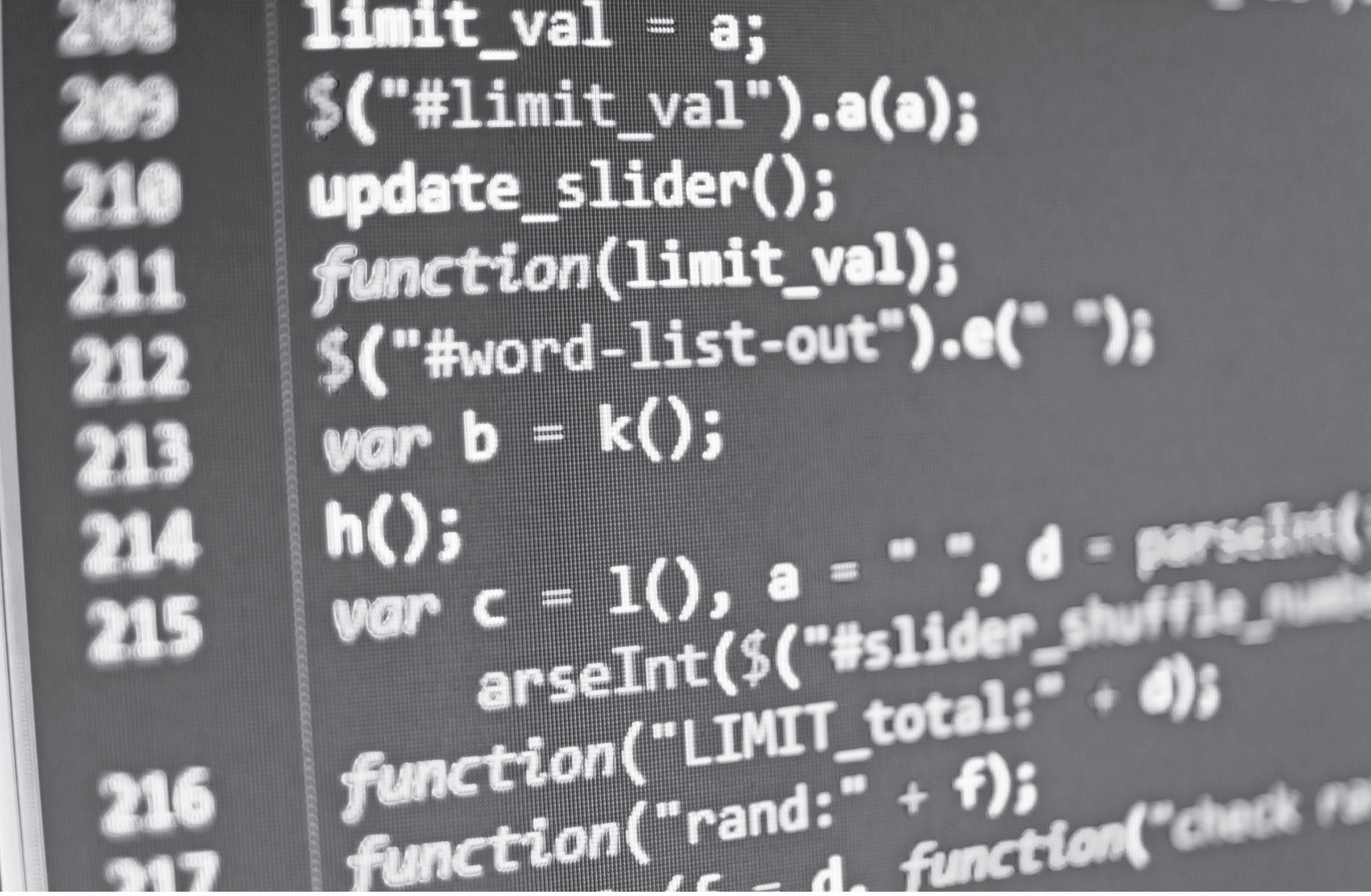

21st Century Skills

Programmieren als neue Kulturtechnik?

Von Stefan Aufenanger

Glaubt man der (Bildungs-)Politik, so kann allein die Programmierfähigkeit kommende Generationen davor bewahren, in einer digitalisierten Welt nicht abgehängt zu werden. Nach wie vor erschweren aber Missverständnisse und unklare Definitionen eine konstruktive Diskussion darum, wie sinnvoll eine Integration des Programmierens in schulische Bildung wirklich ist. Eine genauere Untersuchung des Begriffes Programmieren ist daher unumgänglich, und auch ein Rückblick auf erste pädagogische Versuche mit kindgerechter Software sowie die internationale Perspektive helfen dabei, sich dem tatsächlichen Potenzial des Programmierens anzunähern.

