

Konzepte für den Informatikunterricht an Primarschulen und auf der Sekundarstufe I

1. Leitidee und Bildungsziele

Genau wie im Unterricht der Mathematik oder der naturwissenschaftlichen Fächer muss es im Informatikunterricht in erster Linie um die Entwicklung der entsprechenden Denkweise und um das Verständnis für grundlegende Konzepte und Methoden der einzelnen Wissenschaften gehen. Der Informatikunterricht muss sich insbesondere auf solche Ideen, Konzepte und Begriffe ausrichten, die fundamental für die Informatik sind und in anderen Wissenschaften nicht vorkommen. Somit liefert der Informatikunterricht Beiträge, die sich nicht durch den Unterricht anderer Fächer erzielen lassen.

Die Gestaltung des Unterrichts muss so ausgerichtet werden, dass man immer darauf achtet, welche Konzepte in welcher Tiefe in welchem Alter zugänglich sind. Das konstruktive Vorgehen bei der Lösung von Problemen steht im Vordergrund und garantiert in hohem Mass Erfolgserlebnisse, die ein hervorragender Garant für Lernerfolg sind. In diesem Fall wird dies noch durch den hohen Grad an Selbstständigkeit auf dem Weg zum Ziel unterstrichen. Weil die korrekte Funktionalität des Produkts der konstruktiven Arbeit der Kinder und der Jugendlichen meistens von ihnen selber überprüft werden kann und keine Kritik der Lehrpersonen braucht, rutscht die Lehrperson in die angenehme Rolle des Beraters, für den Fall, dass die Lernenden nicht wissen, wie sie weiter vorgehen sollen.

Um die Ziele des Unterrichts auf den entsprechenden Stufen festzulegen, müssen wir uns zuerst klar machen, was für eine Wissenschaft die Informatik ist. (Eine ausführliche Darstellung findet man im Buch „Sieben Wunder der Informatik“ [1].) Die Informatik ist der Automatisierung gewidmet, insbesondere der Automatisierung der intellektuellen Arbeit. In diesem Zusammenhang sprechen wir von Informationsverarbeitung. Im Vordergrund steht der Begriff des Problems. Ein Problem (wie z.B. Sortieren der Namenslisten zur Herstellung des Telefonbuchs, Lösen von linearen Gleichungssystemen oder Minimierung der Lieferkosten von Waren an Kunden) besteht typischerweise aus unendlich vielen Problemfällen, auch Probleminstanzen genannt.

Die Arbeit der Informatikerin oder des Informatikers ist es, einen Algorithmus (eine Lösungsmethode) für ein gegebenes Problem zu finden. Ein Algorithmus zur Lösung eines Problems ist eine endliche Beschreibung einer Vorgehensweise, die in allen unendlich vielen Fällen des Problems erfolgreich zur gewünschten Lösung führt. Die Automatisierung des Lösungsvorgehens kann man in zwei Phasen sehen. Zuerst wird ein Algorithmus gefunden und in einer Metasprache (z.B. auf Deutsch) beschrieben. Im zweiten Schritt wird der Algorithmus mittels einer Programmiersprache dem Computer mitgeteilt, um die Berechnung der Lösung des Problems dem Computer zu überlas-

sen. Die zweite Phase nennen wir oft auch Programmieren im engen Sinn. Programmieren im weiten Sinn deckt beide Phasen der Automatisierung ab.

Der Begriff des Algorithmus ist das fundamentalste Konzept der Informatik und führte zur Gründung der Informatik als selbstständige Disziplin. Der zweite wichtige Beitrag der Informatik ist die Einführung des Konzeptes der Berechnungskomplexität. Für die Informationsverarbeitung gelten quantitative Naturgesetze. Eine Problemstellung können wir immer ansehen als die Berechnung der gewünschten Information (des Wissens) aus vorhandenen Daten. Die Gesetze der Berechnungskomplexität besagen für jedes Problem, wie viel Computerarbeit notwendig und hinreichend ist, um das gegebene Problem zu lösen. Oft ist die Menge der erforderlichen Arbeit grösser als die physikalischen Möglichkeiten des bekannten Universums, und so stossen wir auf die quantitativen Grenzen der Automatisierung. Im Brennpunkt der heutigen Informatik steht somit die Entwicklung von effizienten Algorithmen, falls solche für gegebene Probleme existieren.

