

SCHREITER, Saskia & DENNHARD, Jens
Heidelberg

Mathematische und informatorische Kompetenzen verknüpfen: Empirische Einblicke in Potenziale und Hürden

Die zunehmende Digitalisierung erfordert einen kompetenten Umgang mit Technologien. Diesem Bewusstsein folgend haben viele Länder weltweit die Förderung informatischer Kompetenzen in ihre schulischen Curricula integriert. In Deutschland gibt es in den meisten Bundesländern noch kein eigenes Fach für Informatik in der Grundschule und unteren Sekundarstufe. Stattdessen sind die entsprechenden Kompetenzen in den bestehenden Fächern zu vermitteln. Das Projekt CoM-MIT (Coden im Mathematikunterricht – Mathematik Informatik Transfer) setzt hier an. Im Projekt werden fächerverbindende Lernumgebungen zum Programmieren im Mathematikunterricht der Klassenstufen 3/4 und 5/6 entwickelt und auf ihre Effektivität hin beforscht. Dieser Beitrag gibt Einblicke in eine entwickelte Lernumgebung zu endlichen Zahlenfolgen und präsentiert erste Ergebnisse aus empirischen Erprobungen mit Schüler*innen der Klassenstufen 4 und 5.

Verknüpfung mathematischer und informatischer Kompetenzen

Die im Projekt entwickelte Lernumgebung zielt darauf ab, eine Förderung mathematischer und grundlegender informatischer Kompetenzen zu verknüpfen. Dabei besteht seit jeher eine Schnittmenge zwischen Mathematical Thinking und Computational Thinking – Letzteres wird u.a. mit Problemlösen und Begründen in Verbindung gebracht (z.B. Tsarava et al., 2022).

Auf der mathematischen Seite sollen die Schüler*innen im Kontext von endlichen Zahlenfolgen Strukturen und Muster erkennen sowie beschreiben. Dabei soll vornehmlich das mathematische Argumentieren gefördert werden. In Anlehnung an Brunner et al. (2022) werden insbesondere drei Teilkompetenzen fokussiert: Das Erkennen eines bestimmten arithmetischen Musters in einer Zahlenfolge, das Beschreiben dieses Musters, sowie das Entwickeln von Begründungen für das Zustandekommen dieses Musters.

Auf der informatischen Seite wird darauf abgezielt, Aspekte des algorithmischen Denkens als Strategie zum Problemlösen zu fördern (Ladel, 2021). In Anlehnung an Kortenkamp et al. (2019) liegt der Fokus auf den informatisch-algorithmischen Grundbausteinen Folge, Wiederholung und Verzweigung. Durch gezielte Aufgaben werden die Schüler*innen angeregt, bestehende Algorithmen zu analysieren, wodurch das Lesen und Verstehen von Codes im Zentrum steht. Darüber hinaus sollen sie Algorithmen produzieren, indem Codes vervollständigt oder eigens geschrieben werden.

Blockbasierte Programmiersprache zum Einstieg für Anfänger*innen

Die im Rahmen des Projekts entworfene Lernumgebung hat Anfänger*innen im Programmieren im Blick. Der Fokus liegt auf der Nutzung der blockbasierten, visuellen Programmiersprache NEPO, die über den frei zugänglichen Online-Editor Open Roberta Lab verfügbar ist. Dabei wird bewusst auf zusätzliche Hardware, wie Roboter, verzichtet. Stattdessen kommt eine Simulation des Einplatinencomputers Calliope mini mit integriertem Display zum Einsatz, um die technischen Anforderungen an den unterrichtlichen Einsatz gering zu halten (siehe Abb. 1).

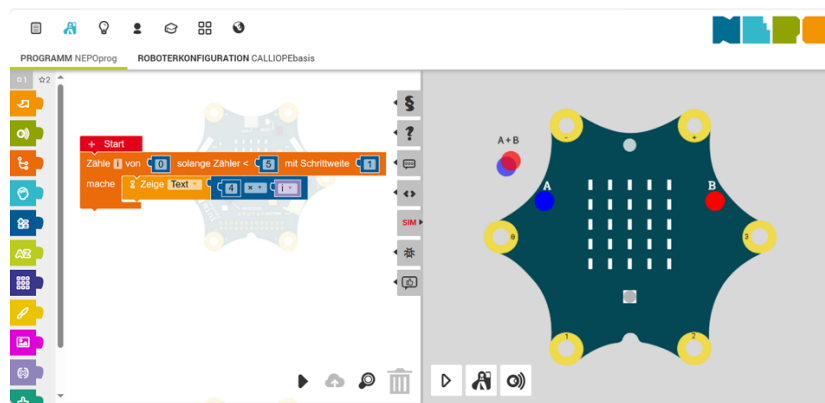


Abb. 1: Ansicht eines Codes im Open Roberta Lab bei geöffneter Simulation des Calliope mini

