



Medien und Informatik (MI) für PS

Frag Fred

Was ist eine Programmiersprache?

04:32 Minuten

Autorinnen Deborah Troxler und Isabelle Rohrer

Zusammenfassung

Smartphones, Anzeigetafeln, elektrische Zahnbürsten – die allermeisten elektrischen Geräte funktionieren nur dank der Software, also einem Computerprogramm. Die Software mit einer Programmiersprache am Computer programmiert. Doch was genau ist eine Programmiersprache und wie kommunizieren wir Menschen mit Maschinen? Welche Sprache versteht der Computer? Roboter Fred erklärt Sandro und Celine, wie man mit Hilfe einer Programmiersprache einem Computer Befehle erteilen kann, was Maschinen- und Quellcode sind und wie Programmiersprachen aufgebaut sind.

Programmierer müssen ihre Anweisungen an den Computer exakt, eindeutig und fehlerlos schreiben, denn sonst versteht der Computer nicht, was er tun muss. Dafür gibt es Programmiersprachen. Es gibt viele verschiedene Programmiersprachen mit unterschiedlicher Komplexität. Für den Schulunterricht wurde die Programmiersprache «Scratch» entwickelt. Damit werden die SuS auf spielerische Art und Weise mit dem Programmieren vertraut gemacht. Die Arbeit mit «Scratch» fördert nicht nur das systematische Denken, sondern auch die Kreativität und Problemlösungskompetenzen.

Didaktik

Dieser Beitrag und die dazugehörigen Arbeitsblätter eignen sich für den Einsatz auf der Mittelstufe, insbesondere ab der 5. Klasse.

Folgende Themen/Fragestellungen gliedern das Unterrichtsmaterial:

- Einstieg Programmiersprachen
- Programmieren Unplugged
- Programmiersprache «Scratch»

Damit die Lektionen mit möglichst vielen Eigenaktivitäten und wenigen Stolpersteinen durchgeführt werden können, empfiehlt sich folgendes Vorgehen:

- Die SuS führen die Aufgaben mehrheitlich in Partnerarbeit durch. Ein Kind programmiert und das andere Kind sitzt daneben und unterstützt mündlich (darf aber nicht mit der Maus oder Tastatur eingreifen). Nach einer vereinbarten Zeit werden die Rollen getauscht.
- Die SuS arbeiten so viel wie möglich selbstständig probieren aus («Trial and Error»). Einfache Anleitungen, Memos oder das gegenseitige Helfen fördern selbstständiges Lernen. Die Lehrperson unterstützt beratend.

Ergänzend kann der Beitrag [«Ada und die Programmiererinnen»](#) von SRF mySchool gesichtet werden, als Abschluss der ganzen Unterrichtsreihe oder in Zusammenhang mit der Aufgabe 2.

Der Inhalt dieser Unterrichtsreihe ist Teil des Lehrplans 21.

Lernziele Die Lernenden...

1. wissen, dass beim Programmieren die Eindeutigkeit und Genauigkeit von Anweisungen unabdingbar ist.
2. wissen, dass Programmiersprachen aus Codes bestehen.
3. kennen und verstehen die wichtigsten Begriffe zum Thema Programmiersprachen.
4. wissen, wie Blockprogrammierung funktioniert und können diese analog anwenden.
5. kennen die Programmiersprache «Scratch» und können damit einfache Ideen umsetzen.

Bezug zu Lehrplan 21

Medien und Informatik (MI 2.2.d)

Die Schülerinnen und Schüler können einfache Abläufe mit Schleifen, bedingten Anweisungen und Parametern lesen und manuell ausführen.

Medien und Informatik (MI 2.2.e)

Die Schülerinnen und Schüler verstehen, dass ein Computer nur vordefinierte Anweisungen ausführen kann und dass ein Programm eine Abfolge von solchen Anweisungen ist.

Medien und Informatik (MI 2.2.f)

Die Schülerinnen und Schüler können Schleifen, bedingte Anweisungen und Parameter in selbst geschriebenen Computerprogrammen anwenden.

Methodisches Vorgehen

Das hier beschriebene methodische Vorgehen und die darauf verweisenden Arbeitsblätter ergeben zusammen eine komplette Unterrichtseinheit zum Thema «Programmiersprachen». Die Aufgaben und Aufträge lassen sich aber auch einzeln einsetzen.

Für einige Aufgaben werden Laptops oder Tablets benötigt.

