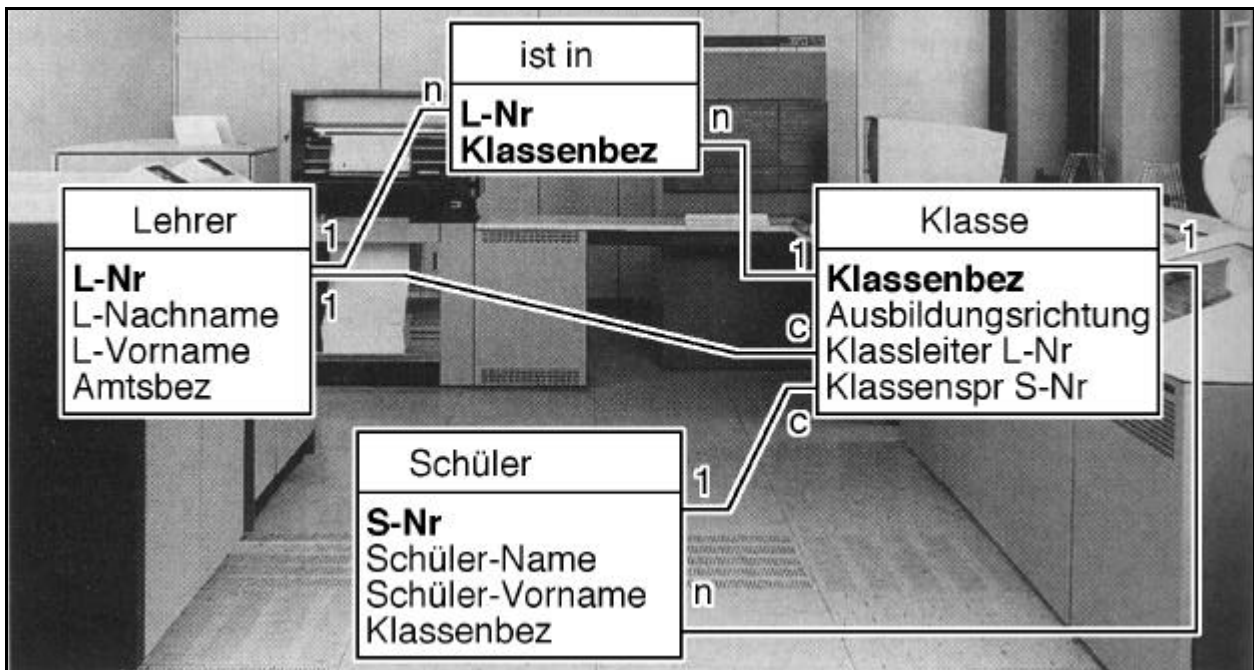


Informatik in der Schule



Informatik in der Schule

Schwerpunkte
Hinweise auf Materialien
Informationsquellen

Zentralstelle für Computer im Unterricht
Augsburg 2000

Erarbeitet im Auftrag des Bayerischen Staatsministeriums für Unterricht und Kultus
an der Zentralstelle für Computer im Unterricht, Augsburg,
im Rahmen des Arbeitskreises

"Überlegungen zur Weiterentwicklung des Informatikunterrichts"

Autoren:

StD Helmut Burger	Gymnasium München-Moosach
RSK Rudolf Galata	Staatliche Realschule München II
StRin Andrea Hechenleitner (ab 1. September 1999)	Staatsinstitut für Schulpädagogik und Bildungsforschung
StD Dr. Klaus Kreisel	Emmy-Noether-Gymnasium Erlangen
StR Max Leppmeier (bis 1. September 1999)	Staatsinstitut für Schulpädagogik und Bildungsforschung
StD Werner Liessel	Zentralstelle für Computer im Unterricht
OStR Albert Wiedemann	Erasmus-Grasser-Gymnasium München

Redaktion:

Werner Liessel, Jutta Niedermaier

© Nachdruck – auch auszugsweise – nur mit Genehmigung der Zentralstelle für Computer im Unterricht, Augsburg

Zentralstelle für Computer im Unterricht, Schertlinstr. 9, 86159 Augsburg

Telefon (08 21) 2 59 19-0, Telefax (08 21) 2 59 19-19

E-Mail-Adresse: zentralstelle@zs-augsburg.de

Internet-Angebot: <http://www.zs-augsburg.de>

Vorwort

Informatik in der Schule kann bislang noch nicht auf eine lange Tradition zurückblicken. Erste Ansätze in Form von Wahlunterrichtsangeboten reichen in die Zeit von 1970 zurück. Durch den raschen Verbreitungsgrad der zugrunde liegenden Technik, spätestens seit der Marktreife der PCs, hat sich zunehmend die Notwendigkeit gezeigt, Voraussetzungen für den sinnvollen Umgang damit zu schaffen. In aller Regel wurde dabei der Nutzungsaspekt in den Vordergrund gestellt und in der Konsequenz der für alle Schülerinnen und Schüler verpflichtende Unterricht in informationstechnischer Grundbildung eingerichtet, der die Kinder und Jugendlichen für eine Informations- und Kommunikationsgesellschaft qualifizieren sollte.

Bei dieser Entwicklung geriet der bis Ende der achtziger Jahre langsam ausgebaute Informatikunterricht im engeren Sinne, der sich mit der Information als zentralem Lerninhalt und ihrer – automatischen – Verarbeitung befasst, wieder fast vollständig aus dem Blickfeld. Ein gesellschaftlicher Anspruch in diese Richtung wurde nicht angemeldet, weder von Wirtschaftsseite noch von den Eltern oder der Politik. Auch entsprechende Forderungen von Expertengruppen wurden nicht aufgegriffen. Lediglich für den Hochschulbereich war und ist die Informatik als eigenständige Wissenschaft akzeptiert.

Zum Beginn des 21. Jahrhunderts existiert nun die prekäre Situation, dass eine gut entwickelte Wissenschaft, die in dem innovativsten Bereich tätig ist, den es derzeit gibt, auf keinen entsprechenden Unterbau im Schulsystem zurückgreifen kann. Die in den vergangenen Jahren niedrigen Studentenzahlen in den entsprechenden Studiengängen haben dies in deutlich sichtbarer Weise belegt. Die Hochschulinformatik hat sich stürmisch weiterentwickelt und ihre Grundlagen weitgehend geklärt. Die Lehrplanvorgaben und Stundentafeln in den Schulen sind demgegenüber auf dem Kenntnisstand der achtziger Jahre stehen geblieben.

Mit dieser Veröffentlichung wird der Versuch unternommen, einerseits die Notwendigkeit auch einer didaktischen Weiterentwicklung der Informatik als Schulfach zu unterstreichen, andererseits aufzuzeigen, wie im Rahmen der bestehenden Gegebenheiten ein inhaltlich und methodisch zeitgemäßer Informatikunterricht in den Schulen verwirklicht werden kann. Die Broschüre richtet sich daher in erster Linie an die in den verschiedenen Schularten unterrichtenden Lehrkräfte und will ihnen Hilfestellung geben in Form von Hinweisen auf eine angemessene Interpretation der geltenden Lehrpläne sowie durch Empfehlungen für geeignete Werkzeuge (Programmierumgebungen) und weiterführende Literatur bzw. erprobte Internet-Adressen. Sie versteht sich daher auch als ein Baustein zu der permanent erforderlichen eigenen Fortbildung der Informatik-Lehrkräfte.

Das Bändchen ist aber auch für die interessierte Öffentlichkeit geeignet, um beispielsweise Eltern oder Vertretern der Politik und Wirtschaft aufzuzeigen, was Schule bei Schaffung entsprechender Voraussetzungen zu leisten in der Lage ist, um die jungen Menschen auf ein Leben in einer von Information geprägten Welt vorzubereiten. Nur wenn es gelingt, über breite Schichten und Gruppierungen hinweg einen Konsens dahingehend zu erzielen, dass nicht nur der reine Anwendungsaspekt, sondern ein tiefergehendes strukturelles Verständnis der Grundlagen der zur Anwendung kommenden Techniken integraler Bestandteil der All-

gemeinbildung sein muss, wird es mittelfristig möglich sein, die sich abzeichnenden Qualifikationsprobleme in unserem Lande zu lösen. Dies betrifft nicht nur den engeren Bereich der unmittelbar mit Informations- und Kommunikationstechnik verknüpften Arbeitsplätze, sondern nahezu alle Arbeits- und Entscheidungsbereiche vom reinen Anwender bis hin zu politischen und wirtschaftlichen Führungs- und Entscheidungsträgern. Grundlegende informatische Inhalte, wie sie in einem Pflichtfach Informatik vermittelt werden sollen, sind zu einem unverzichtbaren Bestandteil der Allgemeinbildung geworden.

Im ersten Abschnitt werden für die interessierte Öffentlichkeit die informatischen Inhalte allgemein und ihre Relevanz für die Allgemeinbildung, wie sie in der Schule vermittelt werden soll, dargestellt. Der zweite Abschnitt entwickelt eine zeitgemäße Didaktik für ein Fach Informatik in den allgemeinbildenden Schulen, gegliedert nach den Schulstufen. Im dritten Abschnitt wird aufgezeigt, wie auf der Grundlage der derzeit gültigen Stundentafeln und Lehrpläne im bayerischen gegliederten Schulsystem diese didaktischen Überlegungen bereits heute – wenigstens zu größeren Teilen – umgesetzt werden können. Zur Unterstützung der unterrichtenden Lehrkräfte sind in den Abschnitten 4 und 5 umfangreiche Hilfen zusammengestellt, die sowohl Aussagen zu der Verwendung von Programmierwerkzeugen als auch kommentierte Literatur- und Internet-Hinweise sowie ein Glossar der verwendeten Fachbegriffe enthalten.

Die Autoren hoffen, dass mit dieser Broschüre einerseits die Diskussion um die Weiterentwicklung des Faches Informatik erneut angeregt wird, andererseits den Lehrkräften entsprechende Unterstützung bei der schwierigen Aufgabe gegeben werden kann, in einem sich so dynamisch entwickelnden Fachgebiet sich einigermaßen aktuelle Kenntnisse durch eigene Fortbildung anzueignen.

Augsburg, April 2000

Werner Liessel,
Leiter der Zentralstelle

Informatik in der Schule

Inhalt

1	Allgemeine Anmerkungen.....	9
1.1	Warum Informatik in der Schule.....	9
1.2	Informatik – eine allgemeinbildende Wissenschaft.....	11
1.3	Wie Informatik in der Schule	15
1.4	Informatik als Teil der informationstechnischen Bildung	17
2	Zielvorstellungen.....	19
2.1	Für die Sekundarstufe I.....	19
2.1.1	Erster Zugang zur Informatik.....	19
2.1.2	Die weiteren Inhalte im Überblick.....	20
2.1.3	Ziele und Schwerpunkte	20
2.1.3.1	Grundsätzliches	20
2.1.3.2	Einführung in die wichtigsten Begriffe und Denkweisen der Informatik.....	21
2.1.3.3	Vertiefung der eingeführten Begriffe und Denkweisen.....	22
2.2	Für die Sekundarstufe II.....	24
2.2.1	Vorbemerkungen.....	24
2.2.2	Themen.....	24
2.2.2.1	Modellierungen	24
2.2.2.2	Verteilte Systeme	25
2.2.2.3	Typische Datenstrukturen.....	27
2.2.2.4	Aufwandsabschätzungen.....	28
2.2.2.5	Spezifikation / Verifikation.....	28
2.2.2.6	Methoden der Softwareentwicklung	29
2.2.3	Schlussbemerkungen	30
2.3	Bemerkungen zur Methodik	30
2.3.1	Teamarbeit.....	30
2.3.2	Themenzentrierter Unterricht	30

3	Hinweise zur Umsetzung	33
3.1	Voraussetzungen.....	33
3.2	Hauptschule.....	34
3.3	Realschule (4-stufig bzw. 6-stufig).....	36
3.4	Gymnasium.....	39
3.4.1	Grundsätzliches.....	39
3.4.2	Wahlunterricht.....	39
3.4.3	Additum (Jahrgangsstufen 9 und 10).....	41
3.4.4	Grundkurs.....	42
3.4.5	Schulversuch "Europäisches Gymnasium" Zweig III.....	44
4	Werkzeuge für den Informatikunterricht	45
4.1	Vorbemerkungen zu den Beschreibungskriterien.....	45
4.2	Standardwerkzeuge	45
4.2.1	Relationale Datenbanken.....	45
4.2.1.1	MS-Access	46
4.2.1.2	DBase.....	46
4.2.1.3	Paradox.....	46
4.2.1.4	Oracle	46
4.2.2	Hypertextsysteme	47
4.2.2.1	HTML	47
4.2.2.2	PDF.....	47
4.2.3	Tabellenkalkulation.....	47
4.2.4	Sonstige Standardwerkzeuge	48
4.2.4.1	Textverarbeitung	48
4.2.4.2	Zeichen- und Malprogramme	48
4.3	Programmierwerkzeuge	48
4.3.1	Imperative und objektorientierte Programmiersprachen (PC)	49
4.3.1.1	Modula-2	49
4.3.1.2	Oberon-2	50
4.3.1.3	Pascal.....	51
4.3.1.4	Java.....	51
4.3.1.5	C / C++	52
4.3.1.6	Basic	52

4.3.2 Imperative und objektorientierte Programmiersprachen (Apple Macintosh)	53
4.3.2.1 MPW	53
4.3.2.2 MetroWerks	54
4.3.2.3 Oberon/F	54
4.3.3 Imperative und objektorientierte Programmiersprachen (Linux)	54
4.3.4 Weitere Programmiersprachen	54
4.3.4.1 LOGO	54
4.3.4.2 Funktionale Sprachen	55
4.3.4.3 Beschreibende Sprachen	55
4.4 Sonstige Programmiersysteme	55
4.4.1 Roboter Karel	55
4.4.2 Kara, der Marienkäfer	56
4.4.3 Lego-Roboter	56
4.4.4 Autorensysteme	56
5 Weitere Hilfen	59
5.1 Bücher (Eine enge Auswahl)	59
5.2 Internet-Adressen (Stand: Februar 2000)	61
5.3 Glossar	63

1 Allgemeine Anmerkungen

1.1 Warum Informatik in der Schule

Die zentrale Aufgabe der weiterführenden Schulen ist es, den Schülerinnen und Schülern eine allgemeine und berufsvorbereitende Bildung zu vermitteln, die sie in die Lage versetzt, sich in den unterschiedlichen Lebenssituationen zurechtzufinden, wertend ihre Umwelt zu begreifen und aktiv ihr Leben zu gestalten. Sie entwickeln dazu in der Schule wichtige Handlungsweisen, Vorstellungen und innere Bilder. Im naturwissenschaftlichen Bereich bilden die grundlegenden Begriffe Energie, Stoff und Leben den Zugang zur realen Welt; die Fächer Physik, Chemie und Biologie haben sich dazu vor etwa hundert Jahren in der Schule etabliert und sind seither allgemein als notwendig anerkannt.

In den letzten Jahren ist auf Grund des explosionsartigen Anwachsens des Wissens, der immer schneller werdenden Verarbeitung und Verbreitung durch Computer und Computernetze ein neuer grundlegender Begriff in den Vordergrund gerückt, die Information. Während sich dazu das Hochschulfach Informatik seit etwa 30 Jahren entwickelt hat und dort im allgemeinen Fächerkanon anerkannt ist, herrscht in der Öffentlichkeit eine eher nebulöse Vorstellung über die Inhalte dieser Wissenschaft. In der Schule steht die Informatik erst an der Schwelle der Anerkennung als eigenes Unterrichtsfach. Zur Zeit wird in der Schule auf diese neue grundlegende Sichtweise des Begriffs Information noch zu wenig oder überhaupt nicht eingegangen.

Im Folgenden soll exemplarisch aufgezeigt werden, welche Bedeutung die strukturellen und technischen Entwicklungen der Informatik auch im täglichen Leben haben:

Ähnlich wie sich die Menschheit bei der Entwicklung der modernen Naturwissenschaften vor allem in den vergangenen zweihundert Jahren die "greifbare" Natur dienstbar gemacht hat, ist es in den letzten Jahrzehnten gelungen, auch die "Information" durch genaue Analysen und entsprechende Modellbildungen – etwa vergleichbar mit den Analysen und Modellbildungen der "materiellen" Natur in der Physik – wissenschaftlich zu begreifen, mit Theorien zu beschreiben und schematisch darzustellen. Als Beispiel sei ein Ergebnis dieser Überlegungen genannt: Es gibt kein Computerprogramm, das ohne Zusatzinformation andere Computerprogramme auf Korrektheit überprüfen oder sie gar reparieren kann. Trotzdem wird, aus Unkenntnis dieses einfachen wissenschaftlichen Ergebnisses, heute immer noch vielerorts auf das Entstehen solcher Prüf- oder Reparaturprogramme gehofft, genau so wie man früher auf das Perpetuum mobile als unerschöpflichen Energiespeicher gesetzt hat.

Teilweise parallel zur theoretischen Fundierung der Informatik vollzog sich eine stürmische technische Entwicklung, so dass durch die Computer und deren Vernetzung immer komplexere Aufgaben erst lösbar und riesige Datenmengen erst verarbeitbar wurden. Durch den Preisverfall haben sich die Computer bis in den privaten Bereich jedes Einzelnen verbreitet und sind aus dem Alltagsleben nicht mehr wegzudenken; man vergleiche nur ein Büro oder

einen Bankschalter von vor dreißig Jahren und von jetzt.

Der zugrundeliegende Begriff Information hat einen ähnlich hohen Stellenwert erreicht wie der der Energie. Für die zum Teil völlig neuen Aufgabenfelder reichten die bis dahin bekannten Schematisierungsverfahren nicht mehr aus; es mussten neue Modellierungstechniken, komplexe Darstellungen von Informationsstrukturen und Operationen darauf erst entwickelt und Grundfragen der Kommunikationsstrukturen geklärt werden, mit weltweiter Bedeutung, gerade auch im und durch das Internet. Für die Sicherheit und die Schutzrechte im Internet ist z. B. jeder einzelne zuständig und sollte dazu auch die nötigen Grundkenntnisse besitzen.

Um auch dem Nichtfachmann die Möglichkeiten und Errungenschaften der Informatik nutzbar zu machen, wird versucht, den Informationsaustausch zwischen Mensch und Maschine immer intuitiver ("menschlicher") zu gestalten. Dabei wird aber zunehmend verdeckt, dass sich die maschinelle Informationsverarbeitung vom menschlichen Denken grundlegend unterscheidet. Gerade bei komplexeren Informationen gerät dadurch sehr schnell in Vergessenheit, dass man den "Wert", d. h. den tatsächlichen Informationsgehalt eines maschinellen Ergebnisses, grundsätzlich hinterfragen muss. Man denke zum Beispiel an einen Arzt, der ein Expertensystem¹ zur Krankheitsdiagnose verwendet, eine Situation, die durch das schnelle Anwachsen des Wissens über Krankheiten und deren Therapien zum "Normalfall" werden wird. Der Arzt gibt die beobachteten Symptome ein und erhält vom Computer eine andere Diagnose als er selbst vermutet. Zwischen zwei Ärzten wäre jetzt eine Fachdiskussion möglich. Bei der Arbeit mit dem Expertensystem kann der Arzt sagen: "Ich bin auf jeden Fall flexibler als eine Maschine, meine Diagnose stimmt." Er kann auch sagen: "Der Computer macht keinen Fehler, also ist seine Diagnose richtig." Wenn er aber mit den Arbeitsweisen der Informatik vertraut ist, so kann er die Entscheidungsfindung aus der Sicht maschineller Informationsverarbeitung nachvollziehen und Überlegungen anstellen wie: "Habe ich Zusatzinformation ausgewertet, die das Expertensystem nicht hatte?" oder "Habe ich mehrere Aussagen assoziativ zusammengefasst, die die Maschine nur über vorgegebene, feste Gewichtungen verknüpfen kann?" Mit solchen Überlegungen kann er sowohl den Wert seiner wie auch den Wert der Maschinendiagnose abschätzen und ein wesentlich fundierteres Gesamturteil bilden. Dazu benötigt er aber Grundkenntnisse aus der Informatik; die informationstechnische Grundbildung (ITG) reicht dazu bei weitem nicht aus.