Allerorten ist die Rede davon, Kinder sollten programmieren lernen, da dies die neue Kulturtechnik sei und unbedingt notwendig für das Leben und Arbeiten in einer digitalen Gesellschaft. Nicht nur die großen Medienunternehmen oder Vertreter der Industrie, sondern auch Politiker, insbesondere Bildungspolitiker, fordern dies. So hat etwa 2014 der damalige Wirtschaftsminister Sigmar Gabriel in einem Interview mit der Rheinischen Post die Einführung von „Programmiersprachen als zweite Fremdsprache an Schulen" gefordert. Und auch Bundeskanzlerin Merkel hat bei ihrer Eröffnungsrede zur Cebit 2017 in Hannover dies unterstrichen und „Programmieren als Grundfähigkeit neben Lesen, Schreiben und Rechnen" bezeichnet. Nach einer von Bitkom – der Digitalverband Deutschlands – begleitend zur learntec 2017 veröffentlichten Studie[1] wünscht sich jede zweite Lehrkraft (54 %), „dass Informatik und andere Digitalthemen in der Schule einen höheren Stellenwert erhalten, auch wenn dies zu Lasten von Fächern wie Sport, Musik oder Religion gehen sollte" und auch 42 % der Eltern wünschen sich, dass Programmieren unterrichtet werden soll[2]. Ergänzt wird das Ganze von der Gesellschaft für Informatik, die sich für ein eigenständiges Unterrichtsfach Informatik ausspricht. All diese Forderungen machen zwei Dinge deutlich: In der digitalen Welt braucht es neue Kompetenzen, die auch in der Schule vermittelt werden sollten – und Programmieren zu können, scheint eine neue Kulturtechnik zu werden.

Missverständnisse prägen die Diskussion

Die Probleme beginnen jedoch schon bei der Frage, ob Programmieren gleich Informatik bzw. Programmieren gleich Coding ist, ob zum angemessenen Umgang mit der digitalen Welt Programmierfähigkeiten tatsächlich unabdingbar sind, und ob die Schule Programmieren überhaupt so vermitteln

kann, dass es für eine spätere Qualifikation in der digitalen Arbeitswelt ausreicht. Fangen wir bei der ersten Frage an: Dies scheint das größte Missverständnis zu sein, wenn Politiker fordern, Programmieren müsste Unterrichtsfach werden. Informatiker weisen immer wieder darauf hin, dass Programmieren nicht Informatik ist, genauso wenig wie Rechnen die Mathematik präsentiert.

Die Grundannahme für die Forderung nach Aufnahme von Coding, Programmieren, Informatik oder digitaler Bildung in Schule und Unterricht besteht darin, heutige Kinder durch Wissen über die digitale Welt auf dieselbe vorzubereiten – ganz so, wie sie in Biologie die Photosynthese, in Physik die grundlegenden Kräfte der Welt wie etwa Elektromagnetismus und Gravitation oder in Chemie den Aufbau des Periodensystems erlernen sollen. Doch ist das so stimmig? Natürlich ist es wichtig, dass Schülerinnen und Schüler lernen, die Welt zu verstehen. Jedoch ändert sich die digitale Welt so schnell, dass ein Vergleich mit den klassischen naturwissenschaftlichen Fächern nicht so einfach ist. Wer meint, dass heute das Wissen um Algorithmen ein Kern informatischer Bildung sei, übersieht das Aufkommen neuronaler Netzwerke und künstlicher Intelligenz, deren Durchschaubarkeit immer schwieriger wird. In diesem Sinne ist in der Schule sicher mehr zu tun, als den Lernenden nur das Programmieren beizubringen. Jedoch können dabei exemplarisch die Herausforderungen der digitalen Welt verdeutlicht werden. So sind auch die Beiträge in diesem Heft zu verstehen, die allerdings weniger aufs Programmieren an sich fokussieren als auf die Möglichkeiten von Schule, Kindern und Jugendlichen das Programmieren, und alles, was damit zusammenhängt, nahezubringen.

