

- Perspectives on Teaching and Learning Mathematics with Virtual Manipulatives* (S. 275–300). Springer International Publishing. DOI:10.1007/978-3-319-32718-1_12
- National Council of Teachers of Mathematics (NCTM). (2014). *Principles to actions: Ensuring mathematical success for all*. Reston, VA.
- National Council of Teachers of Mathematics (NCTM). (2000). *Principles and Standards for School Mathematics*. Reston, VA.
- Rabardel, P. (2002). People and technology: a cognitive approach to contemporary instruments. HAL, Université Paris, tinyurl.com/2yfrqs2r
- Rauh, B. (2012). Höheres Lernen mit digitalen Medien – auch im Bereich der Arithmetik? In S. Ladel, C. Schreiber (Hrsg.), *Lernen, Lehren und Forschen in der Primarstufe* (S. 37–58). Franzbecker.
- Schmidt, S., & Müller, M. (2020). Students learning with digital mathematical tools – three levels of instrumental genesis. In B. Barzel, R. Bebernik, L. Göbel, M. Pohl, H. Ruchniewicz, F. Schacht, & D. Thurm (Hrsg.), *Proceedings of the 14th International Conference on Technology in Mathematics Teaching – ICTMT 14* (S. 378–383). DOI:10.1080/0020739X.2016.1264635
- Shin, D., Smith, R. C., & Kim, S. (2018). Evaluating technology for teaching mathematics. *Mathematics Teaching in the Middle School*, 24(3), 156–163. DOI:10.5951/mathteacmiddscho.24.3.0156
- Smith, R. C., Shin, D., & Kim, S. (2017). Prospective and Current Secondary Mathematics Teachers' Criteria for Evaluating Mathematical Cognitive Technologies. *International Journal of Mathematical Education in Science and Technology*, 48(5), 659–681. DOI:10.1080/0020739X.2016.1264635
- Smith, R., Shin, D., Kim, S., & Zawodniak, M. (2018). Novice secondary mathematics teachers' evaluation of mathematical cognitive technological tools. *Contemporary Issues in Technology and Teacher Education*, 18(4), 606–630.
- Thurm, D., & Barzel, B. (2020). Effects of a professional development program for teaching mathematics with technology on teachers' beliefs, self-efficacy and practices. *ZDM Mathematics Education*, 52(3), 1411–1422.
- Traglová, J., Clark-Wilson, A., & Weigand, H.-G. (2018). Technology and resources in mathematics education. In T. Dreyfus, M. Artigue, D. Potari, S. Prediger, & K. Ruthven (Hrsg.), *Developing research in mathematics education: Twenty years of communication, cooperation and collaboration in Europe* (S. 142–161). Springer.
- Whalen, J. (2020). Should teachers be trained in emergency remote teaching? Lessons Learned from the COVID-19 Pandemic. *Journal of Technology and Teacher Education*, 28(2), 189–199.

PD Dr. habil. Matthias Müller, Friedrich-Schiller-Universität Jena
E-Mail: matthias.mueller.2@uni-jena.de

Alexander Hörig, Friedrich-Schiller-Universität Jena
E-Mail: alexander.hoerig@uni-jena.de

Das Projekt <colette/> Computational Thinking (auch) im Mathematikunterricht

Tim Läufer, Rebecca S. Stäter und Matthias Ludwig

Die digitale Transformation hat unser aller Leben fundamental geändert. Die Nutzung von digitalen Medien in der Freizeit von Schülerinnen und Schülern liegt seit Jahren konsistent hoch, so haben beispielsweise deutlich über 94 % der jugendlichen Schülerinnen und Schüler in Deutschland ein Smartphone oder nutzen zum Lernen oder in der Freizeit das Internet (Feierabend et al., 2021). Konkret sind 12- bis 19-Jährige durchschnittlich 241 Minuten am Tag online – mehr als 4 Stunden täglich (ebd.). Die Kompetenzen für einen verantwortungsvollen und zukunftsorientierten Umgang damit werden aber nicht automatisch aufgebaut. Viele

der sog. *21st Century Skills* sind nicht nur heute, sondern auch zukünftig im beruflichen sowie privaten Leben essenziell (Wing, 2006, Lodi & Martini, 2021).

