

Programmieren in der Grundschule – aber wie?

Dies ist nur eine der Fragen, mit der sich die Professur für Didaktik der Informatik an der TU München in den letzten Jahren beschäftigt. Neben dem Entwickeln und Erproben von geeigneten Materialien und Methoden für das Programmieren in der Grundschule steht im Fokus, wie man Lehrkräfte dazu befähigen kann, die Thematik im Unterricht umzusetzen.

Katharina Geldreich entwickelt im Team um Prof. Peter Hubwieser Ideen, wie man Kindern in der Grundschule das Programmieren näherbringen kann. Dabei ist es ihr besonders wichtig, bei Jungen und Mädchen gleichermaßen Interesse und Motivation für Themen der Informatik zu wecken. Die Schülerinnen und Schüler sollen verstehen, dass Computer keine mystischen Zauberkisten sind – sie führen nur das aus, was ein Mensch programmiert hat. Und ganz wichtig: das können die Kinder auch selbst tun und so die digitale Welt mitgestalten!

Sie betont, dass es beim Programmieren um mehr geht als das Schreiben von Code. Vielmehr geht es darum, Aufgaben und Probleme so aufzubereiten, dass der Computer helfen kann, diese zu bewältigen. Zuerst überlegt man sich genau, was das Programm am Ende tun soll und formuliert dann einen Algorithmus, wie man dieses Ziel erreicht. Erst dann schreibt man den Code. Die Denkprozesse, die hierbei beteiligt sind, werden oftmals als *computational thinking* zusammengefasst. Doch wie kann man sich das in der Grundschule vorstellen?

Was ist ein Algorithmus?

Ein Algorithmus ist eine Reihe von Anweisungen, mit denen ein Problem gelöst oder eine Aufgabe ausgeführt werden kann. Dies wird im Unterricht zunächst *unplugged* – ohne Computer – behandelt.

Programmiert wird in einem ersten Schritt die Lehrkraft. Diese spielt einen Roboter und soll kleine Aufgaben im Klassenzimmer erfüllen, zum Beispiel das Fenster öffnen,

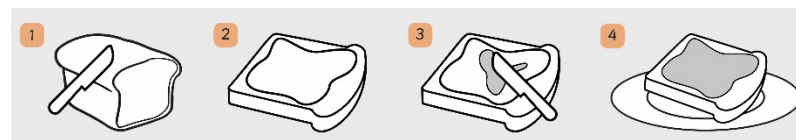


„Mach das Fenster auf!“ – so einfach ist es nicht, der Roboter bewegt sich kein Stück. „Geh geradeaus!“ – der LehrerBot setzt sich in Bewegung und läuft so lange geradeaus, bis er die Wand erreicht – „STOOOP! Nicht so weit!“. Die Kinder begreifen schnell, dass jeder Schritt in einem Algorithmus genau und eindeutig formuliert werden sowie vom Roboter ausführbar sein muss. Größere Vorgänge müssen dabei in Teilschritte zerlegt werden. Es wird außerdem herausgearbeitet, wo ihnen Algorithmen in ihrem Alltag begegnen, zum Beispiel in Form von Wegbeschreibungen, Bastelanleitungen oder Rezepten.

Algorithmen in natürlicher Sprache darstellen

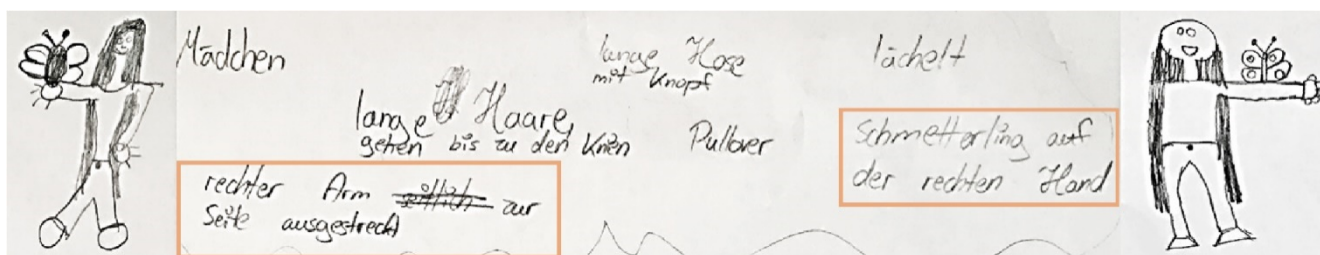
Es gibt ganz unterschiedliche Möglichkeiten, einen Algorithmus darzustellen. Das Beschreiben von Abläufen in Alltagssprache haben die Schülerinnen und Schüler schon beim Programmieren des *LehrerBot* kennengelernt. In verschiedenen Aufgaben wird nun geübt, Abläufe in natürlicher Sprache zu beschreiben und Anweisungen präzise und eindeutig zu formulieren.