Methode und Design der Studie

Im Rahmen der empirischen Untersuchung wurden unter anderem folgende Forschungsfragen gestellt:

Inwiefern fördert die Unterrichtseinheit die Kompetenzen der Schüler*innen im Hinblick auf algorithmisches Denken und mathematisches Argumentieren im Kontext von Zahlenfolgen? Welche Hürden treten dabei auf?

Stichprobe: An dieser aktuell laufenden Studie nehmen insgesamt ca. 250 Schüler*innen der Klassenstufen 4 und 5 teil.

Studienablauf: Die Lernumgebung oder Teile derer wurden in 4-6h in den jeweiligen Klassen durchgeführt. Dabei profitierten Lehrende und Lernende vom einfachen Aufbau der Codes, die aus nicht mehr als sechs Code-Blöcken bestehen (vgl. Abb 2a). Es wurde die kooperative Methode des „Pair-Programming“ (Iskrenovic-Momcilovi, 2019) gewählt, bei der die Schüler*innen immer zu zweit arbeiten, was sich beim Erlernen neuer Programmiersprachen als geeignet erwiesen hat. Die Schüler*innen erhielten zunächst vollständig vorgegebene Codes, die über QR-Codes angewählt werden (vgl. Abb. 2b). Anhand von Forscheraufträgen (z. B. „Was denkst du passiert, wenn du die vordere 1 in der Zählschleife in eine 2 änderst?“

Schreibe deine Vermutung auf und prüfe sie anschließend“) konnten die Schüler*innen die Funktionsweisen der einzelnen Blöcke weitgehend selbstständig kennenlernen.

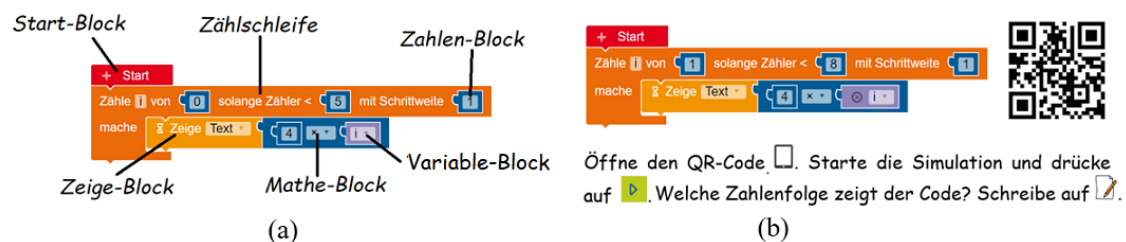


Abb. 2: Verwendete Blöcke (a) und Beispielaufgabe der Lernumgebung (b)

Im weiteren Verlauf untersuchten die Schüler*innen, wie der Calliope mini rechnet, und sollten dies anhand verschiedener Aufgaben zu arithmetischen Folgen 1. Ordnung begründen. Schrittweise wurde der Fokus weg vom Analysieren und hin zum Produzieren von Codes verlagert, indem zunächst lückenhafte oder unvollständige Codes ergänzt wurden, bevor die Kinder eigene Codes schrieben. Um Informationen zu den Lernprozessen der Schüler*innen zu erhalten, wurden unter anderem Arbeitsnotizen zu Aufgaben und Aktivitäten eingesammelt sowie Videos von Bildschirmaufnahmen mit begleitenden Audioaufnahmen der Schüler*innengespräche aufgezeichnet.

Erste Ergebnisse

Dieser Beitrag gibt einen ersten exemplarischen Einblick in die aktuell laufende qualitative Auswertung. Weitere Ergebnisse werden auf der Konferenz präsentiert und diskutiert.