Aber noch einmal zurück: Ist Programmieren und Coding das Gleiche, und warum sollten Kinder und Jugendliche Programmieren lernen? Der Gebrauch der Begrifflichkeiten ist recht unterschiedlich. Manche setzen beide Begriffe gleich, manche sehen Programmieren als den übergeordneten Begriff, der die Umsetzung der Lösung einer Aufgabe in eine Programmstruktur beschreibt, während Coding als die Erstellung konkreter Programmzeilen verstanden wird, also die Konkretisierung des Algorithmus in ein ausführbares Programm. Programmieren meint also eher das systemische und strukturelle Denken zur Entwicklung einer Anwendung. Jedoch hat sich der Begriff Coding, bedingt durch Tendenzen in den USA, auch bei uns als das Lernen von Programmieren durch Kinder durchgesetzt und ist in diesem Sinne auch für den Bereich der Schule, jedenfalls der Grundschule, zunehmend gebräuchlich.[3] Da im deutschsprachigen Raum beide Begriffe aber häufig austauschbar verwendet werden, ist auch im Folgenden beides gemeint.

Erste pädagogische Ansätze

Die Ursprünge des Ansatzes, dass schon Kinder Programmieren lernen und sogar dabei Spaß haben können, sind bei dem amerikanischen Erziehungswissenschaftler Seymour Papert zu suchen. Mit seinem Werk „Mindstorms" (Papert 1980) hat er den Grundstein für die Ausbreitung der von ihm und Kolleg*innen entwickelten Programmiersprache Logo gelegt. Logo war so einfach strukturiert, dass vor allem Kinder sehr gut damit umgehen konnten. Bekannt wurde Logo durch seine Schildkröte, die man durch entsprechende Anweisung auf dem Bildschirm Grafiken malen lassen konnte. Aus diesem Ansatz heraus wurden dann am MIT in Cambridge/USA, an dem Papert lehrte und forschte, die Lego Mindstorms entwickelt, eine programmierbare Einheit, mit der sich durch Verbindung mit Legosteinen Roboter und Fahrzeuge gestalten ließen, die bei entsprechender Programmierung eine Vielzahl von Aktionen ausführen konnten. Auch die Programmanwendung Scratch als visuelle Sprache wurde dort, aufbauend auf Logo, entwickelt. Beides – Lego Mindstorms und Scratch, letzteres vor allem in der Weiterentwicklung von ScratchJr. – sind in Grundschulen sehr beliebt, da sie auch ohne Lesefähigkeiten benutzt werden können. Was den Ansatz von Papert aber weiterhin auszeichnet, ist die Einbettung des Computers, und damit auch des Programmierens, in einen konstruktivistischen Ansatz. Dies hat er als ehemaliger Mitarbeiter des prominenten Entwicklungspsychologen Jean Piaget in einem weiteren Buch „Revolution des Lernens" (Papert 1994) ausgeführt und dabei konstruktivistisches Lernen für den Unterricht empfohlen. Schon 1994 hat Seymour Papert in einem SPIEGEL-Interview auf die notwendigen Veränderungen von Schule hingewiesen, wenn Computer dort Einzug halten sollen.[4]

Neben Logo hat aber auch Basic in der Schule eine Rolle gespielt, vor allem in den unterschiedlichen Ausformungen der „Informationstechnischen Grundbildung", die in den 1980er-Jahren meist ab der Sekundarstufe I den Lernenden neben dem Aufbau eines Computers auch Ansätze des Programmierens beibringen sollte. Basis war allein schon deshalb beliebter als Logo, da es manchmal als Grundausstattung der damaligen Computer mit dem MS-DOS-Betriebssystem von Microsoft ausgeliefert wurde bzw. darauf auch sehr gut lauffähig war. In der Oberstufe konnten dann in manchen Bundesländern die mit Basic gelegten Grundkenntnisse in höheren Programmiersprachen vertieft werden.

Neuerdings wird zur Motivierung von Schülerinnen und Schülern gern auf Programmsprachen zurückgegriffen, mit denen Apps für Smartphones und Tablets entwickelt werden können. Da heutige Jugendliche mit diesen Geräten gut ausgestattet sind, wird das Programmieren von Apps für sie interessant. Der MIT-Inventor ist ein solches Programm, mit dem, aufbauend auf und auch ähnlich strukturiert wie Scratch, ganz einfache Apps für Android entwickelt werden können. Nicht zuletzt muss aktuell auf Calliope hingewiesen werden, eine programmierbare Platine fürs Experimentieren und Programmieren in der Grundschule.

