

Programmieren und denkend Probleme lösen

Informatik wird wieder als Ergänzungsfach von Schweizer Gymnasien angeboten (vgl. Beitrag Seite 26). Einen wesentlichen Teil soll dabei das Erlernen des Programmierens ausmachen. Doch welche Programmiersprache eignet sich am besten als Lehrsprache?

Sicherlich sollte die jeweilige Programmiersprache einfach zu erlernen und syntaktisch klar verständlich sein, es den Lehrenden jedoch ermöglichen, ihre Klassen auf die Praxis vorzubereiten, also die grundlegenden Konzepte für die tatsächliche Anwendung zu vermitteln. Es überrascht nicht, dass die Meinungen der Fachleute auseinanderdriften. Im Folgenden versuchen wir, wichtige Standpunkte darzulegen.

Konzepte vermitteln vs. konkrete Anwendung

Helmar Burkhart von der Universität Basel vertritt die Auffassung, dass die wesentliche Aufgabe der Lehrenden darin bestehe, die Konzepte der Informatik zu vermitteln. Hierbei seien «algorithmische Belange wichtiger als syntaktische Details». Dies scheint umso sinnvoller, wenn man die Schnellebigkeit der Informatik berücksichtigt. Dennoch, fährt Burkhart fort, sei eine Sprachempfehlung notwendig. Er spricht sich für eine möglichst moderne Programmiersprache aus, die z.B. Schnittstellen für eine verteilte Datenverwaltung bietet und durch ihre konkrete Anwendbarkeit für die Lernenden interessant wird.

Nach Abraham Bernstein von der Universität Zürich geht es nicht nur um das Programmieren, sondern um die Denkweise von Informatikfachleuten. Das lässt sich als ein Argument für das frühe Erlernen des Programmierens werten: Das Programmieren erhöht (gerade bei Kindern) die Fähigkeit, sich klar und unmissverständlich auszudrücken und unterstützt die Entwicklung des logischen Denkens.

Von Logo über Pascal zu Delphi

Laut Jürg Gutknecht von der ETH Zürich sind C bzw. C++ für den Einstieg in die Programmierung nicht geeignet. Des Weiteren sei es sinnvoll, mit den Grundlagen – der Programmierung von Datenstrukturen und Algorithmen – zu beginnen und Dinge wie Objekte, Instanzen,

Vererbung usw. zunächst aussen vor zu lassen. Aus diesem Grund sei auch das objektorientierte Java nicht die erste Wahl für den Einstieg in die Programmierung. Eine ähnliche Ansicht vertritt Juraj Hromkovic von der ETH Zürich, der Pascal und Delphi empfiehlt. Auf den sehr erfolgreichen öffentlichen Lehrveranstaltungen «Programmieren für alle» der ETH werden zusätzlich noch einige Lektionen Logo vorgeschoben. So lernen insbesondere Kinder spielend das Programmieren. Hierbei steuern sie eine Schildkröte über den Bildschirm, wodurch Zeichnungen entstehen. Dazu sind alle wesentlichen Bestandteile einer Programmiersprache verfügbar: Bedingungen, Schleifen, Variablen usw.

Gleich von Beginn weg objektorientiert?

Peter Müller von der ETH Zürich möchte nicht mit einer prozeduralen Programmiersprache (Pascal) beginnen und dann zu einer objektorientierten wechseln (Delphi). Er schlägt vor, Schülerinnen und Schüler von Anfang an mit objektorientierten Sprachen wie Java oder C# vertraut zu machen, aber die Idee der Prozeduren und Klassen erst zu einem späteren Zeitpunkt explizit einzuführen. «Dieser Ansatz ermöglicht es, später noch weiterführende Konzepte zu vermitteln und spannende Programme zu schreiben. Ich denke, Lernende lassen sich mit modernen Techniken leichter motivieren.»

Hans Hinterberger von der ETH Zürich betont, dass es zudem sehr vorteilhaft sei, wenn Auszubildende vom Neuling bis zum professionellen Programmierer die gleiche Sprache und die gleichen Konzepte anwenden.

Fazit: Problemlösung im Zentrum

Es besteht Einigkeit darüber, dass die Problemlösung sowie das algorithmische Denken und nicht die Programmiersprache den Mittelpunkt des Programmierunterrichts bilden sollten.

Für Kinder bewährt sich die Programmiersprache Logo, weil sie einerseits spielerisch zu erlernen ist, andererseits alles bietet, was eine «richtige» Programmiersprache auszeichnet. Für Einsteigende und Fortgeschrittene sind Pascal oder Delphi zweckmässig, da sie eine einfache Syntax haben und der Quellcode eines Programms auch für Laien verhältnismässig leicht nachzuvollziehen ist. Für Fortgeschrittene eignen sich objektorientierte Sprachen wie Java oder C++.

Davon abgesehen sollte die verwendete Software möglichst plattformunabhängig und frei zugänglich sein, damit zuhause mit denselben Mitteln wie im Unterricht geübt werden kann.

Herbert Bruderer und Dennis Komm,
ETH Zürich

Weiter im Netz

Der vollständige Wortlaut des Beitrags ist abrufbar unter www.educ.ethz.ch

Schweizer Tag für Informatikunterricht

Der erste Schweizer Tag für Informatikunterricht am 15. Januar 2010 richtet sich an alle Mathematik-, Physik- und Informatiklehrpersonen ab Primarstufe und steht im Zeichen des praxisnahen Lernens. In Workshops können Lehrpersonen Unterrichtsmodule aus der Perspektive der Schülerinnen und Schüler erleben – und später im eigenen Unterricht nutzen. Sie haben die Wahl aus insgesamt neun Workshops, in denen Teilnehmende selbst Konzepte und praktische Ausarbeitungen für den Unterricht durchspielen. Info und Anmeldung bis 30. November 2009: www.abz.inf.ethz.ch/stiu