Einstieg

Aufgabe 1: Zeichnungsdiktat

Die SuS erfahren mit den Aufgaben 1a und 1b, dass die Vermittlung von präzisen Anweisungen elementar für das Programmieren ist. In Zweiergruppen übernehmen die SuS jeweils abwechselnd die Rolle des Programmierers / der Programmiererin und des Computers, der die Befehle ausführt. Dem abschliessenden Austausch über den Code soll genügend Zeit eingeräumt werden. Hier empfiehlt es sich auch den Austausch im Plenum durchzuführen. Welche SuS haben besonders kreative, originelle, kurze, etc. Codes geschrieben?

Aufgabe 2: Begriffe

1. Die Klasse schaut den Filmbeitrag «Programmiersprachen» im Plenum.
2. Die LP hat vorgängig die Begriffskarten auf Papier ausgedruckt und kopiert. Zu zweit erhalten die SuS eine Begriffskarte. Je nach Klassengrösse müssen einige Zweiergruppen die gleichen Begriffe erklären.
3. Jede Zweiergruppe schreibt zu ihrem Begriff eine Definition auf die Rückseite der Karte. Dazu schauen die SuS individuell den Beitrag nochmals. Auf Wikipedia finden die SuS weitere Informationen zu ihrem Begriff. Es gilt zu beachten, dass die Antworten der SuS eventuell von den Lösungen abweichen.
4. Die LP sammelt alle Begriffskarten wieder ein und legt diese in eine Box. Für die Aufgabe 3 werden diese wieder benötigt.

Hinweis zu Aufgabe 2

Die Begriffe «Programmiersprache Ada» und «Quellcode» werden im Filmbeitrag nur kurz besprochen. Eine Recherche auf Wikipedia ist deshalb unverzichtbar. Weitere Recherchemöglichkeiten sind bei den Lösungen aufgelistet. Die LP kann die Begriffskarten gezielt den einzelnen Gruppen zuweisen und so auf das Lernniveau der SuS Rücksicht nehmen.

Programmieren Unplugged

Aufgabe 3: Blockprogrammierung

Die Aufgabe 3a ist textlastig. Je nach Lernniveau der Klasse kann die LP das Arbeitsblatt aushändigen oder die Aufgaben mündlich Schritt für Schritt erklären. Bevor die Aufgabe gelöst werden kann, haben die SuS sich mit der Aufgabe 2 und den Begrifflichkeiten auseinandergesetzt. Die LP hat eine Box mit allen Definitionskarten griffbereit hingestellt (siehe Aufgabe 2). Es empfiehlt sich, nicht alle Gruppen im selben Raum arbeiten zu lassen.

Aufgaben 3b–3f:

Bevor die SuS die zweite Aufgabe lösen, stellt die LP die Befehlskarten vor und stellt sicher, dass alle SuS die Befehle verstehen.

Die Aufgabe 3d schneidet die Themen «Schleifen» und «Debugging» an. Mit «Schleifen» sind Wiederholungen von Anweisungen gemeint, so dass die Blockprogrammierung einfacher und kürzer wird.

Aufgabe 4: «Unplugged2Scratch»

Mit dieser Aufgabe wird versucht, eine Verbindung zwischen analogen Befehlsbausteinen und «Scratch»-Befehlsbausteinen herzustellen. In diesem Sinne dient die Aufgabe 4 als Einstieg in die Programmiersprache «Scratch». Das Programm selbst ist erst in der Aufgabe 5 Thema.

Ablauf:

1. Die LP hat die Aufgabe 4 soweit vorbereitet, dass sie den SuS die Befehlsbausteine von «Scratch» (Skripte) am Beamer zeigen kann.
2. Die Programmierbausteine sind thematisch (und farblich) geordnet. Die LP stellt den SuS diese schrittweise vor. Es werden ausgewählte Bausteine erklärt. Danach wird verglichen, welche Bausteine gleich oder ähnlich sind, wie jene von Aufgabe 3.

Programmiersprache Scratch

Aufgabe 5: Scratch auf einen Blick

Diese Blätter sind ausschliesslich für die LP gedacht. Der Inhalt entspricht einer Übersicht und zeigt, wie man diese Programmiersprache im Unterricht einsetzen kann.

Alle unterstrichenen Begriffe sind verlinkt. Wenn die Blätter ausgedruckt werden, sind diese Verweise nicht sichtbar.

Weitere Informationen zu «Scratch»:

- <https://scratch-dach.info/wiki/Scratch>

Unterrichtseinheiten zu «Scratch»:

- http://scratched.gse.harvard.edu/sites/default/files/scratchchindrestunden_1.pdf
- <https://mia.phsz.ch/Informatikdidaktik/ScratchMaterial>

Bezug zu Lehrmittel

- inform@21: Kapitel «Programmieren»
- connected1: Kapitel 4 «Vom Alltagscode zum digitalen Code»