Wo Deutschland steht

In Deutschland lassen sich mehrere Diskussionsstränge zum Thema Programmieren im Unterricht finden. Zum einen gibt es eine bildungspolitische Diskussion, die – wie schon erwähnt – auch auf höchster politischer Ebene geführt wird, aber übersieht, dass Bildung Ländersache ist und deshalb die notwendigen Entscheidungen entweder in den einzelnen Bundesländern getroffen werden oder in der Kultusministerkonferenz der Länder (KMK). Mit dem Strategiepapier „Bildung in der digitalen Welt" (KMK 2016) wurde ein wichtiger Schritt getan, digitale Kompetenzen in Schule zu vermitteln, sollen doch nach dem Wunsch der Kultusminister der Länder alle Lernenden ab dem Schuljahr 2018/19 im Laufe ihrer Schulkarriere sechs Kompetenzbereiche erwerben können, die einen selbstbestimmten und sicheren Umgang mit digitalen Medien gewährleisten. Darunter findet sich auch, unter dem fünften Kompetenzbereich „Problemlösen und Handeln", ein Unterpunkt „Algorithmen erkennen und formulieren" (s. auch S. 16–18).

Zwar wird die Fähigkeit zum Programmieren nicht konkret angesprochen, aber die Erstellung algorithmischer Struktu-

ren sowie die Planung von algorithmischen Sequenzen ist die abstrakte Beschreibung davon. Somit kann damit gerechnet werden, dass in den nächsten Jahren das Thema Programmieren auch in deutschen Schulen verstärkt Einzug halten wird, wofür auch die Gesellschaft für Informatik schon lange kämpft.[5] (Zum tatsächlichen Sinn eines Pflichtunterrichtfachs Informatik gibt es indes im SPIEGEL einen treffenden Kommentar von Sascha Lobo[6]).

Es wird aber auch diskutiert, ob Informatik allein überhaupt ausreicht, auch wenn diese Disziplin wichtige Grundlagen zum Verständnis digitaler Medien legt. Aus der Medienpädagogik kommt die Forderung, Medienbildung stärker in den Vordergrund zu rücken und die informatischen Kompetenzen des Unterrichtsfachs Informatik mit den Kompetenzen aus der Medienpädagogik zu verbinden. Wie dies aussehen könnte, ist in der Dagstuhlerklärung sowie in dem Buch von Beat Döbeli Honegger (2016) sehr gut erläutert. Danach wird nicht mehr von Informatik als Unterrichtsfach gesprochen, sondern von „Bildung in der digitalen vernetzten Welt (kurz: Digitale Bildung)" die aus „technologischer, gesellschaftlich-kultureller und anwendungsbezogener Perspektive in den Blick genommen werden (muss)". Diese sollen in einem eigenständigen Lernbereich und in allen Fächern aufgegriffen und auch unter fachdidaktischen Aspekten angesprochen werden. Notwendig ist dazu natürlich, dass in der Lehrerbildung entsprechende Voraussetzungen durch eine Verbindung von Informatik und Medienbildung geschaffen werden. Programmieren wird in diesem Konzept gar nicht angesprochen, taucht aber indirekt in den einzelnen Perspektiven auf. Denn mit der Fähigkeit, programmieren zu können, lassen sich technologische Konzepte digitaler Systeme verdeutlichen, gesellschaftliche Folgen von Algorithmen vor Augen führen und konkrete Anwendungen erproben. Mit der Dagstuhl-Erklärung (s. auch S. 42–43) ist eine realistische Perspektive für digitale Bildung in Schule und Unterricht auf bildungspolitischer Ebene eröffnet worden, die verbunden mit dem KMK-Strategiepapier „Bildung in der digitalen Welt" eine zukunftsträchtige Vorbereitung junger Menschen auf die digitale Welt ermöglichen wird.