Einer dieser Skills ist *Computational Thinking* (CT), welches 2006 in einem Seminarpapier von Jeannette Wing für Aufsehen sorgte. Dieses Papier stieß eine große Debatte über die Definition und Fächerverortung von formalem, algorithmischem Denken an, die bis heute andauert (Bocconi et al., 2016; Lodi & Martini, 2021). CT liegt (kurz zusammengefasst) das strukturierte, abstrakte, formale und logische Problemlösen zugrunde, mit dem sich

die heutige Welt betrachten lässt (Wing, 2006, Bocconi et al., 2016, Weintrop et al., 2016, Lodi & Martini, 2021). Ein großer Teil von CT ist algorithmisches Denken.

Probleme für und mit dem Computer lösen

CT wird von vielen Autorinnen und Autoren (auch) im Mathematikunterricht verortet. So setzen bspw. Bescherer und Fest mathematisches Denken und CT in eine Beziehung zueinander, indem sog. Mikrowelten zum entdeckenden Lernen genutzt werden (Bescherer & Fest, 2020). Van Borkulo et al. untersuchten außerdem CT (besonders den Aspekt des algorithmischen Denkens) bei der Nutzung von Geogebra, einer Geometriesoftware (u.A.) für den Mathematikunterricht, und stießen auf positive Rückmeldungen von sowohl den Schülerinnen und Schülern als auch von den Lehrkräften zu Unterrichtsinhalten, die gezielt CT fördern:

Students' overall reflections in the interviews indicate that they welcomed the use of computational tools in their calculus lessons. They perceived the benefits of using dynamic mathematics software as more significant than the challenges new computational tools bring. [...] The teacher was particularly satisfied with the successful completion rate of the assignments [...] (van Borkulo et al., 2021, S. 5)

Obwohl einige negative Aspekte wahrgenommen wurden, haben Lehrkräfte sowie Schülerinnen und Schüler die Arbeit mit Aufgaben zur Generalisierung und algorithmischem Denken im Mathematikunterricht in der Summe als gewinnbringend wahrgenommen.

Im Fokus: Algorithmen

Algorithmen sind nicht nur in der Mathematik (Horner-Schema, Heron-Verfahren, Gauß-Algorithmus, ...), sondern auch im Alltag allgegenwärtig: Ein prominentes Beispiel ist die Wahl der Kasse im Supermarkt. Dort iterieren wir alle implizit über die vor uns befindlichen Warteschlangen, schätzen die Zeiten ab, die wir bei der jeweiligen Wahl anstehen müssten, und suchen daraufhin diejenige aus, welche die absehbar kürzeste Wartezeit hat (Wing, 2006). Algorithmisches Denken kann definiert werden als das Erreichen einer Lösung durch klar definierte Lösungsschritte und ist ein zentraler Teil von CT (Bocconi et al., 2016). Das ist vor Allem mit Hilfe von Programmierung abbildbar.

Andere Autoren führen an, dass das algorithmische Denken zudem das mathematische Argumentieren fördern kann, und umgekehrt, wenn der Algorithmus selbst als Lerngegenstand beleuchtet

wird und nicht als Nebenprodukt genutzt wird (Stephens & Kadjevich, 2020). Dennoch ist die direkte Förderung von algorithmischem Denken in der Mathematik unterrepräsentiert (van Borkulo et al., 2021).

Algorithmisches Denken ist laut Bocconi et al. (2016) nicht der einzige CT-Skill. *Abstraktion* vereinfacht Sachverhalte, durch Beschränkung auf das Wesentliche und damit verbunden auch das Weglassen von unnötigen Details, aber auch die Auswahl der Repräsentation eines Systems, da dieses die relevanten Details hervorhebt und die irrelevanten Details dadurch in den Hintergrund treten (Csizmadia et al., 2015). Davon getrennt ist ein anderer CT-Skill, die *Generalisierung*, welche das Erkennen und Nutzen von Mustern, Ähnlichkeiten und Verbindungen innerhalb und zwischen verschiedenen Inhalten ist. Hier werden Fragen wie „Ist dieses Problem ähnlich wie ein anderes Problem, welches ich schon mal gelöst habe?“ oder „Worin unterscheidet sich das?“ gestellt (Csizmadia et al., 2015).