Aber auch schon in wesentlich einfacheren Situationen ist ein Hintergrundwissen aus der Informatik vorteilhaft, ja wirtschaftlich geboten. Eine Informationsbeschaffung mit einer Suchmaschine² im Internet setzt z. B. voraus, dass man sich der zugrunde liegenden Verarbeitung der Information bewusst ist, wie die eigene Anfrage interpretiert wird, wie das Suchsystem die wichtigen Begriffe aus den vorhandenen Dokumenten extrahiert (die vor-

¹ Computersystem, das auf einem speziellen Wissensgebiet (Domäne) die Kompetenz von menschlichen Experten besitzt und als Beratungs- und Problemlösungssystem eingesetzt wird.

² Computersystem (im Internet), das "alle" anderen Angebote im Netz automatisch absucht (Suchroboter), die Inhalte in einer Datenbank katalogisiert und über ein Abfrageformular recherchierbar macht. Bekannteste Suchmaschinen im WWW sind Altavista und Yahoo.

handenen Dokumente indexiert) hat, in welcher Weise die relevanten Dokumente ausgewählt werden usw. Eine erfolglose Suche für ein Hobby ist für den Betroffenen bedauerlich; sie führt meist zu einem ziellosen, im besten Falle vielleicht sogar kurzweiligen Surfen im Internet. Ein Patentanwalt jedoch, der klären muss, ob für eine Erfindung bereits ein Patent existiert, oder ein Arzt, der alle Nebenwirkungen eines Medikaments sucht, kann bei einer Recherche, die nicht alle relevanten Dokumente zeigt, einen mehr als nur wirtschaftlichen Schaden haben. Hinreichend gute Grundkenntnisse in der Informatik tragen dazu bei, solche Schäden zu vermeiden.

Eine in der Arbeitswelt typische Situation ist die Verlagerung von Betriebsabläufen auf den Computer (Verwaltung, Steuerung, Informationsaustausch). Um über die Vor- und Nachteile des Computereinsatzes entscheiden zu können, ist es notwendig zu wissen, welche Aufgaben der Computer übernehmen kann. Gerade hier zeigt sich immer wieder, dass die Anwender in den wenigsten Fällen abschätzen können, welche Schematisierungsverluste einerseits auftreten können, aber auch welcher Gewinn sich andererseits aus dem Computereinsatz ziehen lässt, wenn die Arbeitsabläufe optimal angepasst werden. Hier können elementare Kenntnisse in den Modellierungstechniken gerade auch bei Nichtinformatikern die notwendigen Voraussetzungen für eine fundierte Entscheidung schaffen. Die Informatiker, die dann die eigentliche Übertragung ausführen, können in diesem Stadium kaum Hilfestellungen geben, weil sie ihrerseits die Anwendersituation zu wenig kennen.

Zentrale Fragen in diesen Beispielen sind: Welcher Ausschnitt der Wirklichkeit soll abgebildet werden? Welche Modelle sind dafür geeignet? Was bleibt bei dieser Abbildung erhalten, was geht verloren? Welcher Teil der Realität kann überhaupt abgebildet werden? Wofür müssen bei Bedarf neue Strukturen geschaffen werden? Wie geht man mit dem nicht abbildbaren, aber dennoch notwendigen Anteil um? Aus diesem Grund haben Modellierungen in der Informatik einen noch erheblich höheren Stellenwert als z. B. in der Physik.

In der sich schnell verändernden Welt sind die Grundthemen der Informatik: Modell, Information, System genauso wichtig geworden wie z. B. das Grundthema Energie der Physik; eine bloße Bedienung von Computern und deren Software reicht nicht mehr aus.

1.2 Informatik – eine allgemeinbildende Wissenschaft

Das Wort "Informatik" ist heutzutage in aller Munde, Informatik wird generell als zukunftsweisend, zukunftsträchtig, zukunftsnotwendig, usw. eingestuft. Fragt man aber nach, was Informatik beinhaltet, so erhält man in der Regel sehr vage Antworten, etwa in der Art: "halt mit dem Computer". Natürlich, ein Fach "Informatik" hat – in Bayern – bisher kaum jemand in der Schule selbst gehabt. Und das, was unter dem Titel "Informatik" angeboten wird, sind fast ausschließlich Programmierkurse, Einführungen in Anwenderprogramme, "Multimedia" (was auch immer das sein mag) oder Ähnliches. Auch die Erfahrbarkeit von Informatik im Alltag ist meist gleichzusetzen mit dem Umgang mit Computern.

Gilt also die Formel: "Informatik = Computer"? Der englische Begriff für Informatik, "computer science (Computerwissenschaft)", deutet auf den ersten Blick genau in diese

Richtung. Allerdings lässt man bei dieser vereinfachenden Sichtweise außer Acht, dass das Wort "science" deutlich mehr bedeutet als technisches Know-how oder die Kenntnis der Bedienmöglichkeiten von Betriebssystem und Programmen. Weiter kommt man, wenn man das deutsche Kunstwort "Informatik", zusammengesetzt aus "**I**nformation" und "Automatik", genauer analysiert. Informatik ist demnach die Wissenschaft von der Information und ihrer automatischen Verarbeitung. Was bedeutet aber wissenschaftliche Betrachtung von Information und deren Verarbeitung? Wo bleibt der Computer? Braucht man für Informatik am Ende gar keine Computer? Und was bringt die Beschäftigung mit Informatik für die heutige Gesellschaft?

Drehen wir das Rad der Geschichte kurz zurück. Wie weit? Bis zu Schickard mit seiner ersten automatischen Rechenmaschine (1623)? Hier könnte man vielleicht den Beginn des Computerbaus definieren. Aber mit Informationsverarbeitung hat eine Rechenmaschine zuerst einmal nichts zu tun, denn Zahlen für sich genommen tragen keine Information. Man kann den Wert der Summe "3000 + 4000" berechnen, aber erst, wenn man der Zahl 3000 die Bedeutung "3000 DM auf dem Konto" und der Zahl 4000 die Bedeutung "4000 DM Gehalt" gibt, könnte der Wert der Summe bedeuten, dass man jetzt 7000 DM zur Verfügung hat. Wie aktuell diese primitiv erscheinende Tatsache sein kann, zeigt der Absturz der amerikanischen Marssonde im September 1999³: Die berechneten Daten allein nützen wenig, wenn Massen bzw. Längen von einer Stelle in Kilogramm und Metern, von der anderen in englischen Pfund und Meilen interpretiert werden.

Soll man noch weiter zurückgehen, etwa zu Al Charizmi (um 800), von dessen Name sich das Wort Algorithmus ableitet? Er hat beschrieben, wie bestimmte Rechenvorgänge durchzuführen sind, aber über Informationsverarbeitung hat er keine Aussagen gemacht. Eher könnte man bei Raymundus Lullus (Ramón LLull, um 1300) anhalten, der als erster das Ziel der Logik konzipierte: Die Entwicklung universeller Kalküle zur Entscheidung der Wahrheit beliebiger Aussagen. Am besten aber hält man bei dem deutschstämmigen, amerikanischen Bergwerksingenieur Hermann Hollerith an, der ab 1884 (für die 1890 stattfindende amerikanische Volkszählung) mehrere Patente einreichte, in denen er die Lochkarte als Träger schematisierter Information einführte und Maschinen zur automatischen Auswertung dieser Information konstruierte. Hiermit wurde das enorme Potential automatischer Informationsverarbeitung zum ersten Mal deutlich aufgezeigt. (Die damalige Firma von Hermann Hollerith ist heute die Firma IBM, in Deutschland hieß sie bis nach dem 2. Weltkrieg "Hermann Hollerith GmbH".)

In den dann folgenden 50 Jahren wurden durch mathematische Analysen und Modellbildungen die Grundlagen geschaffen, um den Begriff Information wissenschaftlich zu erfassen. Damit beginnt die Informatik (wenn auch zunächst noch nicht als eigenständige Disziplin

³ Die 125 Millionen Dollar teure US-Sonde "Mars Climate Orbiter" erreichte am 23. September 1999, nach einer zehnmonatigen Weltraum-Reise über hunderte Millionen Kilometer, völlig intakt den Mars, zerschellte dann aber wegen eines simplen Fehlers auf der Marsoberfläche: Der Hersteller der Sonde, Lockheed Martin Astronautics, arbeitete mit englischen Maßeinheiten, das die Sonde steuernde Jet Propulsion Laboratory aber mit dem metrischen System. Der Unterschied in den Einheiten wurde erst bemerkt, nachdem der Climate Orbiter bereits zerstört war.

unter einem eigenen Namen) als Wissenschaft. Wesentliche Fragestellungen sind: Was ist "Information"? Wie lässt sich Information darstellen und schematisieren? Welche Information lässt sich schematisieren? Bleibt der Informationsgehalt beim Schematisieren erhalten? Welche Operationen lassen sich auf der schematisierten Information durchführen? Welchen Informationsgehalt hat das Ergebnis solcher Operationen, wie muss man es also interpretieren?

Die Frage nach den möglichen Operationen, heute formuliert als "Berechenbarkeit", wurde bereits in den Jahren 1936/37 von Alan Turing fundamental beantwortet. Die von ihm definierte Turing-Maschine kann alles berechnen, was man prinzipiell als berechenbar betrachtet. Von dieser Maschine konnte Turing aber auch zeigen, dass es vergleichsweise einfache Aufgaben gibt, die man damit nicht lösen kann, die also nicht "berechenbar" sind. Ein Beispiel ist das Halte-Problem: Eine Turing-Maschine, verallgemeinert ein Computer, kann nicht überprüfen, ob ein Algorithmus auf einer zweiten Maschine je zu einem Ende kommt oder nicht. Damit ist, übertragen auf den jetzigen Sprachgebrauch, aber auch das bereits oben erwähnte Problem beantwortet, dass es kein Computerprogramm geben kann, das ohne Zusatzinformation andere Computerprogramme auf Korrektheit prüfen oder sie gar reparieren kann.

Mit der Entwicklung und ständigen Verbesserung der elektronischer Rechenanlagen nach dem zweiten Weltkrieg wurden die gefundenen Ergebnisse der Informatik zunehmend universell einsetzbar. Die immens wachsende Geschwindigkeit und Speicherkapazität der Computer schuf die Voraussetzung, immer komplexere Aufgaben in vertretbarer Zeit lösen zu können. Die Computer verbreiteten sich ins Alltagsleben, teilweise unsichtbar als Teile in den unterschiedlichsten Geräten ("embedded" Computer); dies führte auch im Bereich "Darstellung der Information" zu neuen Aufgaben.

Für diese erweiterten Aufgabenbereiche reichten aber die bis dahin bekannten Schematisierungsverfahren (Modellierungstechniken) nicht mehr aus, da sie nur für statische Datenzusammenhänge mit einfachen mathematischen Operationen (z. B. Buchhaltung) entworfen wurden. Um komplexere Informationsstrukturen mit ihren entsprechend komplexeren Operationen klar und übersichtlich beschreiben zu können, wurden neue Modellierungsarten entwickelt (z. B. objektorientierte Modellierung). Gleichzeitig wurden durch die höheren Rechnerleistungen neben einfachen textuellen Darstellungen der Information auch aufwendigere Darstellungen wie Graphiken (inklusive graphischer Bedienoberflächen), Töne und bewegte Bilder bis hin zu Videos möglich, für deren formalisierte, auf diskrete Werte reduzierte Repräsentation ebenfalls geeignete Darstellungen gefunden werden mussten. Die parallel aufkommende Vernetzung der einzelnen Rechner stellte weitere neue Fragen nach geeigneten Modellierungen. Es müssen nicht nur die Grundfragen nach den Kommunikationsstrukturen beantwortet werden; Datenstrukturen und ihre Operationen müssen nun nicht mehr nur auf einem Rechner bzw. lokalen (Firmen-)Netz gleich modelliert werden, es werden weltweit einheitliche Darstellungen benötigt, die sowohl wohldefiniert sein müssen, aber auch soviel Information wie möglich repräsentieren sollen.

Weiter stellte sich heraus, dass oft viele Informationen von verschiedenen Benutzergruppen mit verschiedenen Schwerpunkten gesehen werden. Diese unterschiedlichen Sichtweisen

beziehen sich weniger auf verschieden umfangreiche Zugangsrechte, sondern auf die mit der jeweiligen Aufgabenstellung verbundenen wichtigen Teilinformationen. So wird der Datenbankadministrator⁴ einer relationalen Datenbank eine Strukturierung der Relationen nach indizierten bzw. nichtindizierten Schlüsseln, Anzahl der Schlüsselattribute, Anzahl der Fremdschlüssel und anderen Implementierungsaspekten vornehmen. Für den Buchhalter oder den für Einkäufe zuständigen Mitarbeiter sind diese Informationen völlig nebensächlich. Sie benötigen eine Reduktion (Projektion auf die benötigten Teilinformationen) der Datenbankinhalte mit den für sie wichtigen Zusammenhängen.

Heute stellt die Informatik eine ganze Reihe von Modellierungstechniken für die verschiedenen Aufgabenbereiche und Sichtweisen zur Verfügung; Diagramme spielen dabei eine ähnlich wichtige Rolle wie die graphischen Darstellungen bei den Funktionen in der Mathematik. **Dynamische Modelle** mit ihren Zustands-Übergangs-Diagrammen beschreiben diejenigen Aspekte eines Systems, die mit der Abfolge von Ereignissen bzw. Operationen zu tun haben, unabhängig davon, worauf die einzelnen Operationen ausgeführt und wie sie implementiert werden.

Objektmodelle mit den Instanzen- und Klassendiagrammen dienen zur Darstellung der Struktur der Objekte eines Systems und ihrer Beziehungen untereinander. Sie repräsentieren die statischen, strukturellen, datenbezogenen Aspekte und zugleich das dynamische Zusammenwirken (Methoden). Die Objekte sind darin die Einheiten, in die man die Welt unterteilen kann, quasi die Moleküle des Systems. Die **funktionalen Modelle** mit den zugehörigen Datenflussdiagrammen beschreiben diejenigen Aspekte eines Systems, die mit der Transformation von Daten zu tun haben. Sie erfassen, was ein System tut, unabhängig davon, wie es oder was getan wird.

Wo aber bleibt der Computer in diesem Aufgabenbereich? Für den Informatiker ist mit dem Finden der geeigneten Modellierung der interessante Teil seiner Arbeit gelöst. Die Realisierung auf dem Computer ist dann nur noch lästige Sklavenarbeit. In der Schule muss diese Frage etwas anders beantwortet werden. Der Schüler will seine Überlegungen überprüfen, will das Resultat seiner Arbeit haben, will sehen, "ob das Programm läuft". Eine Realisierung auf dem Computer zeigt außerdem, ob die Modellierung vollständig war, d. h. keine undefinierten Elemente mehr enthält. Ein dritter, nur auf den ersten Blick untergeordneter, Aspekt des Computereinsatzes ist, dass der Schüler Erfahrung über die Laufzeit von Algorithmen sammelt. Zusammen mit der theoretischen Laufzeitanalyse lernt er so abzuschätzen, ob bestimmte Aufgaben nicht nur prinzipiell, sondern auch in realistischer Rechenzeit gelöst werden können. Weiter lernt er, ob eine Lösung, die heute noch eine zu lange Rechenzeit benötigt, mit schnelleren Rechnern bzw. durch Parallelisierung brauchbar wird oder, z. B. bei exponentieller Laufzeitordnung, grundsätzlich nur für umfangsmäßig begrenzte Aufgaben einsetzbar ist. Ein aktuelles Beispiel für diese Fragestellung ist, ob neue Methoden entwickelt werden können, die es erlauben, dass verschlüsselte Nachrichten von Unbefugten entschlüsselt werden können.

Auch wenn ein Fachgebiet nicht durch Ausschlüsse definiert werden kann, ist hier eine Ne-

⁴ Person, die eine Datenbank konzipiert und betreut

gativanmerkung zu einigen dem Informatikunterricht zugeordneten Inhalten nötig. Verstehen von Zusammenhängen in der Informatik heißt sicher nicht, einen Computer einschalten oder ein bestimmtes Programm im Schlaf bedienen zu können. Verstehen heißt genau so wenig, die Syntax und die Möglichkeiten einer Programmiersprache "aus dem ff" zu beherrschen, alle Schnittstellen eines Betriebssystems auswendig zu wissen oder einen Computer zusammenbauen bzw. erweitern zu können. All das sind schöne und je nach persönlicher Situation auch mehr oder weniger nützliche Rezepte bzw. Kenntnisse – aber eben nur gespeicherte (Detail)information, die angewandt wird. Genau so wenig, wie man Physiker ist, wenn man Messgeräte verwenden kann, ist man Informatiker, wenn man einen Computer mit seinen Programmen bedienen kann. Verstehen heißt, einen Überblick über die Grundkonzepte und deren Zusammenhänge zu haben, das Prinzip zu kennen, nach dem Dinge ablaufen und Konsequenzen aus diesem Prinzip einschätzen zu können. Genau dieses Verstehen ist im Bereich der Informatik notwendig, um Gestaltender in unserer Gesellschaft bleiben zu können und nicht zum Abhängigen der Computerisierung zu werden. Das für das Verstehen unserer Gesellschaft und ihrer Zusammenhänge notwendige Grundwissen ist genau das, was Allgemeinbildung ausmacht. Und dazu gehört, wie oben aufgezeigt, heute die Informatik und damit der richtig verstandene Informatikunterricht.

1.3 Wie Informatik in der Schule

Der veränderte Stellenwert der Informatik hat nicht zur Folge, dass die Inhalte des bisher teilweise missverstandenen herkömmlichen Informatikunterrichts wegfallen; ihre Bedeutung reduziert sich nur gegenüber neuen Themen. In einem zeitgemäßen Informatikunterricht wird man weiterhin in irgendeiner Art und Weise programmieren oder Standardsoftware bedienen, jedoch nur als Hilfsmittel, um grundlegendere Inhalte darzulegen. Die Standardsoftware und die Programmierumgebungen sind einem ständigen Wandel unterworfen, wohingegen man erwartet, dass die zentralen Themen des Informatikunterrichts längerfristig Bestand haben. Dieses Zeitkriterium ist eine der Bedingungen, die die grundlegenden Inhalte erfüllen sollen.

Ein weiteres Kriterium, um die wesentlichen Lerninhalte des Unterrichts zu bestimmen, ist die Allgemeingültigkeit: Die Lerninhalte sollten einen möglichst breiten Anwendungsbereich haben, nicht nur innerhalb der Informatik, sondern möglichst auch im Alltag. Die Syntax einer Programmiersprache z. B. erfüllt dieses Kriterium nicht und ist deshalb auch kein grundlegender Inhalt.

Ein Kriterium mit besonderer Bedeutung für die Schule ist der Aspekt der Vermittelbarkeit. Die ausgewählten Lerninhalte sollten in das Anspruchsniveau der entsprechenden Schule passen. Sie sollten bei ihrem ersten, möglichst frühzeitigen Kennenlernen ein intuitives Erfassen und dann ein Aufgreifen in einem späteren Stadium der Schullaufbahn ermöglichen, um sie auf höherem Niveau ausbauen zu können. Das wiederholte "spiralförmige" Aufgreifen auf immer höheren Niveaus dient einem tieferen Verständnis (kumulatives Lernen). Ein Beispiel aus der Vergangenheit: Flussdiagramme (Blockdiagramm zur Beschreibung von Programmabläufen) waren zwar schon auf einem niederen Anspruchsniveau vermittelbar,

erforderten aber ein Umdenken, um auch Wiederholungen graphisch darzustellen; Struktogramme erfüllen dieses "Vertikalkriterium" besser.

Eine Leitlinie für einen zeitgemäßen Informatikunterricht ist das Grundsche ma der Informationsverarbeitung: Zunächst existieren Informationen nur in diffuser Form in den Vorstellungen von Menschen. Um sie weiter (maschinell) verarbeiten zu können, müssen sie in irgend einer Weise sprachlich formuliert, d. h. repräsentiert werden. Auf diesen oft räumlich verteilten Repräsentationen operieren Verarbeitungs- und Transportprozesse und schaffen so neue Repräsentationen. Durch deren Interpretation gelangt man schließlich zu neuen Informationen.