Außerhalb der Schule

Was bei dem Blick auf das Bildungssystem aber nicht übersehen werden darf, ist der außerschulische Bereich, der Eltern anbietet, ihre Kinder in Programmierkurse oder -camps zu schicken. Und auch im Internet steigen die Angebote von Onlinekursen, die die Grundlagen des Programmierens vermitteln wollen. Diese richten sich vor allem an jene Familien, die ihre Kinder schon früh fit für die digitale Gesellschaft machen wollen und dabei unterstellen, dass die Vermittlung der entsprechenden Kenntnisse in frühen Jahren zu einer späteren Qualifikation führen kann. Dabei wird auf gängige Anwendungen wie etwa Scratch oder ScratchJr. Bezug genommen, die erstens kostenlos und zweitens weitverbreitet sind. Ob diese beiden Programme zu lernen unbedingt ein Workshop oder Camp verlangt, sei dahingestellt. Der Vorteil dürfte darin liegen, dass hier gemeinsam in einer sozialen Gruppe Projekte entwickelt und umgesetzt werden. Dies macht ihre Attraktivität aus, und daran sollte sich auch Schule messen.

Aus internationaler Perspektive

Beim Blick ins internationale Umfeld zeigt sich, dass in einigen Ländern Programmieren schon länger fester Bestandteil des Unterrichts ist. In **Großbritannien** wurde 2014 im Nationalen Curriculum das bisherige Fach ICT (Information Communication Technology) durch „Computing" ersetzt und ausgebaut: Alle Schülerinnen und Schüler ab Klasse 1 werden in diesem neuen Fach unterrichtet, das zum Verständnis von Computersystemen und zum „computational thinking" beitragen soll. Mit Letzterem sollen die Schülerinnen und Schüler lernen, wie man Probleme löst, wie Computersysteme gestaltet werden und wo die Grenzen der menschlichen als auch der künstlichen Intelligenz liegen. Dies soll auch auf die spätere, digitalisierte, Arbeitswelt vorbereiten.[7] Programmieren wird zwar in diesem Fach auch praktiziert, steht aber anscheinend nicht im Zentrum. Es ist eher als Motivierung für das Fach gedacht und soll Anwendungsbereiche von „Computing" deutlich machen. Ergänzt wird das Ganze auch durch die Fähigkeit, Informationen zu suchen und zu verarbeiten, also Informationskompetenz zu entwickeln. Das Curriculum für die Grundschule ist sehr ausführlich und differenziert aufbereitet und gibt konkrete Anregungen für die Lehrkräfte zur Gestaltung ihres Unterrichts. Insgesamt enthält das Curriculum vier Kompetenzstufen – so genannte ‚key stages'[8] – die genau beschreiben, was die Lernenden am Ende jeder Stufe wissen und können sollten.

In anderen europäischen Ländern sind ähnliche Tendenzen zu erkennen, wobei manche schon früher angefangen haben, digitale Kompetenzen in der Schule zu vermitteln, wie etwa in **Finnland**, oder dies fester Bestandteil des Unterrichts ist, wie etwa in **Frankreich** (Grass und Weber 2016).

In den **USA** spielt die Webseite code.org eine bedeutsame Rolle, eine non-profit-Organisation, die aber von den großen Medienunternehmen wie etwa Microsoft, Apple, Google und Facebook finanziell unterstützt wird. Die indirekt damit verbundene Intention, zukünftige Arbeitskräfte zu gewinnen, wird vor allem durch das Statement von Apple-CEO Tim Cook deutlich, der dem neuen amerikanischen Präsidenten Donald Trump auf den Weg gab: „Coding should be a requirement in every public school." Ziel von code.org ist es, dass in jeder amerikanischen Schule „computer science" unterrichtet wird. Dazu werden kostenlos Kurse für Kinder und Jugendliche angeboten, und es gibt jedes Jahr eine große Aktion unter dem Motto „Hour of Code", an der meist prominente Politiker, wie etwa Barack Obama, teilgenommen haben. Der Einfluss der Unternehmen auf schulische Curricula wird aber auch kritisch gesehen, da in manchen US-Bundesstaaten die Einführung von Programmunterricht an Schulen häufig mit der Notwendigkeit begründet wird, die Arbeitswelt benötige in Zukunft Menschen mit Programmierkenntnissen, und Curricula dafür zusammen mit Unternehmen entwickelt werden sollten. Da kommen selbstverständlich pädagogische Interessen und der Respekt gegenüber der Schule als eigenständige Institution zu kurz. Um trotzdem Anregungen für Lehrkräfte zu geben, werden für unterschiedliche Schulstufen Unterrichtsmaterialien zur Verfügung gestellt. Zusätzlich stellt die Webseite eine Evaluationsstudie vor, die sehr ausführlich das methodische Vorgehen und erste Ergebnisse darstellt. Weltweit sind auf der Webseite von code.org ca. 20 Millionen Teilnehmende eingetragen. Die Webseite ist in vielen Sprachen, darunter auch auf Deutsch, zugänglich. In Deutschland wird dazu das Motto „Jeder Schüler in jeder Schule sollte die Möglichkeit haben, Informatik zu lernen" ausgegeben (wobei zu fragen ist, ob Informatik das gleiche bedeutet wie das englische „computer science").

