlungsprozessen unterstützt werden.

Literatur

Brandhofer, G., Baumgartner, P., Ebner, M., Köberner, M., Trüttsch-Wihnen, C. & Wiesner, C. (2018). Bildung im Zeitalter der Digitalisierung. In S. Breit, F. Eder, K.

- Krainer, C. Schreiner, A. Seel & C. Spiel (Hrsg.), *Nationaler Bildungsbericht Österreich 2018* (S. 307–362). Leykam.
- Gesellschaft für Informatik e.V. (2018). *Kompetenzen für informatische Bildung im Primarbereich*. <https://uni-w.de/14k>
- Kortenkamp, U., Mahns, P. & Etzold, H. (2019). *Überlegungen zur informatisch-algorithmischen Grundbildung in der Grundschule*. <https://dlgs.uni-potsdam.de/sites/default/files/u3/Fundamentale%20Ideen%20der%20Informatik.pdf>
- Krummheuer, G. (2011). *Die Interaktionsanalyse*. http://www.fallarchiv.uni-kassel.de/wp-content/uploads/2010/07/krummheuer_inhaltsanalyse.pdf
- Kultusministerkonferenz (KMK). (2016). *Bildung in der digitalen Welt. Strategie der Kultusministerkonferenz*. https://www.kmk.org/fileadmin/Dateien/pdf/PresseUndAktuelles/2018/Digitalstrategie_2017_mit_Weiterbildung.pdf
- Ludes-Adamy, P. & Schütte, M. (2018). Cooperative learning in mathematics and computer science learning environments. In N. Planas & M. Schütte (Hrsg.), *Proceedings of the IV ERME Topic Conference “Classroom-based research on mathematics and language”* (S. 103–109). Technische Universität Dresden; ERME. <https://hal.archives-ouvertes.fr/hal-01856531>
- Ludes-Adamy, P. & Schütte, M. (2019). Programmieren mit Grundschullehrkräften: Fortbildung zu digitalen Medien im Mathematikunterricht und Programmieren mit dem Calliope mini. In D. Walter & R. Rink (Hrsg.), *Digitale Medien in der Lehrerbildung Mathematik – Konzeptionelles und Beispiele für die Primarstufe* (S. 239–264). WTM-Verlag.
- Ludes-Adamy, P. & Schütte, M. (2022). Immer der Reihe nach - Algorithmen in mathematisch-informatischen Lernumgebungen für den Primarbereich. In S. Ladel & U. Kortenkamp (Hrsg.), *Lernen, Lehren und Forschen mit digitalen Medien in der Primarstufe. Bd. 8: Informatisch-algorithmische Grundbildung im Mathematikunterricht der Primarstufe*. WTM-Verlag.
- Miller, M. (1986). *Kollektive Lernprozesse: Studien zur Grundlage einer soziologischen Lerntheorie*. Suhrkamp.
- Schütte, M., Friesen, R.-A. & Jung, J. (2019). Interactional analysis. A method for analysing mathematical learning processes in interactions. In G. Kaiser & N. Presmeg (Hrsg.), *Compendium for Early Career Researchers in Mathematics Education. ICME 13 Monograph* (S. 101–129). Springer.
- Schwill, A. (1993). Fundamentale Ideen der Informatik. *Zentralblatt Für Didaktik Der Mathematik*, 25(1), 20–31.
- Tedre, M. & Denning, P. J. (2016). The Long Quest for Computational Thinking. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, 120–129. <https://doi.org/10.1145/2999541.2999542>
- Viermann, M. & Ludes-Adamy, P. (2022). Fachunterricht unter den Bedingungen von Inklusion und Digitalität. In M. Jungwirth, N. Harsch, Y. Notensmeier, M. Stein & N. Willenberg (Hrsg.), *Diversität Digital Denken – The Wider View* (S. 93–102). WTM-Verlag.
- Winter, H. (1995). Mathematikunterricht und Allgemeinbildung. *Mitteilungen Der Gesellschaft Für Didaktik Der Mathematik*, 61, 37–46. <https://ojs.didaktik-der-mathematik.de/index.php/mgdm/article/view/69/80>