Mit Leitfragen kann man die wichtigen Themen des Informatikunterrichts weiter eingrenzen. Eine erste Leitfrage betrifft das Problemlösen mit Informatiksystemen: Wie können durch Entwicklung, Gestaltung und Anwendung von Informatiksystemen Probleme der Lebenswelt gelöst werden? Eine zweite Leitfrage spricht die Wirkprinzipien von Informatiksystemen an: Wie sind Informatiksysteme aufgebaut, welches sind die Prinzipien des Zusammenwirkens ihrer Komponenten und wie ordnen sie sich in größere Zusammenhänge ein? Eine dritte Leitfrage spricht die Grundlagen und Grenzen technischer Wissensverarbeitung an: Welches sind die Grundlagen und wo liegen die Grenzen formaler bzw. technischer Wissensverarbeitung, und wie kann die kognitive Autonomie des Einzelnen gewahrt werden?

Nach diesen Vorüberlegungen kommt man schließlich zu zentralen Lerninhalten:

Beschreibungstechniken, d. h. Methoden und Techniken zur Repräsentation von Informationen. Je nach dem Typ der beschriebenen Objekte unterscheidet man drei Arten von Beschreibungstechniken:

- Datenstrukturen stellen konkrete Repräsentationen von Informationen dar; auf ihnen operieren die Verarbeitungs- und Transportprozesse.
- Verarbeitungsvorschriften beschreiben konkrete Verarbeitungs- und Transportvorgänge auf Datenstrukturen.
- Systembeschreibungen modellieren meist auf einer höheren Abstraktionsebene Teilaspekte von Informations- und Kommunikationssystemen wie Abläufe, Zerlegung in Komponenten und deren Kommunikation, Objektstruktur usw.

Modellierungstechniken: Als Modell wird dabei eine abstrahierende Beschreibung eines realen oder geplanten Systems bezeichnet. Diese Beschreibung enthält alle Eigenschaften eines Systems, die für die vorliegende Zielsetzung wesentlich sind. Ein wichtiger Aspekt der Modellbildung ist die Strukturierung des Problems nach unterschiedlichen Gesichtspunkten. In Anlehnung an die verschiedenen Beschreibungstechniken erhält man als Ergebnis der Modellierung verschiedene Typen von Diagrammen: Datenflussdiagramme, Zustands-Übergangs-Diagramme, Objektdiagramme usw.

1.4 Informatik als Teil der informationstechnischen Bildung

Der Informatikunterricht kann nicht isoliert betrachtet werden, sondern ist eingebunden in den verbindlichen Fächerkanon des schulischen Bildungsauftrags. Einerseits schafft er durch seine spezifischen Ziele und Inhalte neben seinem eigenständigen Bildungswert vielfältige Hilfestellung für andere Fächer im Sinne einer Dienstleistungsfunktion. Andererseits benötigt der Informatikunterricht auch das Vorhandensein bestimmter Eingangsvoraussetzungen. Neben der Beherrschung der klassischen Kulturtechniken Lesen, Schreiben und Rechnen muss wegen der Nutzung technischer Hilfsmittel bei der Umsetzung und Realisierung der geistig geschaffenen Problemlösungen der geläufige Umgang mit den an der Schule vorhandenen informationsverarbeitenden Systemen als Fähigkeit bei den Schülerinnen und Schülern bereits vorhanden sein.

In der verpflichtenden **informationstechnischen Grundbildung** (ITG) wird daher allen Schülerinnen und Schülern ein erster Einblick in die Fragen im Zusammenhang mit der modernen Informationsverarbeitung vermittelt. Dazu gehören technische Fragen in gleicher Weise wie wirtschaftliche, soziale und gesellschaftliche Aspekte. Die informationstechnische Grundbildung zielt auf breit angelegte Grundkenntnisse und -fertigkeiten, damit die Schülerinnen und Schüler die für ihr persönliches Umfeld wichtige Bedeutung der neuen Techniken einschätzen können. Wesentliches inhaltliches Element ist dabei der Werkzeugcharakter des Rechners, wie er für die vielfältigen Aufgabenstellungen in Betracht kommt. Soweit es für das weitere Verständnis notwendig ist, sind auch fachliche und bedienungstechnische Voraussetzungen der Hard- und Software Lerninhalt.

Zur Sicherung der bedienungstechnischen Voraussetzungen (z. B. Gerätekunde, Handhabungskompetenz) dienen fächerverbindende Einführungs-, Projekt- und Abschlusstage, die auch eine systematische Zusammenschau der informationstechnischen Grundbildung erlauben. Da die informationstechnische Grundbildung neben ihrem eigenständigen Bildungsanliegen auch bestimmte Eingangsvoraussetzungen für das Fach Informatik bzw. den Einsatz des Computers im Fachunterricht schaffen muss, sollte sie möglichst früh einsetzen. Dafür spricht auch das zunehmend geringere Alter, in dem Kinder bereits eigene Erfahrungen mit dem Computer haben.

Beim Einsatz des **Computers im Fachunterricht** in den verschiedenen Unterrichtsfächern erleben die Schülerinnen und Schüler den Werkzeugcharakter des Computers in enger Verknüpfung mit den fachbezogenen Unterrichtszielen. Dabei stehen wesentliche Arbeitsmethoden des Faches im Vordergrund, die durch Informations- und Kommunikationstechniken effizient unterstützt werden. Ebenso wird die Bedeutung dieser Techniken für die berufliche Praxis und die Lebenswirklichkeit jedes einzelnen Menschen bewusst gemacht. Computergestütztes Lehren und Lernen eignet sich in besonderer Weise dazu, den Schülerinnen und Schülern lebensnahe persönliche Erfahrungen in der Arbeit mit dem Computer zu vermitteln und die Fähigkeit zum Umgang mit den Informations- und Kommunikationstechniken vorzubereiten, intensiver zu üben und so weiter zu vertiefen. Möglichen geschlechtsspezifischen Einschränkungen im Umgang mit den neuen Techniken kann dadurch entsprechend vorgebeugt werden.

2 Zielvorstellungen

In diesem Kapitel werden die Ziele für den Informatikunterricht in den betroffenen Schulstufen dargelegt, wie sie sich derzeit idealtypisch aus dem Stand der Fachdidaktik ableiten lassen. Es werden Entwicklungslinien aufgezeigt, die bereits jetzt im Unterricht aufgegriffen werden können, auch wenn noch nicht alle aufgeführten inhaltlichen Aussagen in den gültigen bayerischen Lehrplänen *expressis verbis* zu finden sind. Bei künftigen Lehrplanarbeiten sollten sie ihre entsprechende Berücksichtigung erfahren.

2.1 Für die Sekundarstufe I

Die Sekundarstufe I leistet einen wesentlichen Beitrag zur Allgemeinbildung der Schülerinnen und Schüler aller Schularten. Daher kommt ihr gerade für das Fach Informatik eine besondere Bedeutung zu. Da die Inhalte des Informatikunterrichts aufgrund der wachsenden Bedeutung von Informations- und Kommunikationstechniken eine beständige Weiterentwicklung erfahren und auch künftig stärkeren Veränderungen unterworfen sind, bedürfen die bisherigen Lerninhalte einer aktuellen Interpretation. Im Folgenden wird keine grundsätzlich neue Sicht entwickelt, sondern lediglich der Versuch zu einem zeitgemäßen Ansatz unternommen.

Die vertiefte, informationstechnische Bildung kann zwar auf Vorleistungen der ITG zurückgreifen, muss aber zuerst versuchen, die ungesicherten und aller Erfahrung nach sehr unterschiedlich ausgeprägten Vorkenntnisse über Informatiksysteme zu ergänzen, zu systematisieren oder zu wiederholen (u. U. nach bereits erfolgtem Unterricht im Fach Textverarbeitung).

2.1.1 Erster Zugang zur Informatik

Durch Verwendung leistungsfähiger, objektorientierter Entwicklungswerkzeuge (Hypertextsysteme, Programmierumgebungen, Graphikprogramme) sammeln die Schüler erste Erfahrungen mit Informatik. Problemstellungen und Operationen orientieren sich an realen Vorlagen des Alltags. Modellieren beginnt mit dem Konfigurieren der vorhandenen Umgebung gemäß den persönlichen Vorstellungen.

Vorteile des Ansatzes	Nachteile des Ansatzes
Orientiert sich an den kognitiven Voraussetzungen von Anfängern	Beschränkt die Tiefe der zu gewinnenden Informatikkenntnisse
Mit geringen Vorkenntnissen lassen sich leistungsfähige Produkte konstruieren	Ein Umstieg auf richtige Programmiersprachen ist notwendig, da z. B. Datentypen wie Records fehlen

Vorteile des Ansatzes	Nachteile des Ansatzes
Realisiert einen stufenweisen Einstieg in die Programmierung	Löst das Problem unterschiedlicher Vorkenntnisse nur teilweise
Vereinigt Entwicklungsrichtungen wie Objektorientiertheit und Multimedia	

2.1.2 Die weiteren Inhalte im Überblick

Die problembezogene Auswahl und Anwendung geeigneter Anwendungssysteme⁵ führt zum Kennen und Nutzen informatischer Handlungsstrategien. Aufbau, Arbeitsweise und Klassifikation typischer Informatiksysteme sollten erlernt werden. Rechnerarchitektur (von-Neumann-Architektur), Rechnernetze (Client-Server-Architektur, Protokollbegriff), Betriebssysteme (Dienste, Zugriffsrechte) und die Daten- und Kontrollstrukturen von Algorithmen werden als Grundkonzepte der Informatik thematisiert. In die Digitalisierung und Codierung von Informationen zu Daten wird zielgerichtet eingeführt. Das Klassifizieren von Objekten (Zuordnung von Objekten zu vorgegebenen Klassen nach einer festen Vorschrift) und die Konstruktion objektbasierter Lösungen wird als wesentliche Strategie der Informatik erlernt (ohne Polymorphie). Informationsräume (Verzeichnisstruktur, Hypertext, Datenbank, Rechnernetz) werden strukturiert. Es werden Suchstrategien angewendet und mit der Bewertung von Informationen im Hinblick auf ihre Zuverlässigkeit verbunden. Die Schüler entwickeln schrittweise Verantwortung für selbstgestaltete und präsentierte Informationen. Die Sensibilisierung für Datenschutz und Datensicherheit wird um den Bereich der Informations- und Kommunikationssysteme erweitert.

2.1.3 Ziele und Schwerpunkte

2.1.3.1 Grundsätzliches

- Unterrichtsinhalte sind grundlegende Konzepte von Informations- und Kommunikationssystemen, die von hoher Allgemeingültigkeit und Dauerhaftigkeit sind, um den Schülern die Kompetenz zu vermitteln, sich auf die dynamischen Weiterentwicklungen einstellen zu können.
- Der zentrale inhaltliche Begriff ist "Information": Informationen müssen formal repräsentiert werden, wenn sie elektronisch verarbeitet werden sollen. Die Repräsentationen werden dabei verändert oder transportiert, um abschließend zur Informationsgewinnung wieder interpretiert zu werden.
- Konkrete Systeme, sei es nun Hard- oder Software, dienen immer nur als Medium, niemals als Ausgangspunkt didaktischen Handelns.

⁵ (Syn.) Anwendung, Applikation: in der Praxis werden die für bestimmte Funktionsbereiche geschaffenen Programme "Anwendungen" genannt.

Die Einführung in das Fach Informatik sollte altersgemäß und handlungsorientiert erfolgen, dabei aber keinesfalls ungenau oder unsystematisch. Teamarbeit läßt sich dabei in kleinen Projekten gut realisieren.

2.1.3.2 Einführung in die wichtigsten Begriffe und Denkweisen der Informatik

Erstellung und Präsentation von Dokumenten

Graphikprogramme: Der Umgang mit Zeichenprogrammen ermöglicht einen Einblick in typische Datenstrukturen von Vektor- und Pixelgraphik, mögliche Operationen darauf, einschließlich der Manipulationsmöglichkeiten, sowie der Abschätzung des Platzbedarfs.

Textverarbeitung: Die Schüler nehmen im Umgang mit diesem Standardwerkzeug z. B. Texte, Absätze, Zeichen als Objekte mit ihren Attributen und Methoden wahr.

Hypertext: Mit Hypertext-Sprachen werden Schüler auf sehr einfache Weise in die Strukturierung und Darbietung von Informationen eingeführt. Der Aufbau von Hypertextdokumenten sowie der Einbau von Verweisen und multimedialen Objekten lassen sich anschaulich realisieren.

Tabellenkalkulation: Die bereits bekannten Strukturen finden sich in den Zeilen, Spalten und Zellen als Objekte der Tabellenkalkulation wieder. Die Schüler lernen bei der Deklaration von Zellinhalten verschiedene Datentypen kennen sowie die Möglichkeit der Verweise auf andere Zellen. Die Anwendung elementarer Rechenfunktionen sowie einfache Präsentationstechniken runden die Einführung in dieses Standardwerkzeug ab.

Verwaltung und Transport von Dokumenten

Hierarchische Dateisysteme: Die Strukturierung und Ablage von Informationen in geeigneten Dokumenten führt zum Begriff der Datei als Container für Daten. Zur hierarchischen Verwaltung von Dateien bietet sich die Anlage von Ordnerhierarchien an. Die Schüler werden mit Operationen wie Löschen und Kopieren von Dateien vertraut.

Der Versand von Dokumenten mit Hilfe von elektronischer Post führt zu einem der wichtigsten Dienste im Internet, der E-Mail. Dieser Dienst basiert auf dem Client-Server Prinzip, das präzise Absprachen (Protokolle) für die Operationen in diesem Kommunikationssystem erfordert.

Ein weiterer Dienst, das WWW, ermöglicht die Publikation von Dokumenten im Internet. Das Auffinden solcher Dokumente setzt eine systematische Suche in Verzeichnisstrukturen und im Internet voraus.

Verarbeitung von Dokumenten und Automatisierung dieser Verarbeitung

Aus umgangssprachlichen Formulierungen von Verarbeitungsvorschriften für Objekte aus dem Bereich der Standardsoftware entstehen erste Algorithmen. Als grundlegende Bausteine werden Sequenz, Wiederholung und Auswahl erarbeitet. Dabei ergibt sich die notwendige Unterscheidung von Daten- und Programmdateien.

Die Kenntnis von algorithmischen Grundstrukturen ermöglicht es, für die Automatisierung

von Verarbeitungsvorgängen einfache Makros zu erstellen.

Die bei der Modellierung als notwendig erkannte und eingeführte Notation lässt es zu, auch die Funktionsweise von Rechenanlagen in einfachen Blockschaltbildern darzustellen.

2.1.3.3 Vertiefung der eingeführten Begriffe und Denkweisen

Die hier vorgeschlagenen Themenbereiche eröffnen verstärkt die Möglichkeit, Projekte durchzuführen, in denen die Schüler ihre Fähigkeiten zu Teamarbeit und Kommunikation erweitern können.

Prinzipien der Repräsentation von Information

Die Information stellt das zentrale Thema der Schulinformatik dar. Aufbereiten und Darstellen von Informationen, Verarbeiten, Transportieren, Rekonstruieren und Interpretieren von Repräsentationen bilden das Grundprinzip der Informationsverarbeitung.

Zur idealisierten, vereinfachten Beschreibung eines realen oder geplanten Systems bedient man sich eines Modells (Modellierung).

Die Erarbeitung einfacher Beschreibungstechniken geschieht an Beispielen aus der Erfahrungswelt der Schüler: Datenstrukturen (Texte, Tabellen, Graphiken, Dateien und Verzeichnisse), Verarbeitungsvorschriften (Algorithmen, Kontrollstrukturen, Struktogramme), Systembeschreibungen (Datenflussdiagramme, Zustandsübergangdiagramme).

Die Repräsentation der Information durch Sprache bedient sich eines vereinbarten Zeichensatzes und ist durch Syntax und Semantik geprägt, was die Unterschiede zwischen natürlichen und künstlichen Sprachen deutlich werden lässt.

Datenmodellierung

Die informelle Beschreibung des Systems einer Datenbank verlangt bei der Datenmodellierung die Entwicklung eines geeigneten Modells. Als Modelle werden Entity-Relationship-Diagramme (ER-Diagramme) verwendet. Die Datenmodellierung schafft Entitäten (Datenobjekte) und Klassen von Entitäten mit entsprechenden Attributen und Wertebereichen. Den Schwerpunkt bilden die Beziehungen zwischen den Entitätsklassen. Die Relationen werden mit Hilfe von Tabellen implementiert.

Die weiterführende Umsetzung von ER-Modellen in relationale Datenbanken führt u. a. zur Einführung von Primär- und Fremd-Schlüsseln sowie bei Berücksichtigung von Redundanz und Konsistenz zu den Normalformen 1-3.

Die Datenrepräsentation in Form von Relationen verschafft die Möglichkeit der Bearbeitung konkreter Fragestellungen mit diesen Relationen.

Die Schüler lernen Operationen auf Tabellen kennen sowie Abfragen und Berichte. Über die Datenmanipulation werden hier auch Fragen des Datenschutzes angesprochen.

Zustandsorientierte Modellierung und Algorithmen

Bei der Implementierung von Automaten durch imperative Programme werden die Grundbausteine von Algorithmen (Sequenz, Auswahl, Wiederholung) erarbeitet. Die Visualisierung kann durch Struktogramme geschehen.

Grundkenntnisse über die Funktionsweise imperativer Programme sind ein unverzichtbarer Baustein zum Verständnis der Funktionsweise elektronischer Informations- und Kommunikationssysteme. Diese Grundkenntnisse können nur erfolgreich vermittelt werden, wenn die Schüler selbst Programme schreiben, übersetzen lassen und den Ablauf verfolgen können.

Das Automatenmodell des von-Neumann-Rechners erfordert die Realisierung von Variablen und Verarbeitungsvorschriften durch Zustände von Speicherzellen.

Funktionale Modellierung

Bei der Modellierung komplexer Systeme erfolgt eine Aufteilung in Teilsysteme (Komponenten), wobei Datenflussdiagramme die Zusammenhänge veranschaulichen.

Imperative Programme werden modularisiert mit Hilfe von Prozeduren und Funktionen. Besonderes Augenmerk gilt dabei den Schnittstellen (Parameterübergabe).

Objektorientierte Modellierung

Objekte vereinen Datenstrukturen (Attribute) und Verarbeitungsvorschriften (Methoden, auch Operationen genannt). Ein Objekt wird dabei als Instanz seiner Klasse angesehen. Methoden in der objektorientierten Programmierung unterstützen drei grundsätzliche Techniken:

1. Kapselung
2. Vererbung
3. Polymorphismus.

Eine vollständige Auseinandersetzung mit der objektorientierten Modellierung, insbesondere mit Vererbung und Polymorphismus erfolgt in der Sekundarstufe II.

Rechenanlagen - Rechnernetze

Bei der Betrachtung der Konzeption von Rechenanlagen spielen neben dem Aufbau und der Funktionsweise von Computern und Computernetzen auch Konzeption und Planung von Netzwerken, Rechte- und Verzeichnisstrukturen, Sicherheitsstrategien eine wichtige Rolle.

Diese Konzeption in Rechnernetzen zielt auf Dienste im Internet, Informationssuche, Versand und Herunterladen von Dokumenten, sowie die Produktion und Veröffentlichung eigener Dokumente.

2.2 Für die Sekundarstufe II

2.2.1 Vorbemerkungen

Informatik ist von ihrem Wesen her eine stark abstrahierende Wissenschaft. In der Sekundarstufe II ist das Abstraktionsvermögen der Schüler so weit ausgeprägt, dass durch einen wissenschaftspropädeutischen Ansatz die Exaktheit und innere Konsequenz informatischer Arbeitsweisen den Schülern vermittelt werden können. Sie erfahren so die Verlässlichkeit, aber auch die prinzipiellen Grenzen informatischer Systeme.

Die in der Mittelstufe begonnene Untersuchung der Modellierungstechniken wird in der Oberstufe fortgesetzt und systematisiert. Neben komplexeren Modellbildungen und der Modellierung paralleler Vorgänge werden auch immer wiederkehrende, typische Datenstrukturen und deren Operationen analysiert. Die Unabhängigkeit der jeweiligen Datenstruktur von ihrer konkreten Implementierung führt dabei die Vorteilhaftigkeit der Schichtenbildung⁶ als allgemeinem Ansatz vor Augen.