In **Süd-Korea** gibt es schon länger schulische Angebote zum Programmieren. Die meisten Schulen bieten bisher „Coding" als außerunterrichtliche, von externen Lehrkräf-

ten unterrichtete, Aktivität an, die bei Lernenden beliebt ist. Ab 2018 wird „Software" ein Pflichtfach in der Mittelschule (Kinder im Alter von ca. 13–15 J.). Meist wird Coding anhand von Scratch gelehrt. Neben dem schulischen Bereich boomen Privatschulen, wohin Kinder geschickt werden, um Programmieren zu lernen.

Programmieren als Grundfertigkeit

Eine grundlegende Frage bei der Forderung, Kinder sollten in der Schule sowie noch früher Programmieren lernen, ist die, was sie wirklich dabei lernen. Im letzten Jahrzehnt hat sich dazu neben den Begriffen Programmieren und Coding (s. o.) auch der des „Computational Thinking" durchgesetzt. Er soll präziser beschreiben, welche Denkvorgänge beim Erstellen von Algorithmen notwendig sind. Nach Jeanette Wing (2006), die den Begriff eingeführt hat, ist darunter die Fähigkeit zu verstehen, ein Problem zu formulieren und dessen Lösung so auszudrücken, dass ein Computer diese ausführen kann. Es ist also kein technologisches Konzept, sondern ein Prozess des Denkens. Interessanterweise hat Weng diesen Prozess nicht nur für die Lösung von Aufgaben durch einen Computer, sondern auch durch Menschen ausgezeichnet. Damit wird auch deutlich, dass es sich beim Programmieren im allgemeinen Sinn um grundsätzliche Fähigkeiten handelt, die Menschen haben sollten, um Aufgaben oder Probleme anzugehen. Dazu ist strukturiertes und prozessorientiertes Denken notwendig, genauso wie zur Erstellung eines Algorithmus. Das Konzept des „Computational Thinking" wurde in den USA auch entwickelt, um in den STEM-Programmen gezielter informatische Aspekte einzubringen (STEM = science, technology, engineering und mathematics). In der deutschen Version MINT wird mit Mathematik, Informatik, Naturwissenschaften und Technik Programmieren stärker akzentuiert. Inzwischen hat sich jedoch auch dieses Konzept erweitert, nämlich in STEAM, wobei das A für Arts, also Kunst im weitesten Sinne steht – es geht hier auch um Designelemente sowie um Kreativität. Aus dem „Computational Thinking" als Prozess des Problemelösens, der Kreativität von STEAM sowie dem konstruktivistischen Denken von Paperts Ansatz lässt sich nun eine Verbindung herstellen, in die Programmieren eingebettet werden kann. Nur unter der Perspektive eines problem- bzw. aufgabenorientierten Unterrichts können Schülerinnen und Schüler das Programmieren nicht als eine sture Umsetzung von Programmschritten, sondern als die Umsetzung einer Suche und damit verbunden einer Lösung eines Problems in Programmsequenzen verstehen.