Die oben genannten Grundkonzepte der Informatik sind im Primarschulalter und in der Sekundarstufe I nicht in ihrer vollen Abstraktion zugänglich, genauso wenig wie die Quantenmechanik als eines der Grundkonzepte der physikalischen Weltanschauung. Altersgerecht sollte der Unterricht der Informatik auf diesen Stufen auf folgendes Bildungsziel zielen:

1. Das Erlernen des Programmierens im engen Sinn als die Steuerung des Computers oder als die Kommunikation mit den Maschinen. Dies erfordert eine eindeutige Beschreibung der gewünschten Tätigkeit (Vorgehensweise) in einer einfachen Sprache des Computers. Somit wird zusätzlich eine neue Dimension der Kommunikationsfähigkeit der Kinder und der Jugendlichen gefördert.
2. Die Entwicklung der konstruktiven Problemlösungsfähigkeit, indem man für einfache Problemstellungen (überwiegend mit endlich vielen Problemfällen) eine Lösungsstrategie (einen Algorithmus) entwirft und umsetzt.
3. Schulen der modularen Vorgehensweise beim Entwurf von komplexen Systemen (Programmen), die massgebend für den Entwurfsprozess aller technischen Disziplinen ist.
4. Entdecken der Vielfalt der Probleme der Informationsverarbeitung im Alltag und Erlernen, sie mit Fokus auf das Wesentliche grafisch oder mittels der Sprache der Mathematik einfach darzustellen.
5. Einführen der Grundkonzepte des Testens und der Verifikation der Funktionalität der entworfenen und implementierten Programme mit dem Ziel, den gesamten Prozess der Automatisierung von der Problemdarstellung bis zur Umsetzung kennen zu lernen.
6. Verbinden des exakten analytischen Denkens der Mathematik mit dem konstruktiven algorithmischen Denken insbesondere in der Automatisierung der Vorgehensweisen in der konstruktiven Geometrie.

Wie man diese Bildungsziele in einzelnen Altersstufen umsetzen kann, ist das Thema der folgenden Abschnitte.

2. Umsetzung der Bildungsziele in der Primarschule

Die altersgerechte Umsetzung der formulierten Bildungsziele und die Auflistung der erworbenen Grundfertigkeiten und Kompetenzen bilden den Schwerpunkt dieses Abschnittes. Im Mittelpunkt stehen dabei das fünfte und das sechste Schuljahr, aber einige Teile des Unterrichts sind schon im dritten und vierten Schuljahr gut umsetzbar. Wenn das zutrifft, werden wir darauf aufmerksam machen.

Den Informatikunterricht in der Primarschule unterteilen wir in zwei grosse Bereiche.

2.1 Programmieren

Hier stehen folgende Zielsetzungen im Vordergrund:

- Lösung von einfachen Problemstellungen von der Idee bis zur Implementierung in Form eines Programms (Steuerung der Maschinen und Kommunikation mit dem Computer),
- Erlernen grundlegender Programmierkonzepte, die im entsprechenden Alter zugänglich sind, mit besonderem Schwerpunkt auf dem modularen Entwurf,
- Unterstützung des Unterrichts der konstruktiven Geometrie,
- selbständige Überprüfung der Korrektheit (Testen) eigener Programme.

Programmierunterricht in diesem Alter ist eine fachdidaktische Herausforderung, weil der Begriff der Variablen eines Programms in ihrer Abstraktion frühestens im sechsten Schuljahr verständlich ist, und auch dann nicht allen Kindern. Alle höheren Programmiersprachen beginnen mit der Variablenspezifikation, ohne die man nicht anfangen kann zu programmieren. Die Lösung dieses didaktischen Problems liefert die konstruktivistische Schule von Jean Piaget, aus der die speziell für den Programmierunterricht entwickelte Sprache LOGO kommt. Ohne das Konzept der Variablen ermöglicht sie echten Programmierunterricht mit tieferem Verständnis für folgende zentrale Konzepte:

- einfache Schleifen,
- das Konzept von Unterprogrammen und somit des modularen Entwurfs,
- das Konzept der passiven Variablen (Parameter).