Durch die Verwendung geeigneter Beispiele werden zusammen mit den fachtheoretischen Inhalten auch die weitreichenden Auswirkungen der Informations- und Kommunikationstechniken auf die Gesellschaft verdeutlicht. Die positiven und negativen Implikationen automatisierter Informationsverarbeitung auf alle Anwendungsbereiche werden herleitbar. So erkennen die Schüler die große Verantwortung der Informatik für künftige gesellschaftliche Entwicklungen.

2.2.2 Themen

Die für die Oberstufe relevanten Lerninhalte sind im Folgenden ohne Gewichtung aufgelistet. Auch eine zeitliche Anordnung kann nicht gegeben werden, da die Inhalte zum Teil überlappend sind (z. B. bieten die Darstellung von Graphen und deren Algorithmen auch eine gute Möglichkeit, Laufzeitordnungen und Beweistechniken zu diskutieren; Methoden des Softwareengineerings können bei allen größeren Themen dargestellt und angewandt werden).

Für die aufgeführten Lerninhalte werden Einbettung und wesentliche Lernziele dargestellt.

2.2.2.1 Modellierungen

Die bereits bekannten Modellierungsschemata werden erweitert und um abstraktere und / oder komplexere Schemata ergänzt.

Objektorientierte Modellierung

Betrachtet man die Objekte eines Graphikdokuments, so stellt man fest, dass es nicht nur Klassen gleicher Objekte gibt (Rechtecke, Kreise, Ellipsen, etc.), sondern dass es zwischen

⁶ Methode, um von einem hohen Abstraktionsniveau zu den Details zu kommen.

einigen dieser Klassen gemeinsame Eigenschaften gibt, die sie mit anderen Klassen (Strecke, Gerade, etc.) nicht teilen, z. B. das Attribut 'umgebendesRechteck' oder die Methode 'FuelleFlaeche'. Die vorgenannten Klassen werden z. B. aus der Klasse der geometrischen Objekte abgeleitet, wobei als unterscheidende Eigenschaften "hat Fläche" bzw. "hat keine Fläche" zur Bildung einer Zwischenschicht herangezogen wird.

Klassen können eine hierarchische Struktur aufweisen. Bezogen auf die Klassen an der Spitze der Hierarchie besitzen die weiter unten stehenden Klassen zusätzliche Attribute und / oder zusätzliche Methoden. Bestehende Methoden können modifiziert werden. Diese Strukturen werden durch die Mechanismen der Vererbung und des Polymorphismus beschrieben. Hier zeigt sich erst die ganze Mächtigkeit objektorientierter Modellierung. Damit wird deutlich, dass dieser Ansatz gegenüber der rein imperativen Modellierung nicht nur eine andere Sichtweise darstellt, sondern deutlich weiterführende Möglichkeiten bietet.

Endliche Automaten / Kellerautomaten

Bei einer Liftsteuerung fällt die Entscheidung, wie auf einen Knopfdruck zu reagieren ist, allein aus dem aktuellen Zustand (in welchem Stockwerk ist der Lift, Tür offen / zu etc.) und der Eingabe (welcher Knopf wurde gedrückt), die Vorgeschichte (wo war der Lift vorher) wird nicht berücksichtigt. Das abstrakte Modell dieser Situation (Entscheidung aus Zustand und Eingabe), der deterministische endliche Automat, lässt sich auf viele Beispiele anwenden, z. B. auch effizient auf die Suche nach einem bestimmten Wort (z. B. "gegeben") in einem Text. Die Textzeichen sind die Eingabezeichen, der zu suchende Text bestimmt die Zustände des Automaten. Bei der Suche nach Wörtern mit gleichem Anfang ("gegeben", "gegenüber") kommt man sehr schnell zum Modell des nichtdeterministischen Automaten. Diese (scheinbare) Erweiterung führt den Schülern die Tatsache vor Augen, dass eine intuitive Analyse ("das sieht man doch, dass der mehr kann") nicht immer ausreicht. Erst die exakte Analyse der Situation zeigt, dass beide Automatenarten gleichwertig sind (Myhill-Teilmengenalgorithmus). Etliche Aufgabenstellungen sind aber mit Hilfe nichtdeterministischer Automaten wesentlich leichter zu beschreiben. Die Untersuchung der von einem endlichen Automaten verarbeitbaren Eingabefolge führt zum Begriff der akzeptierten Sprache und deren Eigenschaften und Einschränkungen. Die Frage nach Automaten, die mathematische Klammerterme erkennen können, führt schließlich zur Erweiterung zum Kellerautomaten und zur damit verbundenen Erweiterung der akzeptierten Eingabefolgen (Sprachen). Hier ergibt sich ein Einblick in die Sprachhierarchie nach Chomsky sowie die mit komplexeren Sprachen verbundenen Analyseprobleme. Es kann auch aufgezeigt werden, warum menschliche Sprache für den Computer nur in eingeschränkten Fällen auswertbar ist (Übersetzungsprogramme).

2.2.2.2 Verteilte Systeme

Untereinander verbundene Rechner (verteilte Systeme, als deren größter Repräsentant das Internet) bringen weitere Bereiche von Problemstellungen mit sich und damit die Notwendigkeit der Kenntnis adäquater Modellierungstechniken.

Parallele Prozesse

Wenn zwei Benutzer im Internet gleichzeitig auf den gleichen Server zugreifen, um dort Daten zu ändern (z. B. Flugbuchung in zwei Reisebüros für den gleichen Flug), muss unter anderem sichergestellt werden, dass nicht die Änderung des einen Benutzers die des anderen zerstört ("Bekommen beide den gleichen Platz?", "Löscht die Buchung des zweiten die erste Buchung?"). Allgemeiner ausgedrückt treten hier die Fragen nach der Nutzung gemeinsamer Ressourcen (Interaktion) und damit der Synchronisation solcher nebenläufiger Prozesse auf. Hierzu sind eigene Modellierungstechniken wie Interaktionsdiagramme (Methode zur Beschreibung des Verhaltens eines informatischen Systems) oder Petrietze nötig, die insbesondere die (zeitliche) Datenabhängigkeit der einzelnen Prozesse darstellen. Sie sind unabhängig von der Modellierung der einzelnen Prozesse selbst.

Gleichzeitig werden korrekte Synchronisationsmechanismen benötigt (Semaphore, Nachrichtenaustausch usw.), die insbesondere den verändernden Zugriff auf gemeinsame Ressourcen wie gemeinsam genutzten Speicher bzw. gemeinsam genutzte Dateien oder die Verfügbarkeit der Ergebnisse regeln.

Die Erhaltung der Datenkonsistenz ("Stimmigkeit" der Daten in einer Datenbank; Transaktionskonzept⁷) und die Fragen der Ausfallsicherheit sind weitere wesentliche Aspekte verteilter Systeme.

Kommunikation

Sind Aufgaben zur ihrer Lösung auf verschiedene Rechner verteilt, bedarf es geeigneter Mechanismen für den Datenaustausch. Diese Mechanismen (Kommunikationsprotokolle, wie z. B. TCP/IP) müssen zum einen die Struktur der zu übertragenden Daten beschreiben (Datenformat) und die korrekte Übertragung gewährleisten (Fehlerkorrektur, Prüfsummen etc.), zum anderen müssen sie aber auch das Herstellen des Kommunikationsweges definieren (Rechnerbezeichnungen, Datenweiterleitung).

Falls die beteiligten Rechner unterschiedlich sind, müssen diese Protokolle auch eine universelle Repräsentation der Information beinhalten (spezielle Bild-, Ton- und Videoformate, aber auch genormte Zahlen- und Zeichendarstellung), die auf allen beteiligten Rechnern gleich verstanden wird (gleiche Informationssemantik repräsentiert).

Neben der Kontrolle der Zugangsberechtigungen wird vor allem das Problem der Abhörsicherheit (Kryptologie) relevant, sobald die Übertragungswege allgemein zugänglich sind und sensible Informationen übertragen werden.

Client / Server

Einen wichtigen Sonderfall verteilter Systeme stellen Client-Server-Situationen dar, bei denen ein Rechner (Server) Dienstleistungen für eine Vielzahl anderer Rechner (Clients) an-

⁷ Beim Transaktionskonzept werden die einzelnen Aktionen eines größeren Vorhabens (einer Transaktion) nur alle gemeinsam oder überhaupt nicht wirksam. Z. B. wird eine Überweisung nur dann ausgeführt, wenn sowohl der Abbuchungsvorgang vom Absenderkonto als auch der Buchungsvorgang beim Empfängerkonto ausgeführt werden können.

bietet (z. B. World-Wide-Web, News- oder FTP-Server im Internet). Zur Nutzung der Serverdienste sind genaue, aber auch möglichst umfassende und flexible Absprachen (Protokolle) über die Struktur der zu verarbeitenden Daten, der zur Verfügung stehenden Operationen sowie der Zugangsmöglichkeiten (Authentifizierung) notwendig.

2.2.2.3 Typische Datenstrukturen

Bei der Modellierung komplexerer Situationen treten bestimmte (höhere) Datenstrukturen immer wieder auf. Die Schüler lernen, dass derartige Strukturen allein über ihre Operationen und deren Eigenschaften beschrieben und damit festgelegt werden können. Eine Kenntnis der eigentlichen Implementierung ist auf dieser Modellierungsebene nicht nötig. Die Modellierung erfolgt in Schichten, von denen jede eine klare Schnittstelle zur darüber- bzw. darunter liegenden Schicht hat, die aber sonst völlig unabhängig voneinander sind.

Listen

Soll in einem Programm eine von Verwendung zu Verwendung stark schwankende Anzahl von Objekten bearbeitet werden (z. B. Texte von einer Seite bis mehreren hundert Seiten bei der Textverarbeitung), so kann der dafür nötige Speicherplatz nicht bereits bei der Erstellung des Programms festgelegt werden; zur Modellierung solcher Aufgaben werden Datenstrukturen benötigt, die ihre Größe der zu verarbeitenden Datenmenge dynamisch anpassen können. Grundlage aller dynamischen Datenstrukturen und Basis für viele der höheren Datenstrukturen sind Listen von Elementen. Verschiedene Implementierungen (auch sortierte / unsortierte Anordnung der Elemente) des gleichen Datentyps können leicht miteinander verglichen werden. Genau so leicht kann aufgezeigt werden, dass dieser Datentyp als Basis zur Realisierung anderer Datenstrukturen (Stapel, Warteschlange, Menge) dienen kann.

Mengen

Mengen irgendwelcher Dinge treten an sehr vielen Stellen auf. Einige Grundoperationen sind aus der Mathematik bekannt. Weitere Operationen (z. B. systematisches Durchlaufen aller Elemente einer Menge) werden neu definiert. Bei der Diskussion der Darstellung von Mengen kann aufgezeigt werden, dass die Auswahl der effizientesten Implementierung von den gegebenen Randbedingungen abhängt. Hier zeigt sich auch deutlich die Notwendigkeit, zwischen den (implementierungsunabhängigen) Eigenschaften der Datenstruktur und den Randbedingungen für die Implementierung zu unterscheiden.

Graphen und Bäume

Ob Computernetzwerke, Straßenkarten, Wasser-, Strom- oder Telefonleitungssysteme, aber auch abstrakte Dinge wie der Datenfluss in Programmen oder die Lagebeziehung der konstruierten Gegenstände in einem CAD-Programm, alle diese Situationen lassen sich gut als Wege ("Kanten") zwischen Verzweigungsstellen ("Knoten") modellieren. Die so entstehenden Graphen können gerichtete Wege ("Einbahnstraßen") oder ungerichtete Wege ("Hin- und Rückweg möglich") enthalten, die Kanten können gewichtet sein (als Modell für Streckenlängen, Reisezeiten, Transportkosten, Risikoaufwand etc.) oder einfach nur die Existenz von Beziehungen bedeuten. Beispiele wie Weglängenermittlung, Leitungskostenbe-

rechnung, Erreichbarkeitsprüfung etc. zeigen die vielfältigen Anwendungsmöglichkeiten. Die Untersuchung der typischen Aufgabenstellungen (Suche nach Zusammenhängigkeitskomponenten, Suche kürzester Wege etc.) bietet Ansatzpunkte für Effizienzbetrachtungen bzw. Fragen der Korrektheit.

Zur Modellierung von Entscheidungsstrukturen eignen sich Bäume, die man als zyklensfreie Graphen betrachten kann.

2.2.2.4 Aufwandsabschätzungen

Die Untersuchung des Laufzeitaufwands von Algorithmen (Laufzeitordnung) zeigt den Schülern eine weitere Grenze informatischer Systeme auf. Zum einen wird deutlich, dass bei einer Lösung, bei der die notwendige Rechenzeit exponentiell von der Anzahl der zu bearbeitenden Gegenstände abhängt (exponentieller Aufwand), die Verdopplung (oder auch Verzehnfachung) der Rechnerleistung den Umfang behandelbarer Daten nur unwesentlich vergrößert. Gleichzeitig wird gezeigt, dass Untergrenzen für die Laufzeitordnung vieler typischer Probleme existieren, die nicht unterschritten werden können.

Die Ermittlung der Laufzeitordnung von Sortieralgorithmen ist ein einfaches Einführungsbeispiel in diesen Themenkreis. Ein sehr breites Feld für diese Untersuchungen bietet der Vergleich der verschiedenen Listen- und Mengenimplementierungen. Bei der Analyse der Wegesuchaufgaben in Graphen zeigt sich nicht nur eine sehr breite Streuung der Laufzeitordnungen, sondern auch die Notwendigkeit für eine exakte mathematische Behandlung der jeweiligen Problemstellung.

2.2.2.5 Spezifikation / Verifikation

Hier begegnen die Schüler der Nahtstelle zwischen Modell und der modellierten Realität. Die Schüler erkennen die Notwendigkeit einer exakten Definition des Modells als Voraussetzung für gesicherte Aussagen (z. B. Korrektheit, Vollständigkeit) über die aus dem Modell erstellte Implementierung. Gleichzeitig wird deutlich, dass nicht alle in der realen Welt vorhandenen Strukturen mathematisch formalisiert und damit modelliert werden können. (Es ist z. B. nicht möglich, einen vollständig strukturierten Restaurantführer zu konzipieren, hier sind immer unstrukturierte Beschreibungstexte nötig.) Ebenfalls wird erkennbar, dass bei der Abbildung der Realität auf das Modell (Anlage des Modells) und damit auch der Rückinterpretation der Modellergebnisse in die reale Welt die subjektive Sicht des Modellbildenden eine entscheidende Rolle spielt.

Spezifikation

Das reale Problem wird gemäß seiner Grundstruktur mittels eines oder mehrerer Modellierungsschemata in seine Teilprobleme zergliedert. Diese Teilprobleme werden gegebenenfalls weiter zerlegt (Top-Down-Entwicklung), bis alle zu behandelnden Aufgaben auf Grundoperationen zurückgeführt sind. Daraus ergeben sich die zu beschreibenden (spezifizierenden) Daten und Operationen. Die Daten werden als zusammengesetzte Strukturen von bereits spezifizierten Grunddatentypen wie Zahlen, Zeichen, Mengen, usw. angegeben. In der Schule werden die Eigenschaften der Operationen informell beschrieben. Beispielhaft

kann auch an geeigneter Stelle eine formale Spezifikation gezeigt werden. Dabei werden die Operationen als Funktionen auf den verwendeten Datentypen vereinbart, ihre Eigenschaften können durch formale Beschreibung der Funktionsterme sowie durch die formale Beschreibung der Zusammenhänge zwischen den Ergebnissen der einzelnen Operationen spezifiziert werden. Bei der Spezifikation der Operationen muss Widerspruchsfreiheit gesichert werden (sonst ist die Modellbildung fehlerhaft). Das Modell muss aber nicht vollständig spezifiziert werden, wenn das reale Problem Freiheiten erlaubt.

Verifikation

Eine systematische Untersuchung informatischer Beweismethoden ist in der Sekundarstufe II weder leistbar noch anstrebenswert. Die Schüler sollen aber einen Einblick in eine Beweismethode bekommen, um so erkennen zu können, dass die Korrektheit von Programmen im Hinblick auf eine mathematisch exakt beschriebene Aufgabenstellung beweisbar und überprüfbar ist.

Einstieg kann der Nachweis der Korrektheit imperativer Programme durch die schrittweise Zusicherung formaler Bedingungen (Hoare-Kalkül) sein, allerdings sollte man sich auf die Beweise kurzer Programmstücke beschränken.

Zur Verifikation eines Programmes gehört neben dem Nachweis eines korrekten Resultats auch der Nachweis, dass das Programm die Lösung auch erreicht (keine "Endlos-Schleifen"). Methoden für Terminierungsbeweise lassen sich an überschaubaren Programmstücken nachvollziehen.

Halte-Problem

Die Frage nach der Terminierung von Programmen lässt sich auf die allgemeinere Fragestellung nach der Berechenbarkeit gegebener Probleme ausweiten. In der Schule kann diese Problematik nur angerissen werden. Die wichtigste Konsequenz, dass die Frage nach der Terminierung eines Programms im Allgemeinen nicht entschieden werden kann (Halte-Problem), sollten die Schüler kennen.

2.2.2.6 Methoden der Softwareentwicklung

Zentrale Anliegen moderner Softwareentwicklung sind das Auffinden geeigneter Modellierungen für das gegebene Problem, Absicherung der Konsistenz und Korrektheit der Modellierung und die fortwährende Nachführung der Modellierung bei sich verändernden Problemstellungen.

Bei einem hinreichend großen Softwareprojekt (mindestens 16 Wochenstunden) ergibt sich die Anwendung von Zerlegungsmethoden (Top-Down-Entwicklung, schrittweise Verfeinerung) für die Schüler genau so organisch wie die Anwendung verschiedener Modellbildungen und die Notwendigkeit zu sauberer Spezifikation. Eine abschließende Analyse des Projektverlaufs macht den Schülern die angewandten Methoden bewusst, ohne diese "formal" zu behandeln (siehe auch Abschnitt 2.3, insbesondere 2.3.2).

2.2.3 Schlussbemerkungen

Die früher allgemein übliche Behandlung der technischen Realisierung informatischer Systeme (Schaltalgebra und deren elektronische Umsetzung, Rechnerarchitektur, aber auch Assemblerprogrammierung etc.) sind in diesem Ansatz bewusst weggelassen. Ebenso wird auf genauere Untersuchungen rechnerinterner Aufgaben (insbesondere Betriebssystemaufgaben) verzichtet. Die Diskussion dieser Themen bringt für die in Abschnitt 2.2.2 angesprochenen Inhalte keine zusätzlichen Erkenntnisse, da es sich nur um weitere, nach den gleichen formalen Prinzipien modellierte Schichten eines informatischen Systems handelt.

Derartige Themen kann – und sollte – der interessierte Schüler in geeigneten Wahlunterrichtskursen erfahren. Grundkurse des Ergänzungsprogramms sind dafür ebenfalls ein möglicher Rahmen.

2.3 Bemerkungen zur Methodik

Viele Techniken der Informatik wurden speziell für die Bearbeitung komplexer Aufgabenstellungen entwickelt. Um diese Techniken an "realen", d. h. genügend komplexen Beispielen erfahren und einüben zu können, bedarf es geeigneter Unterrichtsformen. Durch diese Unterrichtsformen werden den Schülern zusätzlich allgemein wertvolle Sichtweisen und Kenntnisse vermittelt, die sie in anderem Zusammenhang nicht gewinnen könnten.

2.3.1 Teamarbeit

Die Fähigkeit zur Zusammenarbeit im Team ist eine wichtige Voraussetzung zur Bewältigung umfangreicher Aufgaben und Projekte. In der Schule wird die Fähigkeit zur Teamarbeit nur an sehr wenigen Stellen eingeübt (z. B. Übungsgruppen in Physik bzw. Chemie). Bei der Modellierung und Realisierung der in der Oberstufe behandelten Themen eignen sich die umfangreicheren Aufgaben sehr gut für die Behandlung sowohl in konkurrierenden als auch in sich ergänzenden Teams. Dabei kann sowohl die Arbeit in gleichberechtigten Teams als auch in hierarchischen Teams (definierter Leiter) ausprobiert und reflektiert werden. Arbeiten mehrere Teams ergänzend an der gleichen Aufgabe, kann zusätzlich die Wichtigkeit geeigneter Informationsstrukturen zur Koordination zwischen den Teams erfahren werden. Bei Projekten im Bereich der Informatik wird hier zusätzlich der Aspekt klarer und ausreichender Spezifikationen deutlich.