Die *Dekomposition*, ebenfalls ein CT-Skill, beschreibt das Aufteilen des Problems in kleinere Teilprobleme, und spielt bei der Generalisierung eine große Rolle. Hierbei werden die kleineren Teilprobleme zuerst gelöst, um dann das gesamte Problem zu lösen. Diese Teilprobleme können sich in verschiedenen Problemen wiederfinden und somit sind deren Lösungen auch für andere Probleme wiederverwendbar.

Nun soll die Lösung von Problemen in vielen Disziplinen auch automatisiert ablaufen; *Automatisierung* ist auch ein CT-Skill, wobei es sich um einen arbeitssparenden Prozess handelt, der den Computer zu effizientem Maße nutzt (Lee et al., 2011). Damit eng verbunden ist das oben erwähnte *Algorithmische Denken*. Bei der Automatisierung sollen die Schülerinnen und Schüler lernen, wie sie den Computer und ihre digitalen Werkzeuge optimal nutzen.

Weil die Lösung von Problemen, gerade automatisiert, nicht immer reibungslos verläuft, müssen Schülerinnen und Schüler *debuggen* können. Beim CT-Skill *Debugging* geht es um das systematische, strukturierte Finden und Beheben von Fehlern (Csizmadia et al., 2015). Das Wort „Debugging“ setzte sich dabei aus dem Wort „Bug“, was einen Fehler im Programmcode bezeichnet, und „de“, was deutlich machen soll, dass dieser Fehler im Programmcode gefunden und entfernt wird. Vor Allem beim algorithmischen Denken in Verbindung mit Automatisierung ist diese Kompetenz von Nöten zum Aufbau eines *kompetenten* Umgangs mit den gegebenen Werkzeugen. CT hat eine enge Verbindung zu Problemlösekompetenzen (Labusch et al., 2019), da das Lösen von Problemen auch als ein Prozess von einem unerwünschten Stadium zu

einem erwünschten, finalen Stadium verstanden werden kann. Im Laufe dieses Prozesses müssen Barrieren überwunden, *Fehler* in dem Lösungsweg vielleicht sogar auch in der Herangehensweise gefunden und gelöst werden.

Es zeigt sich also, dass CT ein vielschichtiges Konzept ist, welches in vielen Disziplinen genutzt wird und genutzt werden kann. In dessen Mittelpunkt steht dabei das algorithmische Denken. Diese Kompetenz wird (u. A.) durch Programmierung angesprochen und gefordert:

Almost all teachers expressed their concerns about including programming in the math curriculum, mainly because of time constraints. They thought that learning a specific digital tool or programming language might take too much time. [...] However, almost all teachers saw the added value of using programming in math education to address CT. (van Borkulo et al., 2020, S. 3)

Diese Studie von van Borkulo (2020) zeigt, dass die Förderung von CT im Mathematikunterricht mit Hilfe von Programmierung einen klaren Mehrwert hat. Weiter untersuchten viele Autorinnen und Autoren die Förderung von CT im Mathematikunterricht bspw. mit Geogebra (van Borkulo et al., 2021), Mikrowelten (Bescherer & Fest, 2020), oder durch Entwicklung einer App die Förderung des Verständnisses von Primzahlen (Kong, 2019). Allerdings steht der Umsetzung von CT-Förderung im (Mathematik-)Unterricht ein großer Vorbereitungsaufwand entgegen: Viele Lehrkräfte wünschen sich aus Zeitgründen unterrichtsfertiges Material („ready-to-use teaching materials“, vgl. van Borkulo et al., 2020), welches nur geringen Vorbereitungsaufwand benötigt. Dies ist ein zentrales Ziel von <colette/>. Zusätzlich argumentieren viele Autorinnen und Autoren, dass CT nicht als eigenes Fach in das sonst schon volle Curriculum eingeführt, sondern in MINT-Unterrichtsfächer integriert werden soll (Bocconi et al., 2016; Stephens & Kadijevich, 2020; Weintrop et al., 2016).