Dabei unterstützt Piagets Schule eine systematische Suche nach Fehlern und – durch Zeichnen von geometrischen Figuren – den Unterricht der konstruktiven Geometrie. Durch das Konzept der Unterprogramme schaffen es Primarschulkinder, komplexe Aufgaben zu lösen, indem sie zuerst einfache Programme schreiben und diese als Bausteine (so genannte Module) zur Lösung komplexerer Aufgaben verwenden. Die neuen Programme werden dann als Bausteine zur Lösung noch komplexerer Aufgaben verwendet. Auf diese Art und Weise legen die Kinder drei bis fünf Schleifen problemlos ineinander, was schon für professionelle Programmiererinnen und Programmierer ohne modulare Technik eine ernsthaftere Herausforderung ist.

Die oben genannten Ziele 1, 2, 3, 5 und 6 werden in diesem Programmierunterricht in hohem Mass erreicht. Für den Unterricht stehen altersgerechte Unterrichtsunterlagen und das Lehrbuch „Einführung in die Programmierung mit LOGO“ [2] (erste sieben Kapitel) zur Verfügung. Sie vertiefen das konstruktivistische Prinzip von Jean Piaget, indem die Kinder zum Teil selbst die verwendete Programmiersprache um neue Befehle erweitern und zur Entwicklung nur rund zehn einfache Steuerungsbefehle für den Computer verwenden. Die ersten vier Teile dieses Buches sind schon für Kinder ab der dritten Klasse zugänglich.

2.2 Lösungssuche für konkrete Problemfälle der Informationsverarbeitung

Die Entwicklung von Algorithmen für die Lösung von Problemen mit einer unendlichen Vielfalt von Problemfällen liegt im Regelfall weit über dem Abstraktionsvermögen der Primarschulkinder. Daher lösen die Kinder selber statt automatisch (d.h. ohne Verwendung des Computers) konkrete Problemfälle mit einer überschaubaren Menge an Lösungsmöglichkeiten. Dies trägt zu Erfüllung folgender Zielsetzungen bei:

- Entwicklung der ersten Stufe des algorithmischen Denkens,
- Verwendung von abstrakten grafischen Darstellungen und Objekten mathematischer Natur zur Darstellung von Problemstellungen und Situationen,
- Erkennen der Vielfalt der Aufgaben der Informationsverarbeitung im Alltag,
- die erste Berührung mit wichtigen Konzepten der Informatik.

Die Aufgabestellungen für die Lösungssuche können aus folgenden Bereichen stammen.

(Am ABZ wird an einem Lehrbuch gearbeitet.)

- Suche im Raum oder in einer sortierten Folge,
- Sortieren von unterschiedlichen Objekten nach Gewicht, Grösse usw.,
- Suche nach dem kürzesten Weg (Zug, Auto, zu Fuss) in einem Verkehrsnetz,
- unterschiedliche diskrete Optimierungsaufgaben,
- Vermeiden von Deadlock in der Steuerung von Datenflüssen, Autos usw.
- Verschlüsseln und Entschlüsseln mit einfachen Kryptosystemen.

Eine Vielfalt konkreter Aufgabestellungen sind für unterschiedliche Altersgruppen in den Heften des Wettbewerbs „Informatik-Biber“ [3] oder im Buch „CS Unplugged“ [4] zu finden. Die Erstellung von Unterrichtsunterlagen, die auf die Entwicklung der Strategien für unterschiedliche Aufgabestellungen ausgerichtet sind, steht uns noch bevor. In Primarschulen sollte das Schwergewicht darauf gelegt werden, dass man lernt, alle Möglichkeiten für eine Lösung in Betracht zu ziehen und sie auch bei einigen Aufgaben systematisch aufzulisten.

3. Umsetzung der Bildungsziele in der Sekundarstufe I

Den Unterricht kann man in drei grössere Blöcke unterteilen. Dabei werden alle oben genannten sechs Zielsetzungen verfolgt, nur mit dem Unterschied, dass sich das Ziel 6 als eine Brücke zwischen Mathematik und Informatik nicht nur auf den Bereich der Geometrie beschränkt.