2.3.2 Themenzentrierter Unterricht

Themenzentrierter Unterricht, d. h. die Behandlung vieler einzelner Lernziele im Rahmen eines großen Themas, bringt den Vorteil, dass das Gesamtthema eine permanente Motivation für die Beschäftigung mit den für sich allein gestellt oft weniger motivierenden Teilproblemen liefert. Darüber hinaus lernen die Schüler aber auch, Teillösungen und Einzelwissen in einen großen Rahmen einzuordnen und in flexibler Weise anzuwenden, sie ge-

wöhnen sich an das Denken in großen Strukturen und Zusammenhängen.

Themenzentrierter Unterricht ist aufgrund seiner komplexen Aufgabenstellungen insbesondere auch gut geeignet, Methoden des modernen Softwareengineerings in der Anwendung zu erlernen ("Learning by Doing").

3 Hinweise zur Umsetzung

3.1 Voraussetzungen

Grundsätzlich können Computer in allen Fächern und in allen Jahrgangsstufen eingesetzt werden. Jeder Einsatz des Computers im Unterricht setzt aber ein **fachunspezifisches Handhabungswissen** voraus, auf dem erst der fachspezifische Einsatz fortgeführt werden kann. Die Vermittlung dieses Wissens kostet Unterrichts-Zeit, die kein Fach abgeben kann bzw. will. Außerdem muss dieses Handhabungswissen **dann zur Verfügung stehen, wenn es gebraucht wird; bei den weiterführenden Schulen also am Anfang der Jahrgangsstufe 5!**

Für die Vermittlung dieses Handhabungswissens gibt es noch keine traditionelle Arbeitsverteilung, und es wird sie auch in der näheren Zukunft nicht geben, denn Inhalte und Unterrichtskonzepte dafür müssen sich nach der tatsächlich vorhandenen Ausstattung und den tatsächlich vorhandenen Vorkenntnissen der Schülerinnen und Schüler richten. Beide sind derzeit aber großen Veränderungen unterworfen, die es erforderlich machen, diese Inhalte und Konzepte jährlich schulintern neu zu überprüfen und anzupassen. Der Computereinsatz im Unterricht bedarf also einer sorgfältigen und variablen Planung und pädagogischen Betreuung **vor Ort**: "Wer macht das denn heuer bei uns wann und wie?"

Bei der Einführung der neuen Schüler in die Computernutzung gilt es drei Themenbereiche zu planen:

- Was müssen die neuen Schüler können?
Die nötigen Handhabungsfertigkeiten ergeben sich aus der an der Schule zur Zeit tatsächlich vorhandenen und in den Eingangsklassen eingesetzten Hard- und Software.
- Wie sollen sie diese Fertigkeiten erlernen?
Hier ist zu entscheiden, ob die Einführungsveranstaltungen im Rahmen des stundenplanmäßigen Unterrichts oder im Rahmen von Projekttagen erfolgen sollen.
- Wer soll die Veranstaltungen durchführen?
Hier ist entsprechend der Lehrverteilung zu klären, ob die jeweiligen Fachlehrer der Klasse die Einführungen selbst durchführen können und wollen, oder ob sie von anderen Lehrkräften, z. B. durch das ITB-Team durchgeführt werden sollen. (Zur Einführung eines ITB-Teams siehe zum Beispiel <http://www.isb.bayern.de/bf/kont99.htm>)

Es ist empfehlenswert, gleich in den ersten Unterrichtstagen, in denen der Unterrichtsbetrieb noch nicht voll angelaufen ist, die neuen Schüler in den Umgang mit den schuleigenen Computern einzuführen (z. B. durch Projekttag). Der Computer steht damit allen Fächern von Anfang an als Werkzeug zur Verfügung, ohne dass die zeitliche Belastung dieser Wissensvermittlung einem einzelnen Fach aufgebürdet wird. Sowohl organisatorisch als auch inhaltlich gehört diese Einführung zur informationstechnischen Grundbildung (ITG), nicht zum Fach Informatik! Im Gegenteil: Um sein Ziel – den Schülerinnen und Schülern sprach-

liche und algorithmische Kompetenz zu vermitteln – zu erreichen, muss auch der Unterricht in Informatik auf diese Grundfertigkeiten im Umgang mit informationsverarbeitenden Systemen zurückgreifen können.

Zur Unterstützung der Schulen bei der Planung und Durchführung der genannten Arbeiten hat am Staatsinstitut für Schulpädagogik und Bildungsforschung (ISB) der Arbeitskreis "Hinweise zur Umsetzung der ITB" entsprechende Materialien erstellt; sie sind unter www.isb.bayern.de/bf/itb/module im Internet abrufbar.

3.2 Hauptschule

Die folgenden Ausführungen beziehen sich auf den Fachlehrplan Informatik im **Lehrplan für die Hauptschule in Bayern** nach der Bekanntmachung des Bayerischen Staatsministeriums für Unterricht, Kultus, Wissenschaft und Kunst vom 29. Oktober 1997 Nr. IV/3 - S 7410/2 - 4/141 584.

Der Lehrplan für die Hauptschule weist in Abschnitt I.4.4 die "Einführung in Grundlagen der Informations- und Kommunikationstechniken" als Schlüsselqualifikation und in II.2.19 die "informationstechnische Bildung" (ITB) als eine der verpflichtend durchzuführenden fächerübergreifenden Bildungs- und Erziehungsaufgaben **für jeden Schüler** aus. Beide Anliegen sind deshalb dem Pflicht- und Wahlpflichtunterricht zuzuordnen; das Wahlfach Informatik, das in aller Regel nicht von allen Schülern gewählt wird, steht dafür nur ergänzend und vertiefend zur Verfügung.

In der Jahrgangsstufe 8 – dem Beginn des Wahlfaches Informatik – haben die heutigen Schüler bereits langjährige unterrichtliche und außerunterrichtliche Erfahrungen im Umgang mit dem Computer. Aufgabe des Faches Informatik ist es, diese – meist unterschiedlichen – Erfahrungen aufzugreifen, zu systematisieren und von einem höheren Standpunkt aus zu reflektieren.

Im Folgenden sind aus dem Fachlehrplan Informatik die Grobziele wiedergegeben und mit Kommentaren bzw. Interpretationen aus der Sicht der heutigen Didaktik versehen. Der gesamte Lehrplan ist unter <http://www.isb.bayern.de/ghs/lp-leit.htm> einsehbar.

1 Grundlagen der Informationstechnik

Jahrgangsstufe 8	Jahrgangsstufe 9	Jahrgangsstufe 10
8.1.1 Umgang mit der schuleigenen Computeranlage	9.1.1 Bestandteile und Arbeitsweise einer Computeranlage	10.1.1 Abstimmung der Rechnerkomponenten auf besondere Anwendungsgebiete
8.1.2 Die funktionelle Gliederung und Arbeitsweise der Computeranlage	9.1.2 Umgang mit graphischen Bedienoberflächen	10.1.2 Umgang mit dem Betriebssystem und der Bedienoberfläche

Jahrgangsstufe 8	Jahrgangsstufe 9	Jahrgangsstufe 10
8.1.3 Umgang mit graphischen Oberflächen	9.1.3 Datensicherung	
8.1.4 Gestaltung des Arbeitsplatzes		

Die (in Jahrgangsstufe 8 etwa 14-jährigen) Schüler haben vor dem Beginn des Wahlfaches Informatik die schuleigenen Computer bereits diverse Male in anderen Fächern eingesetzt; vielfach haben sie auch zu Hause oder bei Freunden Zugang zu Computern, die zum Teil sogar leistungsfähiger sind als die in der Schule. Hier gilt es, diese unterschiedlichen Vorerfahrungen einerseits für den Unterricht nutzbar zu machen (versierte Schüler lehren die anderen, um in kurzer Zeit alle Schüler auf den nötigen Wissensstand zu bringen), andererseits müssen die Vorerfahrungen systematisiert und reflektiert werden.

2 Informationsverarbeitung

Jahrgangsstufe 8	Jahrgangsstufe 9	Jahrgangsstufe 10
8.2.1 Darstellung und Bearbeitung von Informationen	9.2.1 Darstellung und Bearbeitung von Informationen	10.2.1 Darstellung und Bearbeitung von Informationen
8.2.2 Beschaffung und Auswertung von Informationen	9.2.2 Beschaffung und Auswertung von Informationen	10.2.2 Beschaffung und Auswertung von Informationen
8.2.3 Aus- und Weitergabe von Informationen	9.2.3 Aus- und Weitergabe von Informationen	10.2.3 Aus- und Weitergabe von Informationen

Die hier angegebenen Verfahren wurden zumeist bereits in anderen Fächern als Hilfsmittel und Werkzeug eingesetzt. Im Fach Informatik gilt es, diese Einsätze unter informatischen Gesichtspunkten zu reflektieren, zu systematisieren und evtl. vorhandene Lücken zu schließen.

3 Steuern von Abläufen

Die Verwendung von Software, die auch in anderen Fächern zum Einsatz kommt (z. B. Makrosprache von Standardsoftware oder Autorensysteme) ermöglicht, die informatischen Inhalte an authentischen Beispielen (anderer Fächer) zu erlernen und ergibt vorteilhafte Synergieeffekte. Bei der Auswahl der Beispiele ist auch die Interessenlage der Mädchen zu beachten.

4 Veränderungen im beruflichen, gesellschaftlichen und privaten Bereich

Jahrgangsstufe 8	Jahrgangsstufe 9	Jahrgangsstufe 10
8.4.1 Computer in der Arbeitswelt	9.4.1 Computer und Umwelt	10.4.1 Computer in der Freizeit

Jahrgangsstufe 8	Jahrgangsstufe 9	Jahrgangsstufe 10
8.4.2 Computer in der Freizeit	9.4.2 Computer und Gesellschaft	10.4.2 Manipulation von Daten
8.4.3 Schutz des Urheberrechts	9.4.3 Computer in der Freizeit	
	9.4.4 Datenschutz	

Diese Themen sind jeweils auch Teil der informationstechnischen Bildung, der Medienerziehung, der beruflichen Orientierung bzw. der Freizeiterziehung; sie sind dementsprechend auch in Pflicht- und Wahlpflichtfächern verankert. Es ist fächerübergreifend sicher zu stellen, dass alle Schüler – und nicht nur die Teilnehmer am Wahlfach Informatik – unterrichtet werden.

3.3 Realschule (4-stufig bzw. 6-stufig)

Die folgenden Ausführungen beziehen sich auf die Fachlehrpläne Informatik im **Lehrplan für die Realschule in Bayern** nach der Bekanntmachung des Bayerischen Staatsministeriums für Unterricht, Kultus, Wissenschaft und Kunst vom 8. Juli 1993 Nr. V/2 - VIII/5 - S 6410 - 10/91 284¹ (**R4**) bzw. aufgrund KMS vom 14. April 1994 Nr. V/2 - S 6642 - 10/172 527/93^{II} (**R6**).

Unterrichtsinhalt sind grundlegende Konzepte von Informations- und Kommunikationssystemen, die von hoher Allgemeingültigkeit und Dauerhaftigkeit sind, um den Schülerinnen und Schülern die Kompetenz zu vermitteln, sich auf die dynamischen Weiterentwicklungen einstellen zu können.

Die Information und ihre Darstellung sollten zum zentralen Inhalt der Schulinformatik werden.

Die Schüler sollen unterschiedliche Modellierungsmöglichkeiten kennen lernen und sie mit verschiedenen Standard- oder Programmier-Werkzeugen realisieren.

Konkrete Systeme, sei es nun Hard- oder Software, dienen immer nur als Medium, niemals als Ausgangspunkt didaktischen Handelns.

Die Stundentafel für die 4-stufige Realschule sieht für das Wahlpflichtfach Informatik in den Jahrgangsstufen 8 - 10 jeweils zwei Wochenstunden vor. In der 6-stufigen Realschule wird das Wahlpflichtfach Informatik nach dem derzeit geltenden Lehrplan nur in den Jahrgangsstufen 9 und 10 mit jeweils zwei Wochenstunden unterrichtet.

Im Folgenden wird versucht, für die beiden derzeit gültigen Lehrpläne zu einzelnen Lernzielen aktuelle didaktische Sichtweisen anzubieten.

Die beiden Lehrpläne sind unter <http://www.isb.bayern.de/bf/isbl/lps/rs/lpsrs.htm> einsehbar.

Grundlagen der Informatik (Inf 8.1)

Denkbar ist der Einstieg mit Kara, einem graphisch programmierbaren Marienkäfer, (siehe auch Abschnitt 4.4.2), der bereits alle Begriffe des Lernziels Inf 8.2 propädeutisch anspricht.

Algorithmische Grundelemente und ihre Umsetzung in eine Programmiersprache (Inf 8.2)

Bei der Modellierung ermöglicht eine übergreifende Struktur, bereits bekannte Begriffe aus dem Fach Textverarbeitung wieder stärker ins Bewusstsein dringen zu lassen. Objekte, ihre Attribute und Operationen (Methoden) finden sich in allen einsetzbaren Werkzeugen (hier: Tabellenkalkulation bzw. Programmierwerkzeug).

Eine graphisch orientierte Programmierumgebung mit objektorientierter Ausprägung trägt neuen didaktischen Ansätzen Rechnung und eröffnet verstärkt die Möglichkeit, sowohl altersgemäße als auch handlungsorientierte Problemstellungen aufzugreifen.

Tabellenkalkulation und Diagrammerstellung (Inf 8.3)

Die Präsentation von Informationen als wichtiger Bestandteil der Kommunikations- und Mediengesellschaft kann hier zu einem frühen Zeitpunkt erfahren werden.

Die Tabelle findet sich hier als grundlegendes Objekt wieder. Andere Anwendungsbereiche sind die Standardwerkzeuge Textverarbeitung, Datenbank und HTML.

Prozeduren und Funktionen in einer Programmiersprache (Inf 8.4)

Die sowohl beim Umgang mit dem graphisch orientierten Programmierwerkzeug als auch beim Standardwerkzeug Tabellenkalkulation gleich zu Beginn verwendeten Prozeduren und Funktionen erlauben eine kreative Begegnung und Auseinandersetzung mit diesem Bereich. Die Schüler erfahren so sehr früh die Wichtigkeit der Modultechnik bei der Softwareentwicklung.

Darstellung von Programmen, Zahlen und Zeichen (Inf 9.1)

Hier lassen sich die Prinzipien der Repräsentation von Information verdeutlichen. Neben der Aufbereitung und Darstellung von Informationen als Grundprinzip der Informationsverarbeitung tritt die Repräsentation durch Sprache, wie sie sich in Zeichensatz, Syntax und Semantik zeigt. Auf diese Weise lassen sich auch die Unterschiede zwischen natürlichen und künstlichen Sprachen herausarbeiten.

Die Kenntnisse der Schüler über einige Binär-Codes ermöglichen es ihnen, neue Betrachtungsweisen bei der Datenfernübertragung zu erschließen.

Logische Funktionen und binäre Schaltungen (Inf 9.2)

Insbesondere die alternative Wahlmöglichkeit im Lehrplan der R6 zwischen den logischen Funktionen und der Tabellenkalkulation sollte verstärkt aktuelle didaktische Sichtweisen berücksichtigen (siehe auch Abschnitt 2.2.3).

Komplexe Datentypen (Inf 9.3)

Da die beiden strukturierten Datentypen Feld und Verbund eine Vielzahl von Eigenschaften aufweisen, lässt sich auch hier die objektorientierte Sicht ansprechen.

Verarbeitung großer Datenmengen (Inf 9.4)

Hier ist bei der Datenmodellierung an die Umsetzung von Entity-Relationship-Modellen in relationale Datenbanken gedacht. Die Begriffe Objekt, Tabelle, Spalte und Zeile (siehe Tabellenkalkulation) werden im Zusammenhang mit Datentypen und Datensätzen wieder aufgegriffen. Die Schüler lernen Operationen auf Tabellen kennen sowie Abfragen und Berichte. Über die Datenmanipulation werden hier auch Fragen des Datenschutzes angesprochen.

Standardprogramm zur Arbeit mit einer Datenbank (Inf 10.1)

Die informelle Beschreibung des Systems einer Datenbank verlangt bei der Datenmodellierung die Entwicklung eines geeigneten Modells. Die Tabelle als Relation wird mit Hilfe von Entity-Relationship-Diagrammen implementiert. Die Datenmodellierung schafft Entitäten (Datenobjekte) und Klassen von Entitäten mit entsprechenden Attributen und Wertebereichen. Den Schwerpunkt bilden die Beziehungen zwischen den Entitätsklassen.

Die weiterführende Umsetzung von ER-Modellen in relationale Datenbanken führt u. a. zur Einführung von Primär- und Fremd-Schlüsseln sowie bei Berücksichtigung von Redundanz und Konsistenz zu den Normalformen 1-3.

Die Datenrepräsentation in Form von Relationen verschafft die Möglichkeit der Bearbeitung konkreter Fragestellungen mit diesen Relationen.

Bei der Bearbeitung und Auswertung von Daten mit Hilfe einfacher Programme ist an die Verwendung der Datenbanksprache SQL (Structured Query Language) gedacht.

Geschichtliche Entwicklung der Datenverarbeitung (Inf 10.2)

Aufbau und Funktionsweise einer EDV-Anlage (Inf 10.3)

Bei der Betrachtung der Konzeption von Rechenanlagen spielt neben dem Aufbau und der Funktionsweise von Computern und Computernetzen auch die Konzeption und Planung von Netzwerken, Rechte- und Verzeichnisstrukturen, Sicherheitsstrategien eine wichtige Rolle.

Diese Konzeption in Rechnernetzen zielt auf Dienste im Internet, Informationssuche, Versand und Herunterladen von Dokumenten sowie die Produktion und Veröffentlichung eigener Dokumente.

Rechtliche Aspekte der Datenverarbeitung (Inf 10.4)

Aktuelle Aspekte von Computeranwendungen (Inf 10.5)

3.4 Gymnasium

3.4.1 Grundsätzliches

Die folgenden Ausführungen beziehen sich auf die Fachlehrpläne Informatik im **Lehrplan für das Gymnasium in Bayern** nach der Bekanntmachung des Bayerischen Staatsministeriums für Unterricht, Kultus, Wissenschaft und Kunst vom 2. August 1990 Nr. II/19 – S 5410 – 8/77 064 bzw. für den Grundkurs KMBI I 1986, So.-Nr. 11, S. 309.

An bayerischen Gymnasien wird die Informatik zum aktuellen Zeitpunkt vorwiegend in Wahlunterrichtskursen behandelt. Der einzige rein auf Informatik bezogene Kurs mit einem Lehrplan ist der Grundkurs Informatik des Zusatzangebotes in der Kollegstufe mit Stand von 1986. Das Additum Informatik des Mathematiklehrplans der Jahrgangsstufen 9 und 10 ist ausgerichtet auf die Lösung mathematischer (numerischer) Probleme; bestimmte informatische Inhalte, insbesondere im Bereich Algorithmik, können bei geeigneter Vorgehensweise und Schwerpunktsetzung thematisiert werden. Im Schulversuch "Europäisches Gymnasium" wird im Zweig III die Informatik in den Jahrgangsstufen 6, 10 und 11 als Pflichtfach eingeführt; die dafür vorgesehenen Inhalte korrelieren sehr stark mit den in Kapitel 2 vorgestellten.

Die Lehrpläne sind unter <http://www.isb.bayern.de/bf/isbl/lps/gym/informatik.htm> einsehbar.