Das Projekt <colette/>

Mit dem Projekt <colette/> (Computational Thinking Learning Environment For Teachers in Europe, Erasmus+; KA201, 09/20–08/23), wird in einem Zweikomponentensystem mit Webportal zur Lernpfaderstellung (Roth, 2015) und einer Hybrid-App für Android und iOS zur Aufgabenbearbeitung ein Werkzeug entwickelt, welches im Unterricht einfach genutzt werden kann. Dabei wird explizit CT gefördert, und zwar ganz ohne teure programmierbare Drohne, schwer vorzubereitende Materialien, komplexe Programmierumgebungen oder



Abbildung 1. Der AR-Marker, der bei der Verwendung der <colette/>-App zur Positionierung des Objekts in der „Realität“ verwendet wird.

andere zeit- und kostenintensive Voraussetzungen. Dadurch adressieren wir den oben von van Borkulo et al. angesprochenen Punkt der „ready-to-use teaching materials“ direkt.

Ein weiterer Punkt ist die gezielte Entwicklung für so viele Schülerinnen und Schüler wie möglich, ohne Abhängigkeit des sozioökonomischen Status: Es werden nur Smartphones (für die App für die Schülerinnen und Schüler) und ein PC (für das Webportal für die Lehrkraft), sowie ein Drucker für das Drucken des *Augmented Reality* (AR)-Markers (Abbildung 1) benötigt, falls es sich um eine AR-Aufgabe handelt. AR ist die kameragestützte Ansicht von (meist dreidimensionalen) Objekten, wobei dieses Objekt auf dem Bildschirm „in die Realität“ gelegt wird, da es virtuell auf dem Bildschirm des Smartphones in die Ansicht der Umgebung eingebaut wird.

Damit greift <colette/> auch bei den Schulen das Problem auf, welche die mangelnde digitale Ausstattung als die größte Herausforderung in der Corona-Krise wahrgenommen haben (forsa, 2020). Mit dem „Bring Your Own Device“-Ansatz, wo alle Schülerinnen und Schüler das eigene Handy nutzen, kann auch jeder teilhaben, da, wie oben erwähnt, über 94 % der Schülerinnen und Schüler in Deutschland Smartphones besitzen (Feierabend et al., 2021). Alternativ kann die App auch auf Schul-tablets installiert und genutzt werden.

Im Webportal können Lehrkräfte generische Aufgaben (vgl. Ludwig & Jablonski, 2020) modifizieren (Abbildung 2), sie können hierbei aus einer generischen Aufgabe eine auf die Bedürfnisse der Schülerinnen und Schüler und des Unterrichtsgeschehens angepasste Aufgabe erstellen. Dabei wählen die Lehrkräfte bei einer generischen Aufgabe ein Szenario, welches als eine Art Kontext- oder Inhaltsvariation aufgefasst werden kann, und ein Antwortformat inklusive Aufgabenstellung aus.

Zu jeder generischen Aufgabe gehören in <colette/> auch Hinweise für die Aufgabenbearbeitung, die in jedem Szenario bei jedem Antwortformat didaktisch ausgearbeitet sind (vgl. Ludwig & Jablonski, 2020). Dort besteht im ersten Prototyp die