Der folgende Dialog zeigt Ausschnitte aus dem Problemlöseprozess eines Schüler*innenpaares. Die Aufgabe ist, einen lückenhaften Code zur Folge der ungeraden Zahlen von 1 bis 21 zu ergänzen. Hierbei werden erste Einblicke zu typischen Hürden gegeben, sowie Strategien der Schüler*innen aufgezeigt, diese selbstständig zu überwinden.

(Es läuft die Zahlenfolge 1,4,7,10,13,16,19,22 über das Display)

S1: „Das ist falsch, das ist immer drei mehr. Wir brauchen nur zwei.“

S2: „Mach mal da (vordere Zahl im Mathe-Block) die zwei hin. Damit müsste es klappen. Weil er rechnet: 2 mal 0 plus 1, 2 mal 1 plus 1 und so weiter. Schau“

(S2 startet die Simulation. Es laufen die Zahlen 1,3,5,7,9,11,13,15 über das Display)

S1: „Ja, aber jetzt läuft es zu kurz. Wir müssen bis 21 hoch. Mach da (zweite Zahl in der Zählschleife) mal ne 10 hin. Jetzt müsste es!“

(S1 startet die Simulation)

S2 (zählt laut mit): „1,3,5,...19. Da fehlt noch eins mehr.... Ja klar, es ist ja kleiner, also bis 11 muss es (zweite Zahl in der Zählschleife), weil er bis 2 mal 10 plus 1 am Schluss rechnet“.

Im Rahmen des Mathematischen Argumentierens lassen sich folgende Teilaspekte erkennen: Die Schüler*innen haben das arithmetische Muster in der Zahlenfolge erkannt und diese Einsicht genutzt, um ihre erste Lösung zu verbessern. Sie beschrieben das Muster rekursiv („immer zwei mehr“) und entwickelten eine Begründung für das Zustandekommen dieses Musters („weil er rechnet...“). Gleichzeitig wurden die informatisch-algorithmischen Grundbausteine Folge und Wiederholung berücksichtigt. Die Schüler*innen erkannten, dass die einzelnen Schritte des Algorithmus der Reihe nach ausgeführt werden, was im Code vorgeben wird (hier: mit schrittweise 1). Zudem mussten Zählschleifen in den Codes beachtet werden, ebenso wie Start- und Stopp-Bedingungen, die über die Zahlen-Blöcke in der Zählschleife vorgegeben werden. Diese wurden von den Schüler*innen zunächst fehlerhaft ergänzt und dann eigenständig korrigiert. Insgesamt gelang es den Schüler*innen, einen bestehenden lückenhaften Code zu analysieren, zu vervollständigen und gemäß der Aufgabenstellung eigenständig zu korrigieren. Diese Symbiose aus Mathematik und Informatik steht im Mittelpunkt des Projektes.

Förderhinweis: Das Projekt CoM-MIT wird von der Klaus-Tschira-Stiftung gefördert. Darüber hinaus besteht eine Zusammenarbeit mit dem Fraunhofer Institut IAIS, das die Einrichtung eines projektspezifischen Servers unter <https://commit.open-roberta.org> ermöglichte.

Literatur

- Brunner, E., Lampart, J. & Jullier, R. (2022). Schriftliches mathematisches Argumentieren in zwei unterschiedlichen Inhaltsbereichen in den Jahrgangsstufen 4–6. *Journal für Mathematik-Didaktik*, 43, 463–493.
- Iskrenovic-Momcilovic, O. (2019). Pair programming with scratch. *Education and Information Technologies*, 24, 2943–2952.
- Kortenkamp, U., Etzold, H., & Mahns, P. (2021). Algorithmen im Alltag zur Umsetzung fundamentaler Ideen der Informatik in der Grundschule. In V. Frederking & S. Ladel (Hrsg.), *Grundschule digital. Innovative Konzepte für die Fächer Deutsch und Mathematik* (S. 163–182). Waxmann.
- Ladel, S. (2021). Digitale Medien im Mathematikunterricht der Grundschule. In V. Frederking & S. Ladel (Hrsg.), *Grundschule digital. Innovative Konzepte für die Fächer Deutsch und Mathematik* (S. 29–41). Waxmann.
- Tsarava, K., Moeller, K., Roman-Gonzalez, M., Golle, J., Leifheit, L., Butz, M., Ninaus, M. (2022). A cognitive definition of computational thinking in primary education. *Computers & Education*, 129, 1–14.