Die Kinder wandeln beispielsweise eine bildliche Anleitung, wie das Schmieren eines Marmeladenbrotts, in sprachliche Befehle um.



© TU München, Katharina Geldreich

Im Idealfall werden die sprachlichen Anweisungen auch ausgeführt. Hierbei fällt zumeist schnell auf, dass es gar nicht so einfach ist, Befehle so zu formulieren, dass jeder das gleiche darunter versteht. Hinzu kommt: Auch wenn Anweisungen nicht ganz eindeutig formuliert sind, können wir Menschen meist aus dem Zusammenhang schließen, was gemeint ist. Bei Computern ist das anders.



© TU München

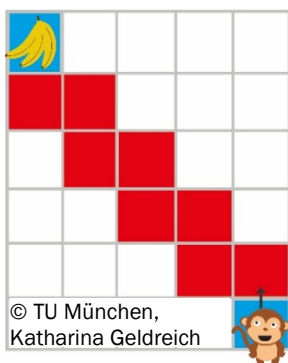
Genaues Formulieren und Lesen

In einer weiteren Übung malt jedes Kind oben auf ein Blatt Papier ein beliebiges Bild. Alle Blätter werden danach nach rechts weitergereicht. Unter dem Bild sollen die Schülerinnen und Schüler nun beschreiben, was sie sehen. Im Anschluss wird das Bild nach hinten umgeknickt und die Blätter werden erneut nach rechts weitergegeben. Nun malt jeder unter die Beschreibung ein dazu passendes Bild – spicken ist natürlich nicht erlaubt!

Am Schluss wird verglichen wie ähnlich sich das Ausgangs- und das Endbild sind. Im Beispiel oben fällt auf, dass die beiden Mädchen spiegelverkehrt sind und der Schmetterling an einer anderen Stelle sitzt. Woran liegt's – wurde nicht genau formuliert oder nicht genau gelesen? In diesem Fall sowohl als auch!

Hin zur Programmiersprache

Etwas eindeutig in Alltagssprache zu beschreiben ist eine Herausforderung. Um Anweisungen für den Computer zu schreiben gibt es deswegen Programmiersprachen, die einer festgelegten Syntax folgen. Wenn man einen Algorithmus in einer Sprache darstellt, die der Computer versteht, spricht man von einem Programm. Programme sagen dem Computer sozusagen, was er zu tun hat.



© TU München, Katharina Geldreich

Um den Umgang mit vorgegebenen Befehlen einzuüben, lösen die Schülerinnen und Schüler verschiedene Aufgaben in einem Parcours. Im Beispiel links programmiert man den Weg des Affen zu seinem Futter. Hierfür verwenden sie drei vorgegebene

Befehle in Symbolform: *Schritt vorwärts*, *Vierteldrehung links* und *Vierteldrehung rechts*:



© TU München, Katharina Geldreich

Optimal ist, wenn die Kinder ihre Lösungen in einem Feld selbst ablaufen können. Hierzu eignen sich beispielsweise quadratische Teppichfliesen sowie Straßenmarkierungen oder Kreppband zum Markieren der Felder. Alternativ können die Schülerinnen und Schüler ihre Lösungen auch mit kleinen Figuren (z.B. LEGO-Männchen) auf ihren Arbeitsblättern ablaufen.

Scratch unplugged

Nach dem Beschreiben von Algorithmen in Alltagssprache und mit Symbolen lernen die Kinder schließlich die visuelle Programmiersprache SCRATCH kennen. SCRATCH besteht aus Programmblöcken, die wie Puzzleteile ineinanderpassen und miteinander kombiniert werden müssen.



© Grundschule Obertheres

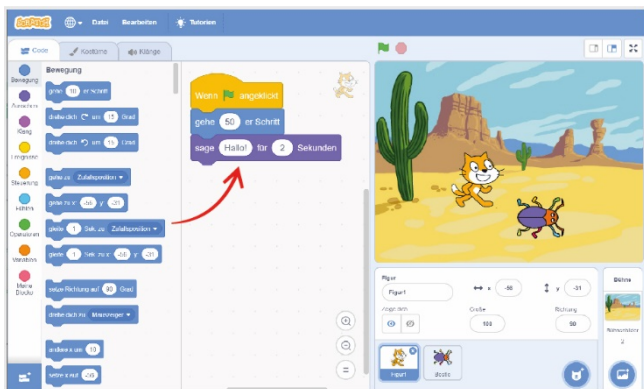
Um die Schülerinnen und Schüler nicht zu überfordern, werden die Programmblöcke und algorithmischen Strukturen zunächst analog eingeführt. Dazu werden haptische Programmblöcke verwendet, die genauso aussehen wie die SCRATCH-Befehle. Diese sind mit Magneten und Klettverschlüssen ausgestattet, sodass mit ihnen sowohl an der Tafel als auch auf Filzbahnen programmiert werden kann.