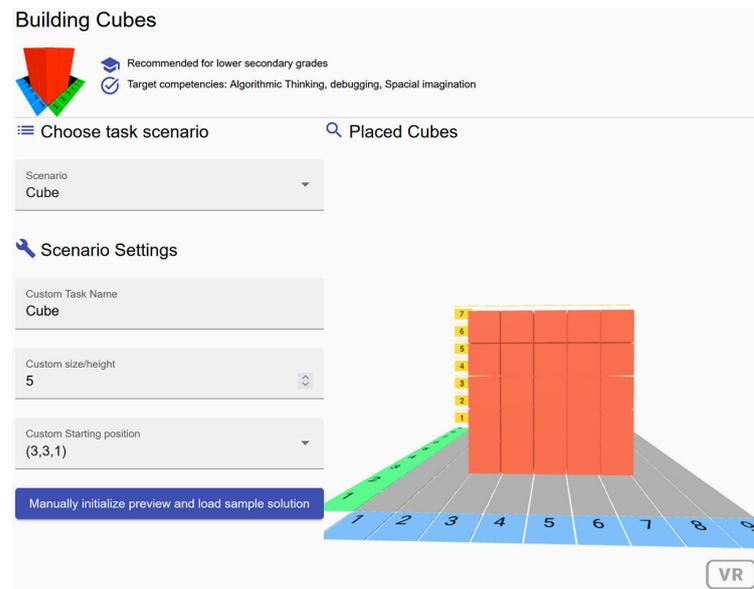


Abbildung 2. Der Webportal-interne Aufgabeneditor mit den dazugehörigen, anpassbaren Einstellungen der generischen Aufgabe „Building Cubes“. Das ist die Ansicht der Lehrkräfte während der Erstellung eines Pfads für die Schüler*innen.

Möglichkeit für textuelle Hinweise (Abbildung 4). Weitere Medieninhalte wie Bilder und Videos sind geplant.

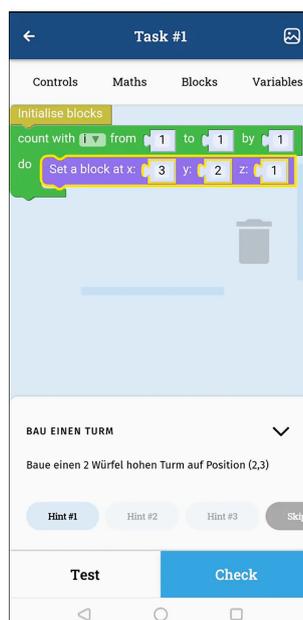


Abbildung 3. Der App-interne Editor für die Implementationsaufgaben. Das ist die Ansicht der Schüler*innen, bei der Bearbeitung der Aufgaben.

Sobald eine individuelle Aufgabe erstellt wurde, kann sie zu einem Lernpfad hinzugefügt werden. Dieser Pfad ist durch einen Nummerncode identifizierbar. Wenn diese nun in der App der Schüler und Schülerinnen eingegeben wird, werden die von der Lehrkraft erstellten Aufgaben auf dem Handy der Schüler und Schülerinnen zur Bearbeitung angezeigt (Abbildung 3). Die Schülerinnen und Schüler können nun die für sie konzipierten Aufgaben be-

arbeiten und ihre Lösung abgeben, welche dann die Lehrkraft für die ganze Klasse gesammelt im *digitalen Klassenzimmer* (Ludwig & Jablonski, 2020) in einem *Lernportfolio* einsehen kann. Ziel des digitalen Klassenzimmers ist die Auf- und Darstellung der Klassenfortschritte mit Möglichkeit der Anzeige einzelner Lösungsversuche. Dadurch ist eine tiefere Analyse der Lösungen und Lösungswege, sowie der eventuellen Schwierigkeiten möglich.

Ausblick

Zu Beginn des Jahres 2022 ist ein Prototyp des Webportals und der App mit der Aufgabengruppe der Bausteine live gegangen. In den kommenden Monaten sind weitere generische Aufgaben geplant, beispielsweise zu den Themen *Graphenprobleme*, *Prozessdiagramme*, *Drohnen-Flüge* sowie *Informatik Unplugged*. Das digitale Klassenzimmer wird vermutlich in der zweiten Version von <colette/> in 2023 verfügbar sein. Zum Ende des Projektes (August 2023) soll ein voll funktionsfähiges Webportal inklusive App und reichhaltiger Aufgabensammlung für Unterrichtszwecke kostenfrei zur Verfügung stehen. Weitere Informationen finden Sie unter www.colette-project.eu.