3.1 Programmieren im Kleinen

Programmieren im Kleinen fokussiert auf die Entwicklung von Programmen zur Lösung konkreter Aufgaben und das Erlernen der grundlegenden Programmierkonzepte zu diesem Zweck. Es meidet die Themen des Software Engineering (Programmieren im Grossen), das der Modifikation und Einbettung riesiger Programmpakete gewidmet ist.

Aus der Sicht der Programmierkonzepte lernt man:

- das Konzept der Variablen und Arrays,
- bedingte Schleifen und Verzweigung der Programme mit if,
- einfache Rekursion.

Dabei kann man zuerst mit dem Lehrbuch „Einführung in die Programmierung mit LOGO“ [2] beginnen und danach zu einer hohen Programmiersprache (am besten aus der ALGOL-Familie) wechseln. Ausserdem ist es empfehlenswert, Roboter zu bauen und Programme zu ihrer Steuerung zu entwickeln.

Aufgabentypen können aus der Geometrie (LOGO), aus allen Bereichen der Mathematik, aus der Visualisierung der Zusammenhänge und der Simulation von Abläufen aus der Physik und der spezifischen Aufgabestellungen der Informationsverarbeitung kommen. Dabei liegt das Schwergewicht auf der Verzahnung des analytischen Denkens der Mathematik und des konstruktiven algorithmischen Denkens.

3.2 Entwicklung von Strategien zur Lösung unterschiedlicher Aufgaben der Informationsverarbeitung

Hier kann man alle Aufgabentypen wie in Abschnitt 2.2 in Betracht ziehen. Der Fokus ändert sich aber von der Lösungssuche für konkrete Problemfälle zur Entwicklung einfacher systematischer Vorgehensweisen zur Lösung bestimmter Aufgabentypen. Zusätzlich wird dabei die Berechnungskomplexität und somit die Effizienz anhand einfacher Beispiele angesprochen.

Binäre Suche und einfache Sortierverfahren sollten erlernt werden. Für einfache Aufgabestellungen kann man auch Backtracking unterrichten.

3.3 Informationssysteme, Datenschutz und sichere Kommunikation

Die Vorteile einer strukturierten Abspeicherung von Daten bei Bewahrung wichtiger Zusammenhänge sollten erläutert werden. Eine einfache relationale Datenbank für einen gut motivierten Zweck sollte so implementiert werden, dass die Antworten auf typische Fragestellungen effizient erledigt werden. Die Wichtigkeit des Datenschutzes soll erläutert werden, und grundlegende Konzepte des Datenschutzes sind zu vermitteln.

Einfache historische Beispiele von Kryptosystemen zum sicheren Kommunikationsaustausch sollen vorgestellt werden und neue Varianten im Unterricht entworfen sowie geknackt werden. Der Unterricht fokussiert gleichmässig auf den Entwurf von Kryptosystemen (Kryptographie) sowie auf die Kryptoanalyse. Die einfachen statistischen Konzepte zur Analyse der monoalphabetischen Kryptosysteme sowie zum Entwurf des Kryptosystems Vigenère sollen vermittelt werden. Die Grundidee der Public-Key-Kryptosystemen als Basis für e-Commerce sollte erläutert werden. Zum Unterricht der sicheren Kommunikation steht das Lehrbuch des ABZ, „Einführung in die Kryptologie“ [5] zur Verfügung.

Juraj Hromkovic

April 2012

Referenzen

- [1] Juraj Hromkovic, „Sieben Wunder der Informatik“. 2. Auflage, Vieweg+Teubner 2008.
- [2] Juraj Hromkovic, „Einführung in die Programmierung mit LOGO“. 2. Auflage, Springer Vieweg 2012.
- [3] Informatik-Biber Schweiz. www.informatik-biber.ch.
- [4] Computer Science Unplugged. csunplugged.com.
- [5] Karin Freiermuth, Juraj Hromkovic, Lucia Keller und Björn Steffen, „Einführung in die Kryptologie“. Vieweg+Teubner 2010.