3.4.2 Wahlunterricht

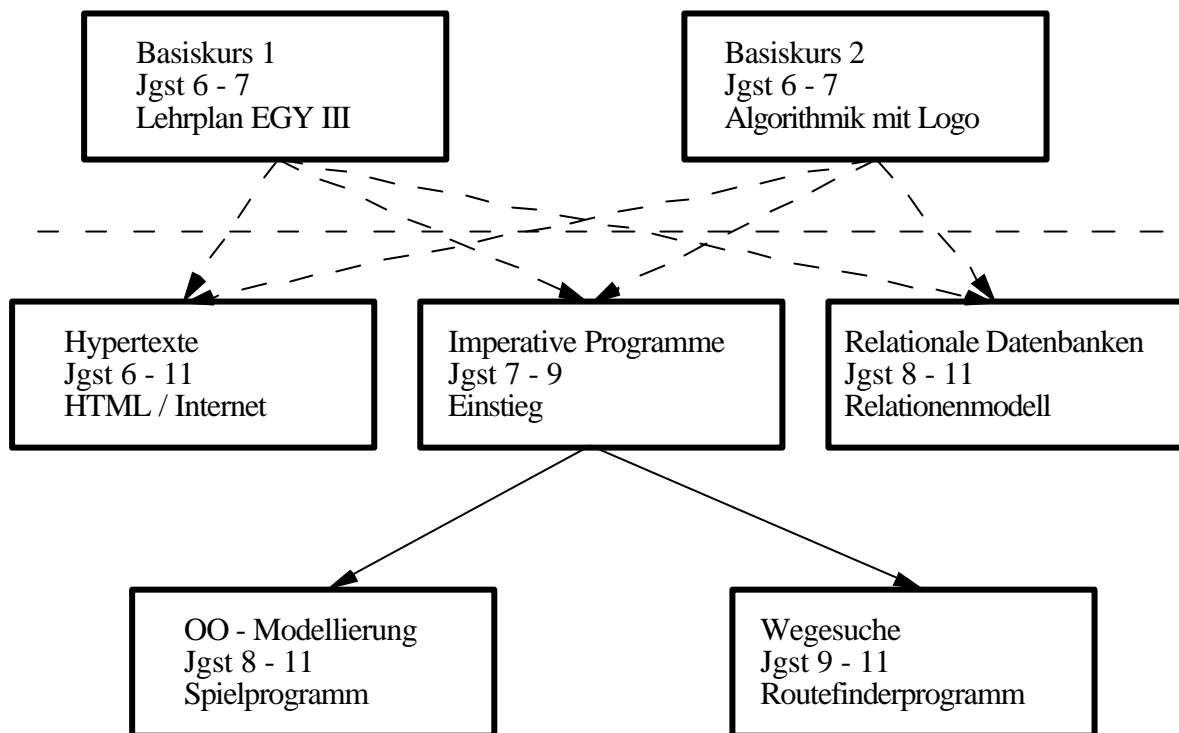
Das Informatikangebot im Wahlunterricht umfasst einen sehr weiten Inhaltsbereich von propädeutischen Einsteigerkursen in der Unterstufe bis zu sehr speziellen Themen für fortgeschrittene Schüler in der Mittel- und Oberstufe. Die in Kapitel 2 dargestellten Inhalte können im Wesentlichen angesprochen werden. Es muss aber, insbesondere bei den für die Oberstufe vorgesehenen Themen, der Alterssituation Rechnung getragen werden sowie bei umfangreichen Inhaltsgruppen die besondere Wahlunterrichtssituation berücksichtigt werden.

In der Regel wird der Wahlunterricht von den Schülern nicht durchgehend besucht. Deshalb sind mehrjährige Kurskonzeptionen nur sehr schwer zu realisieren, Angebote in "Baukastenform", die sich gegenseitig ergänzen aber nicht bedingen, sind geeigneter. Diese Angebote können zum Teil für alle Altersstufen gleichzeitig angeboten werden, die meisten Bausteine sind aber aufgrund von Abstraktionsniveau und Darstellung auf bestimmte Jahrgangsstufen zugeschnitten.

Das folgende Beispiel soll eine Anregung für ein derartiges Bausteinkonzept sein, das natürlich mit weiteren Angeboten im Bereich spezieller Vertiefung (z. B. Kurs Tabellenkalkulation, Kurs Assemblerprogrammierung) angereichert werden kann. Hier sind nur Kurse mit Inhalten des Kapitels 2 dargestellt. Alle Kurse sind auf eine Wochenstunde ausgelegt, wobei sich für die höheren Jahrgangsstufen eine Konzentration auf 2 Stunden alle 14 Tage oder ein halbes Jahr lang 2-stündig bewährt hat. Insbesondere bei den Bausteinen mit grö-

Bereits in den Projekten ist eine Ausweitung auf 2 Stunden pro Woche sehr leicht möglich und wünschenswert.

Das folgende Diagramm gibt eine Übersicht über die Kursangebote und die Abhängigkeiten der Kurse voneinander. Gestrichelte Linien signalisieren eine wünschenswerte, aber nicht notwendige Voraussetzung.



Basiskurs 1

Der Basiskurs führt über die objektorientierte Betrachtung von Graphik- bzw. Textdokumenten in Modellierungsstrukturen der Informatik ein und legt gleichzeitig Grundbegriffe über die Strukturierung der auf einem Rechner abgelegten Information. Mit geeigneten Werkzeugen, z. B. Roboterprogrammierung, wird anschließend in die Algorithmik eingeführt.

Bezug zu Kapitel 2: 2.1.2, 2.1.3.3

Basiskurs 2

Mit einer Programmiersprache wie Logo (Turtlegraphik) wird anhand graphischer Aufgaben eine Einführung in die Algorithmik gegeben. Auch die Verwendung höherer Datenstrukturen kann mit den in Logo als Sprachmittel vorhandenen Listen bereits in dieser Altersstufe an geeigneten Beispielen (z. B. Speichern der Daten eines Polygons) thematisiert werden.

Bezug zu Kapitel 2: 2.1.2, Teile aus 2.1.3.2, stark vereinfacht: Teile aus 2.2.2.3.

Einsteigerkurs in eine imperative Programmiersprache

Dieser Kurs, für den ein Besuch der Basiskurse hilfreich, aber nicht notwendig ist, führt

zunächst schrittweise in die grundlegenden Modellierungsmittel einer imperativen Sprache ein. Anschließend können (alternativ) Themen wie Sortieralgorithmen (Umgang mit Feldern, Laufzeiteffizienz), Rekursion / Backtracking oder dynamische Datenstrukturen (Listen, Bäume) behandelt werden.

Bezug zu Kapitel 2: 2.1.2, Teile aus 2.1.3.3, stark vereinfacht: 2.2.2.4, 2.2.2.6

Hypertexte

HTML / Internet; Textmodellierung (Setzen der Attribute) / Hypertexte / Ausnützen dieser Modelle für das Internet / Erweiterung von Pfadangaben für Dateibezüge auf URLs (Modellierung!)

Bezug zu Kapitel 2: 2.1.2, Teile aus 2.1.3.2, stark vereinfacht: Teile aus 2.2.2.2

Relationale Datenbanken

Einführung in das Relationenmodell, Modellierung einfacher und mittlerer Probleme und Realisierung in einer geeigneten relationalen Datenbank. Definition der Relationen am besten graphisch, für Abfragen ist SQL sinnvoll.

Bezug zu Kapitel 2: Teile aus 2.1.3.3

OO-Modellierung

An einem geeigneten Beispiel – Abenteuerspiele mit verschiedenen Akteuren und Gegenständen sind inhaltlich sehr gut geeignet und besitzen gleichzeitig einen hohen Motivationsgrad – werden zuerst grundlegende Begriffe wie Klasse / Objekt / Attribut / Methode geklärt und deren Realisierung in der verwendeten Programmiersprache untersucht (Spieler bewegt sich im vom Lehrer vorgegebenen Szenarium). Anhand der Gegenstände und Akteure werden Klassenhierarchien als Modellierungsmöglichkeit erkannt und das Konzept des Polymorphismus erarbeitet.

Bezug zu Kapitel 2: Teile aus 2.1.3.3, Teile aus 2.2.2.1, 2.2.2.6, 2.3, vereinfacht: 2.2.2.5

Wegesuche (Routefinderprogramm)

Bei der Realisierung eines Routefinderprogramms (Analyse eines kommerziellen Programms zur Erstellung des Pflichtenhefts für das Projekt) können baukastenartig die Themen Dynamische Datenstrukturen / Sortieralgorithmen / Backtracking / Graphenalgorithmen bearbeitet werden. Gleichzeitig werden die Schritte der Programmentwicklung thematisiert und Erfahrungen in Teamarbeit und Analyse komplexer Fragestellungen gesammelt.

Bezug zu Kapitel 2: Teile aus 2.1.3.3, 2.2.2.6, 2.3, vereinfacht: 2.2.2.3, Teile aus 2.2.2.5

3.4.3 Additum (Jahrgangsstufen 9 und 10)

Grundlagen

Dieser erste Teil des Additums legt die programmiertechnischen Grundlagen für die folgenden numerischen Anwendungen. Durch allgemeinere Betrachtung, insbesondere im

Punkt 2 des Lehrplans, lassen sich Prinzipien der Repräsentation von Information, vor allem aber die Bereiche Zustandsorientierte Modellierung / Algorithmen (s. Abschnitt 2.1.3.3) ansprechen. Einstiege in die Algorithmik nicht über eine Programmiersprache sondern über Robotersimulationen wie "Karel" (s. Abschnitte 4.4.1 / 4.4.2) heben diese prinzipiellen Ansätze noch deutlicher hervor.

Fortführung

Der zweite Teil des Additums behandelt einige Mechanismen zur Bearbeitung größerer Programm- (Prozeduren) und Datenstrukturen (Felder). Bei der Realisierung von Beispielen zu Punkt 1 (Prozeduren) lassen sich Grundideen der Softwareentwicklung, insbesondere die Top-Down-Entwicklung ansprechen (s. Abschnitt 2.2.2.6). Im Punkt 2 (Felder) kann beim Unterpunkt "Sortieren" das Problem der Laufzeiteffizienz (s. Abschnitt 2.2.2.4) angeschnitten werden. Beide Punkte können in den numerischen Anwendungsbeispielen des Punktes 3 wiederholt mit einbezogen werden.

3.4.4 Grundkurs

Der Lehrplan für den Grundkurs Informatik hat zwei wesentliche Rahmenbedingungen, die bei der Umsetzung moderner Informatikinhalt besonders bedacht werden müssen:

1. Im Gegensatz zur in dieser Broschüre vorgestellten Konzeption sieht er einen Einstieg in die Informatik in der Jahrgangsstufe 12 vor. Das bedeutet insbesondere, dass zur Realisierung der Inhalte gegebenenfalls ein Einstieg in eine geeignete Programmiersprache vorgesehen werden muss.
2. Aufgrund der didaktischen Sicht der Entstehungszeit sind algorithmische und numerische Aspekte sehr stark betont. Auch in der (unverbindlichen) Spalte "Hinweise zum Unterricht" und in der ebenfalls unverbindlichen Auswahl der zur Umsetzung des Lehrplans angegebenen Beispiele wird diese Sicht deutlich. Die Lernziele und Lerninhalte sind aber so allgemein formuliert, dass viele der in Kapitel 2 genannten Inhalte im Rahmen dieser Lernziele umgesetzt werden können.

Im Folgenden ist für die einzelnen Punkte des Lehrplans angegeben, welche der in Kapitel 2 angesprochenen Inhalte hier realisiert werden können. Der zeitliche Rahmen des Lehrplans lässt aber bei weitem nicht alle möglichen Inhalte zu.

I Zusammenfassung der Grundelemente

Die Lehrplanpunkte 1 mit 4 bilden zusammen mit dem zum zweiten Abschnitt gehörenden Lehrplanpunkt 5 Raum für eine Einführung in zustandsorientierte Modellierung und Algorithmik.

II.6 Kenntnis der grundlegenden Möglichkeiten, Daten problemangemessen zu strukturieren

II.7 Einsicht in grundlegende Verfahren, Problemlösungen systematisch zu gliedern; Überblick über diesbezügliche Möglichkeiten

II.8 Fähigkeit, strukturierte Problemlösungen zu realisieren

In diese Sequenz passt die objektorientierte Modellierung (s. Abschnitt 2.2.2.1) genau so wie die typischen Datenstrukturen Liste, Baum, Graph (s. Abschnitt 2.2.2.3). Auch die Automatentheorie (s. Abschnitt 2.2.2.1) ist ein Aspekt dieser Inhalte.

II.9 Einsicht in die Möglichkeit, Algorithmen durch geeignete Datenstrukturen zu vereinfachen

Effizienzvergleiche verschiedener Algorithmen (s. Abschnitt 2.2.2.4) sind Teil dieses Lehrplanpunktes.

II.10 Einblick in die Organisation großer Datenbestände, Kenntnis von Grundelementen der Datenverarbeitung

Dieser Punkt kann durch die Analyse statischer Datenmodellierung (Entity-Relationship-Diagramme) und Umsetzung in relationale Datenbanken erfüllt werden (s. Abschnitt 2.1.3.3).

III Rechnerarchitektur, Betriebssystem, Softwarearchitektur

Ein besonderer Aspekt dieses Lehrplanabschnitts sind verteilte Systeme (s. Abschnitt 2.2.2.2).

Der Grundkurs „Elektronische Datenverarbeitung“ ist ein Grundkurs des „Zusatzangebotes ohne Lehrplan“. Hier lassen sich unter der Voraussetzung, dass dem Schulleiter eine Lehrplanskizze vorgelegt wird, alle Inhalte eines zeitgemäßen Informatikunterrichts thematisieren, wie sie in Kapitel 2 dargestellt sind. Bei der Umsetzung von in Kapitel 2.1 angesprochenen Inhalten sollte man auf Altergemäßheit achten, d. h. insbesondere auf ein entsprechendes Abstraktionsniveau. So könnten beispielsweise bei der Behandlung der Datenmodellierung (s. Kapitel 2.1.3.3) die Normalformen oder die Mengenoperationen hier wesentlich formaler gefasst werden, als das bei einer Durchnahme in der Sekundarstufe I möglich ist.

3.4.5 Schulversuch "Europäisches Gymnasium" Zweig III

Jahrgangsstufe 6

Dieser erste Schritt in die Informatik soll zunächst systematisch eine gemeinsame Basis aus Wissen und Fähigkeiten herstellen. Nach und nach werden die Schüler anhand von Dokumentbeispielen aus verschiedenen Unterrichtsfächern in die Sprech- und Denkweisen der Informatik eingeführt. Da dieser Lehrplan noch wenig bekannt ist, sind die inhaltlichen Stichpunkte der Ebene 3 im Folgenden angegeben, der Lehrplanteil selbst kann für alle Jahrgangsstufen unter http://www.isb.bayern.de/gym/math_inf/ gefunden werden. Die Inhalte der Jahrgangsstufe 6 stimmen im Wesentlichen mit den in den Abschnitten 2.1.2 und 2.1.3.3 dargestellten Inhalten überein.

Inhaltsstichpunkte aus "Rahmenpläne der Jahrgangsstufen 6 - 11", Informatik Jahrgangsstufe 6:

- Darstellung von Information mit Hilfe von Dokumenten
- Verwaltung von Dokumenten
- Versand von Dokumenten
- Information in vernetzten Umgebungen
- Automatische Verarbeitung von Information

Jahrgangsstufe 10

Übergeordneter Inhalt der Jahrgangsstufe 10 ist "die systematische Erarbeitung des Grundprinzips der Modellierung, bei der Ausschnitte der Realität zielgerichtet vereinfacht und strukturiert dargestellt werden". Dabei werden verschiedene in 2.1.3.3 angeführte Modellierungsarten angesprochen und verglichen.

Inhaltsstichpunkte aus "Rahmenpläne der Jahrgangsstufen 6 - 11", Informatik Jahrgangsstufe 10:

- Datenmodellierung
- Modellierung von Abläufen mit Algorithmen
- Modellierung als Grundprinzip

Jahrgangsstufe 11

Mit der Behandlung weiterer Modellierungstechniken und Anwendung auf komplexere Probleme wird die in Jahrgangsstufe 10 begonnene Systematisierung der Modellierungstechniken fortgesetzt. Dabei werden weitere Inhalte aus Abschnitt 2.1.3.3 und aus Abschnitt 2.2.2.1 angesprochen. Bei der Durchführung können auch gut die Gedanken aus Abschnitt 2.3 realisiert werden.

Inhaltsstichpunkte aus "Rahmenpläne der Jahrgangsstufen 6 - 11", Informatik Jahrgangsstufe 11:

- Zustandsorientierte Modellierung
- Objektorientierte Modellierung
- Wertende Gegenüberstellung der behandelten Modellierungstechniken

4 Werkzeuge für den Informatikunterricht

4.1 Vorbemerkungen zu den Beschreibungskriterien

Ein Überblick über die im Informatikunterricht einsetzbaren Werkzeuge kann niemals vollständig sein. Deshalb sind hier nicht nur einzelne Werkzeuge beurteilt. Mit der Nennung allgemeiner Kriterien für die verschiedenen Themenbereiche des Informatikunterrichts soll auch eine Grundlage zur Beurteilung weiterer Werkzeuge gegeben werden. Gleichzeitig können dadurch die gegebenen Beurteilungen auf dem Hintergrund eigener Erfahrung, Situation und Bedürfnisse gewichtet werden. Insbesondere bei den Standardwerkzeugen ist eine Absprache mit anderen Fächern sinnvoll, da die Wahl schulweit einheitlicher Werkzeuge große Vorteile für den Unterricht hat.

Die Wertungen in den Beschreibungen richten sich an den Erfordernissen des Informatikunterrichts aus. Sie sind insbesondere keine Aussagen über die Verwendbarkeit eines Werkzeugs für ITG oder ähnliche Inhalte. Werkzeuge, die nur im Bereich der ITG (oder im Fachunterricht anderer Fächer) verwendet werden können, sind hier nicht aufgeführt.

Werkzeuge sind dazu nötig, Inhalte des Informatikunterrichts zu realisieren. Daher sind die Werkzeuge am angemessensten, die genau so viel können, wie für die Umsetzung der Inhalte nötig ist. Jedes mehr an Möglichkeiten ist für die Schüler eher verwirrend oder (insbesondere bei Entwicklungsumgebungen für Programmiersprachen) ablenkend.

Da ein großer Teil der im Informatikunterricht angesprochenen Modellbildungen mit Programmierwerkzeugen realisiert wird und gerade bei den Entwicklungsumgebungen die größte "Artenvielfalt" herrscht, ist dieser Punkt in der folgenden Darstellung besonders ausführlich behandelt.

4.2 Standardwerkzeuge

Bei den Standardwerkzeugen ist es besonders wichtig, darauf zu achten, dass nicht das Werkzeug an sich thematisiert wird, also keine Produktschulung betrieben wird, sondern dass es im Wesentlichen zur Umsetzung übergeordneter Themen benutzt wird. Eine klare, einfache Bedienung, insbesondere bei den Standardoperationen, ist daher wesentlich wichtiger als eine Vielzahl spezieller Möglichkeiten.

4.2.1 Relationale Datenbanken

Bei der Umsetzung statischer Datenmodellierung (Entity-Relationship-Modell) mittels relationaler Datenbanken ist die Verwendung eines kommerziellen Datenbanksystems fast immer der günstigste Weg. Nur wenn die Implementierung einer relationalen Datenbank

thematisiert werden soll (z. B. in einem anspruchsvolleren Wahlunterrichtskurs oder einem Pluskurs) wird man auf Programmierwerkzeuge zurückgreifen.

Bei der Auswahl des Werkzeugs sollte gute graphische Bedienbarkeit im Vordergrund stehen. Insbesondere die Beschreibung der Relationen direkt mit SQL-Anweisungen stellt eine unnötige Komplizierung für die Schüler dar. Andererseits sollte SQL für die Abfragen zur Verfügung stehen, da hier die Modellierungsstruktur am Besten zum Ausdruck kommt.

4.2.1.1 MS-Access

MS-Access enthält alle im Unterricht benötigten Funktionen. Die graphische Oberfläche sorgt zusammen mit einer klaren Gliederung der Funktionsgruppen für einfache Bedienbarkeit, auch für Anfänger. Einfache Datenabfragen können zum Einstieg in QBE ("query by example") formuliert werden, für komplexere Abfragen steht SQL zur Verfügung.

URL: <http://www.microsoft.com/office/access/default.htm>

Unter dem Titel "Datenbank" ist von der Zentralstelle für Computer im Unterricht eine Ausarbeitung zum Thema Datenbanken erhältlich, deren Beispiele in MS-Access realisiert sind.