Beiträge in diesem Heft

Das Heft knüpft mit der Thematik auch an das Themenheft 105 „Maker Education" von Computer+Unterricht an und ergänzt dies. Aus der Perspektive der Informatik wird die Notwendigkeit eines Pflichtfachs Informatik für alle Schülerinnen und Schüler verdeutlicht und die dazu notwendigen Bedingungen aufgeführt. Dabei wird nicht nur die Notwendigkeit eines gendersensiblen Angebots angesprochen, sondern auch, dass dieses Fach in allen Klassenstufen verortet werden muss. Dass man die Grundzüge des Programmierens auch ohne Computer, ja selbst ohne einen Computerraum, Kindern näher bringen kann, wird in dem Beitrag von Daniel Siebrecht gezeigt. Er betont auch den wichtigen Unterrichtsschritt der Reflexion, denn nur durch das Nachdenken über das Getane können Bildungsprozesse in Gang gesetzt werden. Wie anhand von Rauchmeldern Sicherheitsfragen angesprochen und die gesellschaftliche Dimension informatischer Systeme deutlich gemacht werden kann, spricht Adrian Salomon an. Da in allen Haushalten solche Geräte vorhanden sind, kann das Thema den Schülerinnen und Schülern sehr alltagsnah aufgezeigt werden. In eine ähnliche Richtung geht das Unterrichtsbeispiel von Christian Kleinhanß, in dem Schülerinnen und Schüler ein analoges Spiel kreativ entwickeln, um dann durch die Integration digitaler Medien Einsichten in algorithmische Strukturen zu erhalten. Dass schon Grundschülerinnen und -schüler programmieren können, wird sehr anschaulich von Janina Bistron geschildert. Mithilfe der Anwendung Scratch entwickeln diese eigene Ideen und setzen sie in Programmsequenzen um. Damit bekommen sie eine Vorstellung, wie man programmiert. Die Integration von Programmieren in MINT-Projekte wird in dem Beitrag von Hannah Hoffmann und Martin Uckert deutlich. Dort wird nämlich geschildert, wie man in der Sekundarstufe I, aufbauend auf analoge Befehlssequenzen, Grundkenntnisse im Programmieren mit Lego WeDo sowie den Lego Mindstorms erwerben kann. Unterstützt wird das Projekt von Auszubildenden aus Unternehmen sowie den MINT-Lehrkräften.

Die Beiträge in diesem Themenheft zeigen natürlich nur ausschnittsweise, wie kreativ und interessant Programmieren oder Coding in Schule umgesetzt werden kann. Sie wollen dazu inspirieren, einen zukunftsweisenden Unterricht zu gestalten, der Lernenden jene Kompetenzen vermittelt, die für das digitalisierte 21. Jahrhundert notwendig sind.

Stefan Aufenanger
Professor für Erziehungswissenschaft und Medienpädagogik
Johannes Gutenberg-Universität Mainz
aufenang@uni-mainz.de

Anmerkungen

(1) *https://www.bitkom.org/Presse/Presseinformation/Jeder-zweite-Lehrer-wuerde-gerne-haeufiger-digitale-Medien-einsetzen.html* [2. 8. 2017]
(2) *https://www.bitkom.org/Presse/Presseinformation/Eltern-wuenschen-sich-eine-digitale-Schule-fuer-ihre-Kinder.html* [2. 8. 2017]
(3) *http://www.huffingtonpost.com/kiki-prottsman/coding-vs-programming-bat_b_7042816.html* [2. 8. 2017]
(4) *http://www.spiegel.de/spiegel/print/d-13685014.html* [2. 8. 2017]
(5) *http://www.sn.schule.de/~istandard/docs/bildungsstandards_2008.pdf* [2. 8. 2017]
(6) *http://www.spiegel.de/netzwelt/web/programmieren-in-der-schule-sollen-kinder-programmieren-lernen-kolumne-a-1140928.html* [2. 8. 2017]
(7) *http://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf* [2. 8. 2017]
(8) *https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study* [2. 8. 2017]

Literatur

(1) Döbeli Honegger, Beat (2016): Mehr als 0 und 1. Schule in einer digitalisierten Welt. Bern: Hep-Verlag.
(2) Grass, Karen/Weber, Enzo (2016): EU 4.0 - The debate on digitalisation and the labour market in Europe. IAB-Discussion Paper. Nürnberg. – *http://doku.iab.de/discussionpapers/2016/dp3916.pdf*, [10. 01. 2017]
(3) KMK, Sekretariat der ständigen Konferenz der Kultusminster der Länder in der Bundesrepublik Deutschland (2016): Bildung in der digitalen Welt. Strategie der Kultusministerkonferenz. Berlin, Seiten. – *https://www.kmk.org/fileadmin/Dateien/pdf/PresseUndAktuelles/2016/Bildung_digitale_Welt_Webversion.pdf*; [10. 1. 2017].
(4) Papert, Seymour (1980): Mindstorms. Children, Computer and Powerful Ideas. New York: Basic Books.
(5) Papert, Seymour (1994): Revolution des Lernens. Kinder, Computer, Schule in einer digitalen Welt. Hannover: Heise.
(6) Wing, Jeannette M. (2006): Computational thinking. In: Communications of the ACM, 49, 3, 33–35.