Im Gegensatz zum Programmieren mit den Symbolen können in SCRATCH nicht nur Sequenzen (mehrere Anweisungen hintereinander) programmiert werden, sondern auch Wiederholungen und bedingte Anweisungen. Diese algorithmischen Grundstrukturen werden nach und nach anhand verschiedener Aufgaben im Parcours eingeführt.

Programmieren am Computer

Nach den „Trockenübungen“ programmieren die Kinder am Computer. Programmiert werden in SCRATCH einzelne Figuren – auf diesem Wege kann man z.B. Geschichten oder Spiele programmieren.

Unter <https://scratch.mit.edu> kann man SCRATCH kostenlos herunterladen oder im Browser verwenden.



© Lifelong Kindergarten Group, MIT Media Lab

Die SCRATCH-Oberfläche besteht aus verschiedenen Bereichen. Links in der *Blockpalette* findet man sämtliche Programmblöcke. Diese kann man nach rechts in den *Programmierbereich* schieben und dort miteinander kombinieren. Rechts befindet sich die *Bühne* – hier führen die Figuren das aus, was man für sie programmiert hat.

Die Schülerinnen und Schüler bearbeiten einen Lernzirkel, in dem die Grundfunktionen von SCRATCH nacheinander behandelt werden. Ausgehend von Fragen der Bedienung führen die Stationen über einfache *Sequenzen* hin zu Kontrollstrukturen wie *Wiederholungen* und *bedingte Anweisungen*. An jeder Station wird die entsprechende Funktion zunächst schrittweise erklärt – die Schülerinnen und Schüler programmieren jeden Schritt an ihrem eigenen Rechner nach.

2. Sprechen

Der Zirkusdirektor sagt etwas

- 1 Öffne den Ordner „Zirkel“. Öffne dort die Datei „2. Sprechen“.
- 2 Klicke auf Aussehen!
- 3 Schiebe die Kachel insgesamt drei mal nach rechts. So soll es aussehen:

- 4 Klicke auf die weißen Felder und ändere den Text des Direktors:
- 5 Klicke auf die grüne Flagge um das Programm zu starten!
- 6 Klicke auf „Datei“ und wähle „Speichern“ aus!
- 7 Du kannst das Fenster jetzt schließen.

© TU München, Katharina Geldreich

Im Anschluss daran bearbeiten sie eine dazu passende Aufgabe: das eingeführte Programm muss je nach Station angepasst, erweitert oder in einem anderen Kontext verwendet werden.

Eigene Ideen umsetzen

Der letzte Schritt im Unterrichtskonzept führt die Schülerinnen und Schüler hin zur Planung und Umsetzung eigener Programmideen – sozusagen die Königsklasse des Programmierens. In Einzel- oder Partnerarbeit überlegt sich jedes Kind eine kurze Geschichte, beschreibt diese in einem Drehbuch und setzt sie schließlich in Scratch um. Im Anschluss stellen sich die Kinder ihre Programme gegenseitig vor.

Fortbildung von Lehrkräften

Ihr Unterrichtskonzept hat die Arbeitsgruppe der TU München über zwei Jahre hinweg mit Schulklassen der dritten und vierten Klassenstufe erprobt, evaluiert und weiterentwickelt. Seit Beginn 2018 bilden sie im Projekt *AlgoKids – Algorithmen für Kinder* Grundschullehrkräfte weiter und untersuchen, wie viel und vor allem welche Weiterbildung diese benötigen, um die neuen Themen selbstsicher und kompetent in ihrem Unterricht zu behandeln. Thema ist außerdem, zu welchen Fächern sie Anknüpfungspunkte sehen.

Rückmeldungen aus dem Schulalltag

Insgesamt beteiligen sich 20 Grundschulen aus ganz Bayern am Projekt *AlgoKids*. Die Rückmeldung der Lehrkräfte ist dabei sehr positiv, selten hätten sie eine derart hohe Motivation und Begeisterung bei ihren Kindern gesehen. Das gilt sowohl für Jungen als auch für Mädchen. Natürlich gibt es auch Schwierigkeiten – diese sind überwiegend „administrativer“ Natur. Die technische Ausstattung an den Schulen ist sehr unterschiedlich und oftmals mangelt es an technischem Support. Darüber hinaus fehlt es im Schulalltag oft an Zeit für das zusätzliche Thema. Die 40 Lehrer im Projekt sind sich dennoch einig – das Programmieren sollte längerfristig Einzug in die Grundschule halten!

Einblick in die Erfahrungen der Lehrkräfte und Materialien zum Download gibt es Anfang 2020 unter www.edu.tum.de/ddi.

DIE AUTORIN

Katharina Geldreich, Wissenschaftliche Mitarbeiterin an der TU München