4.2.1.2 DBase

DBase ist ein schon relativ lang bestehendes Datenbanksystem, das mittlerweile aber deutlich weiter entwickelt wurde. Sein Leistungsumfang ist ähnlich dem von MS-Access.

Information zur neuesten Version ist verfügbar unter <http://www.dbase2000.de>.

4.2.1.3 Paradox

Paradox ist eine zum Lieferumfang von Borlands Entwicklungsumgebung "Delphi" gehörende relationale Datenbank. Da bei der auf den professionellen Einsatz zielenden Konzeption vor allem an z. B. in Delphi ausprogrammierte Clientprogramme gedacht wurde, hat das mitgelieferte Administrationstool nur einen eingeschränkten Leistungsumfang (z. B. eingeschränkte Abfrage, keine Formulierung von Integritätsbedingungen).

Zur Realisierung der Beispiele bei einer allgemeinen Behandlung von statischer Datenmodellierung / Datenbanken ist Paradox daher nur eingeschränkt einsetzbar. Dagegen können spezielle Implementierungsdetails (z. B. Realisierung von Integritätsbedingungen, Verwendung eines Cursors auf einer Relation) durch die einfache Programmieranbindung über das zu Delphi gehörende Framework sehr gut thematisiert werden.

4.2.1.4 Oracle

Oracle ist eine sehr mächtige, professionelle Datenbank, die für den Einsatz umfangreiche Kenntnisse des Administrators erfordert. Zusätzlich müssen geeignete Clientprogramme vorhanden sein. Ein Einsatz im Unterricht kommt daher nur in speziellen Fällen (z. B. kostenloser Zugang zu einem Oracleserver) in Frage, wenn die Entwicklung von Datenbanken vertieft behandelt werden soll.

4.2.2 Hypertextsysteme

Beschäftigung mit Hypertextsystemen erlaubt die Betrachtung recht vielfältiger Arten von Informationsrepräsentation. Im Informatikunterricht ist nicht die Erstellung möglichst professioneller Dokumente das Thema. Herausgestellt werden müssen die prinzipiellen Unterschiede zwischen einem linearen (sequentiell betrachteten) Text, der mit Bildern angereichert sein kann (klassische Form des Buchs) und einem Hypertext, bei dem zur linearen Betrachtung noch die verschiedensten Zugriffsgraphen eben über die eingebauten Verweise ("Links") realisierbar sind (Querverweise).

4.2.2.1 HTML

Bekannteste Realisierungsform für Hypertextdokumente ist die Beschreibungssprache HTML (weil auch in WWW-Dokumenten verwendet), die die Erstellung von Hypertextdokumenten auf rein textueller Basis ermöglicht. Kommerzielle Wysiwyg-HTML-Editoren sind für den Einsatz im Informatikunterricht nicht zu empfehlen. Erstens wird der Vorgang der Textstrukturierung durch sie eher verschleiert. Zweitens erzeugen diese Editoren in der Regel unnötig komplizierte Formatierungen (z. B. mehrfach geschachtelte Tabellen). Dadurch ist eine spätere manuelle Nachbearbeitung oder eine Analyse des HTML-Textes deutlich erschwert.

HTML-Dokumente können mit jedem einfachen Texteditor (z. B. "WordPad" o. ä.) direkt erstellt werden. Zur Betrachtung und gleichzeitigen Überprüfung der Verweise und Einbindungen wird der vorhandene Webbrowser (z. B. "Netscape Communicator", "Internet Explorer") verwendet.

4.2.2.2 PDF

Das zur Anzeige insbesondere von Bedienungsanleitungen, Dokumentationen etc. entwickelte Dateiformat PDF erlaubt ebenfalls die Definition von Verweisen. Für den Einsatz im Informatikunterricht ist es bedingt geeignet. Es kann nur mit (teuren) kommerziellen Editoren erzeugt werden und bietet keinen Einblick in die erzeugten Textstrukturen.

4.2.3 Tabellenkalkulation

Auch wenn hier Grundlagen der Algorithmik gelegt werden können, findet Tabellenkalkulation für den Informatikunterricht nur wenig Akzeptanz.

Alle wesentlichen prinzipiellen Eigenschaften, insbesondere absolute und relative Bezüge auf Zellen, sind in allen gängigen professionellen Werkzeugen enthalten. Unterschiede treten erst in Details auf, die für den allgemeinbildenden Unterricht unwesentlich sind.

Geeignet sind daher für diesen Themenbereich ohne Unterschied alle gängigen Tabellenkalkulationsprogramme (MS-Excel, StarCalc, etc.), ebenso das lange bekannte Beispielprogramm Calc.

4.2.4 Sonstige Standardwerkzeuge

Weitere Standardwerkzeuge (z. B. Textverarbeitung, Mal- und Zeichenprogramme) können in der propädeutischen Informatik, z. B. Jahrgangsstufe 6 des EGY III, zum Einsatz kommen. Weitere Themen wie Einblick in die objektorientierte Modellierung lassen sich mit diesen Werkzeugen gut realisieren.

4.2.4.1 Textverarbeitung

Als Mindestvoraussetzung sollten Attribute der Objekte der Klasse Buchstabe (z. B. Zeichengröße, Schriftart, Schriftschnitt) und der Objekte der Klasse Absatz (z. B. Ausrichtung, Einzug) frei veränderbar sein. Programmiereditoren kommen daher in der Regel nicht in Frage. Echte Layoutprogramme benötigen im Allgemeinen zu viel Anfangsaufwand, als dass sie für den Einstieg verwandt werden könnten.

Geeignet sind für diesen Themenbereich ohne Unterschied die gängigen Textverarbeitungsprogramme (MS-Word, StarWriter, WordPerfect, etc.).

4.2.4.2 Zeichen- und Malprogramme

Objekte unterschiedlicher Art lassen sich bei beiden Arten von Programmen ansprechen. Malprogramme (Pixelgraphik, z. B. Paint) haben den Nachteil, dass die Attribute eines einmal erzeugten Objekts nachträglich nicht mehr geändert werden können. Zeichenprogramme (Vektorgraphik, z. B. Corel Draw) erlauben eine beliebige Veränderung der Attribute eines Objekts und zusätzliche Operationen (z. B. Verschiebung, Skalierung). Daher sind Zeichenprogramme für diesen Themenbereich angemessener, bei geeigneter Ausrichtung des Unterrichts sind Malprogramme durchaus anwendbar.

4.3 Programmierwerkzeuge

Bei der Auswahl von Programmierwerkzeugen muss man zwei in der Werbung häufig vermischte Kriterienbereiche sauber trennen: Programmiersprache und Entwicklungsumgebung. Eigentlich stellt die bei der jeweiligen Entwicklungsumgebung mitgelieferte Library ein drittes Entscheidungsmerkmal dar; für den Informatikunterricht ist dieses Kriterium aber deutlich weniger wichtig als für die Erstellung kommerzieller Programme.

Bei der Auswahl der Programmiersprache sind folgende Kriterien maßgeblich:

- Es sollen möglichst viele Modellierungsarten mit einer einzigen Sprache realisierbar sein, ideal ist die Verwendung eines einzigen Werkzeugs.
- Der Sprachaufbau soll möglichst klar und intuitiv sein. Gut lesbare Schlüsselwörter sind Abkürzungen durch Sonderzeichen vorzuziehen, Syntaxelemente sollen die mit ihnen verbundene Semantik deutlich machen.
- Die Sprache soll ein klares, ausgereiftes Typenkonzept besitzen.
- Fehlermeldungen sollen präzise sein. Insbesondere soll der Fehler am Ort des Auftretens

markiert und durch einen klaren Fehlertext beschrieben werden können.

Aufgrund der oben genannten Kriterien ist die Familie der von Professor Niklaus Wirth definierten Sprachen (Pascal, Modula-2, Oberon-2) deutlich vorzuziehen. Detailliertere Kriterien sind bei der jeweiligen Sprachbeschreibung angegeben.

Triviales Kriterium für die Auswahl der Entwicklungsumgebungen ist möglichst einfache Bedienbarkeit. Im Zeitalter graphischer Bedienoberflächen gehören dazu neben einem in den Entwicklungsablauf eingebundenen Editor (insbesondere sollte bei Fehlermeldungen des Compilers der Editor direkt auf die angezeigte Fehlerstelle positionieren) eine Menüsteuerung des gesamten Entwicklungsvorgangs und ein automatisches Erkennen von Quellabhängigkeiten bei einem mehrteiligen Programm ("make"). Hilfreich für die Realisierung größerer Programme ist eine Projektunterstützung durch die Entwicklungsumgebung.

Ein häufig übersehener Aspekt ist die Frage der Verfügbarkeit. Freeware-Entwicklungsumgebungen mit ausreichendem Leistungsumfang sind käuflich zu erwerbenden deutlich vorzuziehen, da sie den Schülern auch zu Hause uneingeschränkt zur Verfügung stehen. Dabei ist weniger an die Möglichkeit der Erstellung von Hausaufgaben im Pflichtunterricht gedacht, vielmehr erhalten interessierte Schüler dadurch die Möglichkeit, das Erlernete zu Hause weiterzuführen und zu vertiefen.

Im PC-Bereich werden Entwicklungsumgebungen oft danach beurteilt, ob und wie einfach professionell aussehende Windows-Applikationen erstellt werden können. Dieser Aspekt ist für einen allgemeinbildenden Unterricht völlig irrelevant, auch im Wahlunterricht sollte er eine deutlich untergeordnete Rolle spielen. Trotz der Faszination des schnellen Erstellens "gut" aussehender Programme darf nicht übersehen werden, dass damit keinerlei Lernzuwachs verbunden ist. Insbesondere schwächere Schüler wenden sich in solchen Entwicklungsumgebungen gern von der eigentlichen Problematik ab und vertun ihre Zeit mit der Anordnung, Form- und Farbgebung etc. der Oberflächenelemente.

4.3.1 Imperative und objektorientierte Programmiersprachen (PC)

Objektorientierte Sprachen sind heutzutage in der Regel Erweiterungen imperativer Sprachen bzw. sind aus imperativen Ansätzen abgeleitet. Deshalb ist eine gemeinsame Betrachtung angebracht; für die Realisierung objektorientierter Lösungen sollte nach Möglichkeit keine neue Sprache verwendet werden (müssen).

Im Bereich IBM-kompatibler PCs sind meistens Entwicklungsumgebungen auf genau eine Sprache ausgelegt. Deshalb ist dieses Kapitel nach den Sprachen gegliedert, dort werden dann vorhandene Entwicklungsumgebungen beschrieben.

4.3.1.1 Modula-2

Modula-2 ist die ausgereifteste der von N. Wirth entwickelten Sprachen. Als Weiterentwicklung von Pascal sind hier die erkannten Schwachstellen (insbesondere die fehlende Selbstklammerung der Kontrollstrukturen) behoben. Wesentlichste Neuerung ist die Einführung eines durch die Sprache abgesicherten Konzepts zur Zerlegung großer Programme

in einzelne Funktionsblöcke (Module), das auch das Verbergen von Implementierungsdetails (Information hiding) und damit die saubere Modellierung abstrakter Datentypen (ADTs) ermöglicht.

Im Zuge der Standardisierung (ISO-Standards IS 10514-1,2,3) erhielt die Sprache als Ausbau der ADTs objektorientierte Erweiterungen mit einem klaren Referenzmodell, d. h. im Gegensatz zu C++, Oberon-2 oder Borlands Pascal-Erweiterungen (in Delphi) sind keine Zeiger nötig.

Als einzige der für den Einsatz in der Schule in Frage kommenden Sprachen erlaubt Modula-2 die Modellierung paralleler Prozesse mit Sprachmitteln ("COROUTINES").

Stony Brook Modula-2

Eine sehr gute Entwicklungsumgebung für Modula-2 wird von der kanadischen Firma Stony Brook vertrieben. Implementiert ist der vollständige Standard mit einigen spezifischen Erweiterungen.

Stony Brook Software
187 East Wilbur Road Suite 4
Thousand Oaks, CA 91360
e-mail: stonybrook@compuserve.com

GPM-Modula-2

Die australische Firma GardensPoint entwickelt Compiler für verschiedene Sprachen und Zielplattformen, insbesondere im UNIX-Umfeld. Eine alte DOS-Version von GardensPoint Modula-2, die nur den Basisstandard (ohne objektorientierte Erweiterungen) implementiert, ist als Freeware verfügbar. Wo nur ältere Rechner zur Verfügung stehen (80386 abwärts), ist dieser Compiler immer noch interessant.

URL: <http://sky.fit.qut.edu.au/CompSci/PLAS/GPM>

4.3.1.2 Oberon-2

Die jüngste Sprache von N. Wirth (Oberon) wurde ursprünglich entwickelt, um sowohl die Basiskonstrukte für objektorientiertes Programmieren zu realisieren als auch einen möglichst kleinen und schnellen Compiler zu erhalten. Weiterentwickelt von Professor Mössenböck (früher Assistent von N. Wirth) zu Oberon-2 entstand eine Sprache, die zwar gegenüber objektorientiertem Modula-2 einige Einschränkungen, aber auch deutliche Vorzüge gegenüber Pascal (eingeschlossen Borlands objektorientierter Pascal-Erweiterungen) besitzt.

Pow

Pow ist eine unter der Leitung von Professor Mössenböck an der Universität Linz entwickelte, frei verfügbare Entwicklungsumgebung, für die insbesondere ein guter Oberon-2-Compiler einschließlich umfangreicher Library existiert.

URL: <http://www.fim.uni-linz.ac.at/pow/pow.htm>

Oberon/F

Oberon/F ist eine im nicht-kommerziellen Bereich frei verfügbare Entwicklungsumgebung für Oberon-2, sowohl auf IBM-kompatiblen PC wie auf Apple Macintosh verfügbar. Das sehr schöne Framework (Interaktives Tutorial!) setzt allerdings bereits für den Einsteiger gute Kenntnisse objektorientierter Modellierung voraus, so dass diese Umgebung wohl nur für spezielle Wahlunterrichtskurse in Frage kommt.

URL: <http://www.oberon.ch/>

4.3.1.3 Pascal

Pascal war die Antwort von Professor Niklaus Wirth auf die Ende der 60er, Anfang der 70er Jahre immer mehr ausufernden Sprachkonzepte. Ursprünglich als reine Ausbildungssprache konzipiert wurde Pascal wegen seiner klaren Struktur und Sicherheit sehr bald auch kommerziell verwendet. Die gerade bei der Strukturierung großer Programme auftretenden Schwächen wurden in der Fortentwicklung Modula-2 beseitigt. Heutzutage kann Pascal daher als weitgehend durch Modula-2 bzw. Oberon-2 abgelöst angesehen werden. Die Pascal-Standardisierung wurde vor ca. 10 Jahren eingestellt. Einziger namhafter Compilerhersteller für Pascal ist Borland (Turbo Pascal, Delphi).

Turbo Pascal

Turbo Pascal ist eine klare, einfache Entwicklungsumgebung, ab Version 6 mit objektorientierten Erweiterungen. Wo vorhanden ist es immer noch sinnvoll einsetzbar. Das käuflich auf dem Markt nicht mehr erwerbbares Produkt ist bei Bedarf für den unterrichtlichen Einsatz bei der Zentralstelle für Computer im Unterricht gegen eine Schutzgebühr erhältlich.

Delphi

Die aktuelle Entwicklungsumgebung der Firma Borland erfreut sich wegen ihres umfangreichen Frameworks (einfache Handhabung der Windows-Oberfläche) großer Beliebtheit an Schulen. Mehrere Untersuchungen haben aber klar ergeben, dass gerade durch dieses Framework der Einstieg relativ aufwendig ist, vor allem aber die Schüler durch die ausgeprägte graphische Oberflächengestaltung sehr stark von den eigentlichen Fragestellungen abgelenkt werden. Für die Vermittlung ernsthafter Grundkenntnisse in Informatik sind – den Preisvorteil noch gar nicht einbezogen – die didaktische Intentionen stärker unterstützende Entwicklungsumgebungen vorzuziehen.

4.3.1.4 Java

Java ist eine sehr konsequent objektorientierte Sprache, die ein sauberes Referenzkonzept (keine Zeiger) für die Objekte und eine umfangreiche, festgelegte Klassenbibliothek besitzt. Java wird in der Regel für eine virtuelle Maschine (Bytecode-Engine) übersetzt. Daher sind Java-Programme (und speziell die innerhalb eines Webbrowsers ausführbaren Applets) auf jeder Maschine ausführbar, auf der die Java-Maschine installiert ist. Vorteile sind die konsequente Objektorientierung, die zweckgerichtete Klassenbibliothek und die Laufzeitüberprüfungen. Nachteile sind die von C übernommene Syntax und die Notwendig-

keit, bereits für einfache Programme Objekte der Bibliothek nutzen zu müssen.

JDK von Sun

Eine einfache, aber voll funktionale Entwicklungsumgebung (einschließlich Projektverwaltung) ist das frei verfügbare "Java Development Kit" von SUN.

URL: <http://www.sun.com/>

Visual J++

Die professionelle Java-Entwicklungsumgebung Visual J++ wird von Microsoft angeboten.

URL: <http://msdn.microsoft.com/visualj>

4.3.1.5 C / C++

C ist eine in den 70er Jahren entwickelte Sprache, die ursprünglich als maschinenunabhängiger Ersatz für Assemblersprachen konzipiert war und daher nur ein sehr weiches Typenkonzept und keinerlei Laufzeitüberwachungen besitzt. Aufgrund der Syntaxdefinition von C können die Übersetzer nicht mit Recursive-Descent-Parsern arbeiten. Daraus resultiert die für den Unterricht sehr unangenehme Tatsache, daß Fehlermeldungen nur sehr grob sind und oft nicht an der eigentlichen Fehlerstelle stehen.

Die auf C aufbauende objektorientierte Sprache C++ hat alle diese Nachteile übernommen (siehe z. B. "Annotated C++ Manual" von Björn Stroustrup, den Entwickler von C++). Das Objektkonzept ist sehr verwaschen, da versucht wurde, das in C fehlende Modul- und Scopekonzept, wie es insbesondere die Wirth-Sprachen besitzen, durch die Klassen einzubauen.

Insgesamt kann wegen der Fehlerintoleranz und der unklaren Konzepte C / C++ für den Unterricht nur bedingt empfohlen werden.

Visual C++

Eine sehr bekannte Entwicklungsumgebung für C++ ist Visual C++ von Microsoft.

URL: <http://msdn.microsoft.com/visualc>

RHIDE C++

Eine vollständige C++-Entwicklungsumgebung im Stil von Turbo-Pascal ist auf der dem folgenden Buch beigelegten CD zu finden:

Breyman, Ulrich

C++ Eine Einführung

München 1997, Carl Hanser Verlag

ISBN: 3-446-19233-6 (<http://www.hanser.de>)

4.3.1.6 Basic

Basic ist eine sehr alte Sprache, die als Untermenge von FORTRAN für den Gebrauch auf Kleinrechnern konzipiert wurde. Wenn auch der Sprachumfang mittlerweile um echte Kon-

trollstrukturen erweitert wurde, sind doch Typen- und Unterprogrammkonzept derart rudimentär und veraltet, dass die Sprache heute für den Unterrichtseinsatz in Informatik nicht mehr geeignet ist.

Visual Basic

Der heute relevanteste Basic-Dialekt ist das von Microsoft entwickelte Visual Basic. Hier wird Basic als Makrosprache für eine Laufzeitumgebung verwendet, die alle wesentlichen Elemente der Windowsoberfläche und der Programmverwaltung von Windows graphisch bedienbar macht. Für die Elemente der Laufzeitumgebung wird objektorientierte Nomenklatur verwendet, aber es ist z. B. konzeptionell falsch, eine Ereignisprozedur zur Behandlung z. B. eines Mausklicks als Methode zu bezeichnen. Die Sprache selbst besitzt keinerlei Möglichkeiten zur Vereinbarung von Klassen und damit zur Realisierung objektorientierter Modellierung.

Auf Grund ihrer fehlerhaften Begriffsverwendung und ihrer mangelnden Grundkonzepte sollte Visual Basic im allgemeinbildenden Unterricht nicht verwendet werden. Für begrifflich gefestigte Schüler kann diese Sprache für Spezialthemen rund um die Windowsoberfläche in Frage kommen.