Literatur

Bescherer, C., & Fest, A. (2020). Mathematische Entdeckungen und Computational Thinking. In H.-S. Siller, W. Weigel & J. F. Wörler (Hrsg.), *Beiträge zum Mathematikunterricht 2020. 54. Jahrestagung der Gesellschaft für Didaktik der Mathematik*. WTM-Verlag. DOI:10.17877/DE290R-21232



Abbildung 4. Die Aufgabenvorschau inklusive Hinweiseditor. Es können weitere Hinweise hinzugefügt werden und bestehende Hinweise abgeändert oder ganz gelöscht werden. Dies stellt sicher, dass die Hinweise immer auch passend zum Unterrichtsgeschehen sind.

- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *JRC Science For Policy Report: Developing Computational thinking in Compulsory Education*. Implications for policy and practice. Seville (Spain). European Commission, Joint Research Centre. DOI:10.2791/792158
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational Thinking: A Guide for Teachers*. computingatschool.org.uk/computationalthinking
- Feierabend, S., Rathgeb, T., Kheredmand, H., & Glöckler, S. (2021). *JIM 2021: Jugend, Information, Medien*. Basisuntersuchung zum Medienumgang 12- bis 19-Jähriger in Deutschland. Stuttgart. tinyurl.com/yxlxusf8
- forsa (2020). *Das Deutsche Schulbarometer Spezial: Corona Krise: Folgebefragung*. tinyurl.com/2d3as37u
- Kong, S.-C. (2019). Learning Composite and Prime Numbers Through Developing an App: An Example of Computational Thinking Development Through Primary Mathematics Learning. In S.-C. Kong & H. Abelson (Hrsg.), *Computational Thinking Education* (1. Aufl.). Springer Singapore. 145–166
- Labusch, A., Eickelmann, B., & Vennemann, M. (2019). Computational thinking processes and their congruence with problem-solving and information-processing. In S. C. Kong & H. Abelson (Hrsg.), *Computational Thinking Education* (S. 65–78). Singapore: Springer
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37. DOI:10.1145/1929887.1929902
- Lodi, M., & Martini, S. (2021). Computational Thinking, Between Papert and Wing. *Science & Education*, 30(4), 883–908. DOI:10.1007/s11191-021-00202-5
- Ludwig, M., & Jablonski, S. (2020). MathCityMap: Mit mobilen Mathtrails Mathe draußen entdecken. *MNU Journal*, (01/2020), 29–36.
- Roth, J. (2015). Lernpfade – Definition, Gestaltungskriterien und Unterrichtseinsatz. In J. Roth (Hrsg.), *Medienvielfalt im Mathematikunterricht: Lernpfade als Weg zum Ziel* (S. 3–25). Springer Spektrum.
- Stephens, M., & Kadjevich, D. M. (2020). Computational/Algorithmic Thinking. In S. Lerman (Hrsg.), *Encyclopedia of Mathematics Education* (S. 117–123). Springer International Publishing. DOI:10.1007/978-3-030-15789-0_100044
- van Borkulo, S., Chytas, C., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Computational Thinking in the Mathematics Classroom: Fostering Algorithmic Thinking and Generalization Skills Using Dynamic Mathematics Software. In M. Berges, A. Mühlhling & M. Armoni (Hrsg.), *The 16th Workshop in Primary and Secondary Computing Education* (S. 1–9). ACM. DOI:10.1145/3481312.3481319
- van Borkulo, S., Kallia, M., Drijvers, P., & Tolboom, J. (2020). Computational practices in mathematics education: Experts' opinions. In *Conference: The 14th International Congress on Mathematical Education*, Shanghai.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. DOI:10.1007/s10956-015-9581-5
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. DOI:10.1145/1118178.1118215
- Tim Läufer, Goethe Universität Frankfurt am Main
E-Mail: laeufer@math.uni-frankfurt.de
- Rebecca S. Stäter, Goethe Universität Frankfurt am Main
E-Mail: staeter@math.uni-frankfurt.de
- Matthias Ludwig, Goethe Universität Frankfurt am Main
E-Mail: ludwig@math.uni-frankfurt.de