URL: <http://msdn.microsoft.com/vbasic>

4.3.2 Imperative und objektorientierte Programmiersprachen (Apple Macintosh)

Entwicklungsumgebungen für Applerechner sind in der Regel für beliebige Sprachen ausgelegt. Deshalb werden hier nur noch die Entwicklungsumgebungen selbst angesprochen; für den Vergleich der Sprachen siehe Abschnitt 4.3.1.

4.3.2.1 MPW

Die klassische Entwicklungsumgebung auf dem Mac ist MPW (Macintosh Programmers Workshop), ein frei verfügbares, sehr mächtiges, eigentlich kommandozeilenorientiertes Werkzeug, das aber über frei definierbare Menüs alle Vorgänge auch menügesteuert ausführen läßt. Für MPW sind alle bekannten Programmiersprachen verfügbar und in der Regel gut integriert.

MPW:

URL: <http://developer.apple.com/tools/mpw-tools/index.html>

Modula-2

p1 Gesellschaft für Informatik mbH

Hogenbergstraße 20

D-80686 München

Tel.: 089 / 54 61 31 - 0

Fax: 089 / 5802597

e-mail: m2@p1.spacenet.de

4.3.2.2 MetroWerks

Die Entwicklungsumgebung von MetroWerks ist ein sehr modernens Werkzeug und ebenfalls für beliebige Sprachen offen. Von MetroWerks selbst sind als Sprachen C / C++, Java und Pascal verfügbar. Dritthersteller können weitere Sprachen anbieten.

URL: <http://www.metrowerks.com/>

4.3.2.3 Oberon/F

Die in Abschnitt 4.3.1.2 vorgestellte Entwicklungsumgebung Oberon/F ist auch für Apple-Macintosh als abgeschlossenes System verfügbar (frei für nichtkommerziellen Gebrauch).

4.3.3 Imperative und objektorientierte Programmiersprachen (Linux)

In der freien Linuxwelt gibt es Compiler für alle bekannten Programmiersprachen. Da die Entwicklung hier in der klassischen Form Editor / Compiler / Linker über Kommandozeilenaufruf erfolgt, sei nur auf die Sprachbeschreibungen in Abschnitt 4.3.1 verwiesen.

4.3.4 Weitere Programmiersprachen

Neben der großen Gruppe imperativer Sprachen gibt es noch eine Reihe anders konzipierter Sprachen, die in speziellen Bereichen des Informatikunterrichts durchaus interessant sein können.

4.3.4.1 LOGO

LOGO ist eine interpretierende Sprache, die dafür entwickelt wurde, Kindern den ersten Zugang zum Computer zu ermöglichen. Dazu enthält sie eine spezielle, einfach zu bedienende Graphik (Turtle-Graphik), mit der sich sehr schöne, altersstufengerechte Aufgaben bearbeiten lassen. Großer Vorteil in dieser Altersstufe ist die direkte Visualisierung der Lösungsversuche und der Direkteingabemodus des Interpreters. Konzeptionell als funktionale Sprache entwickelt kann LOGO aber auch für wesentlich abstraktere Aufgabenstellungen verwendet werden. Da Listen als Grunddatentyp mit allen wesentlichen Operationen implementiert sind, lassen sich auch höhere Datenstrukturen bereits in sehr frühen Jahrgangsstufen thematisieren. Leider merkt man z. B. der zeilenorientierten Syntax das Alter der Sprache an. Ein genereller Einsatz kann, insbesondere auch wegen des fehlenden Typenkonzepts, allerdings nur bedingt empfohlen werden.

MS-Logo

Diese frei verfügbare Implementierung erweitert die eingebaute Graphik um die wichtigsten Elemente der Windows-Oberfläche (Fenster, Knöpfe, Maus mit zugeordneten Aktionsprozeduren).

URL: <http://www.softronix.com/logo.html>

Logo deutsch

Diese frei verfügbare Variante mit deutschen Befehlen ist besonders dann empfehlenswert, wenn die fehlenden Englischkenntnisse der Schüler den Umgang mit der Originalsprache noch nicht zulassen.

URL: http://www.ph-ludwigsburg.de/nutzer/klaudt_dieter/logo.htm

4.3.4.2 Funktionale Sprachen

Funktionale Sprachen sind eher von beweistechnischem Interesse. Trotzdem kann dieses Programmierparadigma in einem Grundkurs oder einem gehobenen Wahlunterrichtskurs sehr gut thematisiert werden. Ein sehr brauchbarer Vertreter funktionaler Sprachen ist Gofer, für PC (DOS-Box) und Apple Macintosh sind frei verfügbare Entwicklungsumgebungen vorhanden. Aber auch andere Sprachen wie Scheme, Lisp oder ML sind verwendbar.

URL: <http://haskell.systemsz.cs.yale.edu/haskell/gofer>

4.3.4.3 Beschreibende Sprachen

Prolog ist eine regelorientierte Sprache, bei der Daten anhand vorgegebener Regeln in einem Backtrackingverfahren bearbeitet werden. Früher häufig als sehr geeignet für den Informatikunterricht propagiert, ist diese Sprache heute wieder in Vergessenheit geraten.

4.4 Sonstige Programmiersysteme

Für eine Einführung in die Algorithmik in der Unterstufe des Gymnasiums oder für die Arbeit in der Hauptschule bieten sich wegen ihrer Anschaulichkeit und der direkten Sichtbarkeit der Resultate neben den Programmiersprachen auch einfach programmierbare Maschinen (Roboter) an. Dabei ist eine reale Maschine nicht unbedingt notwendig, eine Simulation am Bildschirm ist für die Schüler bereits faszinierend genug. Auch Autorensysteme sind für die Realisierung des Hauptschullehrplans in der Hauptschule geeignet.

4.4.1 Roboter Karel

Karel ist ein frei verfügbarer, simulierter Roboter, der über einen sehr einfachen Befehlsatz verfügt. Neben Bewegungsbefehlen (einschließlich Aufnehmen, Tragen und Ablegen von Ziegelsteinen) umfaßt der Befehlssatz die Kontrollstrukturen und Abfragefunktionen für verschiedene Sensoren (Platz belegt, etc.). Damit lässt sich eine anschauliche Einführung in die Algorithmik schon in sehr frühen Jahrgangsstufen durchführen. Neben der originalen DOS-Variante gibt es mittlerweile auch Implementierungen als Java-Applets.

DOS-Variante

FTP-Server der Zentralstelle für Computer im Unterricht, "hs-inf4.exe"

Java-Variante

URL: <http://www.mtsu.edu/~untch/karel/index.html>

4.4.2 Kara, der Marienkäfer

Kara ist eine an der ETH Zürich entstandene Weiterentwicklung des Roboters Karel; im Gegensatz zu Karel wird er aber nicht in einer einfachen Sprache, sondern schlicht graphisch programmiert. Dadurch entfallen besonders für Einsteiger die lästigen Syntaxfehler. Zu Kara gibt es unter der angegebenen Adresse auch ausgearbeitetes Material.

URL: <http://educeth.ethz.ch/informatik/interaktiv/kara/>

4.4.3 Lego-Roboter

Die Firma Lego hat einen relativ komplexen Roboter entwickelt, der sein Steuerprogramm über eine Infrarotschnittstelle von einem PC laden kann. Zum Roboter wird eine Entwicklungsumgebung geliefert, mit der sich die Steuerprogramme graphisch auf dem PC entwickeln lassen. Die zur Verfügung stehenden graphischen Steuerelemente sind dabei in vier Komplexitätsstufen aufgeteilt; in der Grundstufe sind nur einfache Steuerelemente verfügbar ("Anfängermodus"), in Stufe vier sind alle Elemente vorhanden.

URL: <http://www.legomindstorms.com>

4.4.4 Autorensysteme

In diesem Bereich gibt es mehrere Programme mit zum Teil verschiedenen Grundansätzen. Am besten bewährt hat sich das Programm "Mediator", insbesondere wegen seiner leichten Erlern- und Handhabbarkeit.

URL: <http://www.mwin.com/>

Die Broschüre "Autorensysteme, Multimedia von Schülern für Schüler, Kriterienkatalog – Begutachtungen, Programmübersicht" der Zentralstelle für Computer im Unterricht beschreibt insgesamt sechs verschiedene Autorensysteme.

5 Weitere Hilfen

5.1 Bücher (Eine enge Auswahl)

Didaktische Literatur

Verfasser	Titel	Verlag	Jahr
Baumann, Rüdiger	Didaktik der Informatik	Klett, Stuttgart	1996
Hubwieser, Peter	Didaktik der Informatik	Springer, Berlin	2000
Modrow, Eckart	Zur Didaktik des Informatik-Unterrichts, Bd.1 u. 2	Dümmler, Köln	1991
Baues, Jürgen et al.	Informatik erleben, Lehr- und Übungsbuch für die Sekundarstufe I, Bd. 1 u. 2, Lehrerhandreichung	Dümmler, Köln	1997
Baumann, Rüdiger	Informatik für die Sekundarstufe II, Band 1 und 2	Klett, Stuttgart	1992
Friedrich, Steffen (Hg.)	Informatik und Alltag, Anwendungsorientierte Einführung für die Sekundarstufe I, Basis- und Profilband, Lehrerhandreichungen	Dümmler, Köln	
Kreutzer, K.-H. et al.	Informatik für Einsteiger	Ehrenwirth, München	1992
ISB	Handreichungen für den Informatikunterricht am Gymnasium, Teil 1 und 2	ISB, München	1997/8
ISB	Bausteine zur Didaktik der Informatik	ISB, München	1997
AK. "Datenbank im Unterricht"	Datenbank	Zentralstelle für Computer im Unterricht, Augsburg	1997

Weiterführende Literatur

Verfasser	Titel	Verlag	Jahr
Balzert, Helmut	Lehrbuch der Grundlagen der Informatik, mit 2 CD-ROMs	Spektrum Verlag	1999
Balzert, Helmut	Softwareentwicklung, mit CD-ROM	Spektrum Verlag	1996
Bauer, Florian	Informatik, Teil 1 und 2	Springer, Berlin	1991/2
Broy, Manfred	Informatik – eine grundlegende Einführung in vier Bänden	Springer, Berlin	ab 1997
Broy, Manfred et al.	Übungen zur Einführung in die Informatik mit CD-ROM	Springer, Berlin	1997
Doberkat, Ernst-Erich et al.	Einführung in die objektorientierte Programmierung	Addison-Wesley, Bonn Oldenbourg, München	1996
Goos, Gerhard	Vorlesungen über Informatik I bis III	Springer, Berlin	ab 1996
Hartmann, Werner et al.	Informationsbeschaffung im Internet	Orell Füssli Verlag, Zürich	2000
Oesterreich, Bernd	Objektorientierte Softwareentwicklung	Oldenbourg, München	1997
Steiner, René	Theorie und Praxis relationaler Datenbanken	Vieweg, Braunschweig / Wiesbaden	1996
Tanenbaum, Andrew S.	Computernetzwerke	Prentice Hall	1998

Nachschlagewerke

Verfasser	Titel	Verlag	Jahr
Broy, Manfred Spaniol, Otto	VDI-Lexikon Informatik und Kommunikationstechnik	Springer, Berlin	1999
Claus, Volker et al.	Duden Informatik	Duden Verlag, Mannheim	1993
Rechenberg, Peter et al.	Informatik-Handbuch	Hanser Verlag, München	1997

5.2 Internet-Adressen (Stand: Februar 2000)

Diese Liste ist nicht auf Vollständigkeit, sondern auf Nützlichkeit hin konzipiert.

<http://ddi.uni-paderborn.de/didaktik/Veroeffentlichungen>

Diverse Publikationen zur Didaktik der Informatik

<http://www-schulen.informatik.tu-muenchen.de/didaktik/>

Peter Hubwieser: Vorlesungsskriptum zur Didaktik der Informatik I (WS 1998/99)

<http://www-schulen.informatik.tu-muenchen.de/didaktik/material.html>

Unterrichtsprojekte und Materialsammlung

http://www-cg-hci-d.informatik.uni-oldenburg.de/v_didaktik.html

H. Donker: Mediendidaktische Aspekte einer Kommunikationswerkzeuggestaltung

<http://educeth.ethz.ch/informatik/>

Erprobte Informatik-Unterrichtsmaterialien des Bildungsservers EducETH der ETH Zürich für alle Schulen der Sekundarstufe I und II.

<http://educeth.ethz.ch/informatik/interaktiv/>

Interaktive Programme (Applets) als Informatik-Unterrichtsmaterialien, die direkt aus dem Browser (Netscape, Explorer) gestartet werden können. Irgendwelche Installationen auf den lokalen Schulungs-PCs sind nicht notwendig

u. a.

<http://educeth.ethz.ch/informatik/interaktiv/javakarel/applet.html>

<http://educeth.ethz.ch/informatik/interaktiv/javakarel/>

<http://educeth.ethz.ch/informatik/grundkurs/>

Unterrichtsmaterialien aus einem einsemestrigen Grundkurs für die Jahrgangsstufe 11

<http://nobi.inf.ethz.ch/werner/infdidii/>

Informatik Didaktik: Beispiele für Materialien für den Informatikunterricht

<http://nobi.inf.ethz.ch/werner/infdidii/#Uebungen>

Beispiele von Unterrichtsübungen (s. o.)

<http://www.didaktik.cs.uni-potsdam.de/Lehre/Erweiterungsstudium/Skriptum>

A. Schwill: Skriptum zur Didaktik der Informatik – 1997

<http://www.didaktik.cs.uni-potsdam.de/HyFISCH/Informieren/VisuProgramm/CDViS2/Tutorial.htm>

M. Thomas: Visuelle Programmiersprachen im Informatikunterricht

<http://www.didaktik.cs.uni-potsdam.de/INFOS99/Wettbewerb/SpeichermodellFleischer>

R. Fleischer: Variablen und Speichermodell – 1999

[http://www.didaktik.cs.uni-potsdam.de/Lehre/Erweiterungsstudium/
Unterrichtshilfen/Beispiele](http://www.didaktik.cs.uni-potsdam.de/Lehre/Erweiterungsstudium/Unterrichtshilfen/Beispiele)

D. Weber-Wulff: Visualizations in Programming –1999

<http://www.learn-line.nrw.de/Faecher/inf.htm>

Geprüfte und kommentierte Verweise zu themenbezogenen Arbeitsbereichen

<http://www.fmi.uni-passau.de/schule/unterricht/links/informatik.html>

Informatik-Sammlung der Uni Passau (u. a. Betriebssysteme, Programmierung, Informatik in der Schule)

<http://region.hagen.de/OOP/>

Materialien als Grundbausteine für den Arbeitsbereich "Objektorientierung im Informatikunterricht der Sek.II" (NRW-Bildungsserver)

<http://did.mat.uni-bayreuth.de/baustein/>

Bausteine zur Didaktik der Informatik – ISB München

<http://www.fachdid.fu-berlin.de/Didaktik/Informatik/Kallmeyer/index.html>

Bildungskonzepte zur Informatik

<http://www.informatik.fernuni-hagen.de/import/schulinformatik/gespraech.html>

Gesprächskreis Informatik-Didaktik (GID)

5.3 Glossar

Verweise auf mehrere Online-Glossare finden sich unter

<http://members.aol.com/WBLexikon/lexstart.htm>

Ein sehr ausführliches Stichwortverzeichnis enthält auch "Informatik – eine grundlegende Einführung in vier Teilen (Teil IV), Manfred Broy unter <http://www.imn.htwk-leipzig.de/~feustel/br4/b4sw.html>

Automat	(Syn.: Maschine) Modellmechanismus zur Verarbeitung von Datenstrukturen: Elemente einer (abstrakten) Eingabemenge werden unter speichergestützter innerer Kontrolle abgearbeitet, transformiert oder erkannt und zur weiteren Verwertung ausgegeben. Der Turing Automat (Turing Maschine) ist das erste mathematische Modell einer idealisierten Rechenmaschine (1936 von A. M. Turing entwickelt; ausführlich unter http://129.69.58.100/Ausbildung/Referate98/russtern/turing/node4.html)
Betriebssysteme	Die Summe derjenigen Programme, die als residenter Bestandteil einer EDV-Anlage für den Betrieb der Anlage und für die Ausführung der Anwendersoftware erforderlich sind.
Client-Server-Architektur	Ein (zentrales) Server(-Programm) arbeitet mit (mehreren, dezentralen) Client(-Programmen) zusammen. Z. B. WWW-Server und Browser (Client); E-Mail-Postamt und lokales E-Mail-(Client-)Programm
Datenflussdiagramm	(Syn.: Datenflussplan) zeigt den Datenfluss durch ein informationsverarbeitendes System. Es besteht im Wesentlichen aus Sinnbildern mit zugehörigem Text für das Bearbeiten, Sinnbildern mit zugehörigem Text für Datenträger und der "Flusslinie".
Datenkapselung	Bei der Datenkapselung bilden die Vereinbarung der Daten und eine Sammlung von Prozeduren, die diese Daten verwalten eine "Datenkapsel". Von außen sind nur bestimmte Prozeduren (Zugriffsmethoden) zugänglich, nicht aber die Daten selbst. Datenkapselung wird bei der →objektorientierten Programmierung verwendet.
Datenmodell	Formale Methode, Information aus einem wohldefinierten, engbegrenzten Informationsbereich (Domäne) strukturiert darzustellen (→Datenstruktur)
Datenschutz	soll den Missbrauch von Daten verhindern (z. B. Zugangsbeschränkungen bei personenbezogenen Daten) (s. http://exai2.wu-wien.ac.at/~kaiser/mis/projekte/projekt10.html)

Datensicherheit	soll Verfälschung oder Verlust von Daten verhindern (s. http://exai2.wu-wien.ac.at/~kaiser/mis/projekte/projekt10.html)
Datenstrukturen	Strukturen von Informationsdarstellungen, die unter Verwendung von →Datenmodellen definiert werden. Eine klassische Unterscheidung von Datenstrukturen ist die zwischen <i>logischen</i> Datenstrukturen, <i>physischen</i> Datenstrukturen und <i>konzeptionellen</i> Datenstrukturen.
deterministischer endlicher Automat	endlicher →Automat, der aus Ist-Zustand und Eingabe eine eindeutige Ausgabe erzeugt ("eindeutige Transitionsfunktion")
Dienste	Leistungen/Arbeiten/Funktionalitäten von Computern, die für den Benutzer "unsichtbar" ablaufen, um die "sichtbaren" Ergebnisse zu ermöglichen. Z. B. "Transportdienst" für die Daten im Internet: Der Nutzer klickt auf einen Link und "sieht" darauf die Informationen; "unsichtbar" für den Nutzer wurde Kontakt zu verschiedensten anderen Rechnern aufgenommen, der Weg zum Rechner mit den gewünschten Daten ermittelt und Anfrage und Daten über das Netz hin- und hertransportiert.
Endlicher Automat	→Automat mit einer endlichen Kontrollstruktur
Entität	Element eines Informationsbereichs, über das Information vorhanden ist. Diese Information ist gegeben durch elementare Beziehungen zu anderen Entitäten.
Entity-Relationship-Diagramme	Darstellung der Beziehungen zwischen Entitäten
Firewall	Mechanismen zur Abwehr unberechtigter Zugriffe auf ein Computersystem: Eingehende Daten/Anfragen laufen auf der Firewall auf und werden erst nach entsprechender Prüfung an das eigentliche Rechnersystem weitergegeben. Ausführlich in http://www.oreilly.de/german/internet/miis/kapitel.html
Fremdschlüssel	Schlüssel, der von einer anderen →Entität übernommen wird (→relationale Datenbank)
gerichteter / ungerichteter Graph	→Graph
Graph	Ein Gebilde, das aus Knoten und Kanten besteht, welche entweder zwei Knoten oder einen Knoten mit sich selbst verbinden (Schlinge); ist den Kanten eine Richtung zugeordnet, heißt der Graph gerichtet; die Kanten können auch gewichtet (bewertet) sein.

Hypertextsystem	In einem Hypertextsystem können die angegebenen Quer-Verweise (Links) auf andere (Text-)Dokumente maschinell unterstützt sofort verfolgt werden: Auf "Mausklick" wird auf das verwiesene Dokument verzweigt und dieses angezeigt. Typisches Beispiel ist das World Wide Web im Internet
imperative Programme	Programme, deren typische Konstrukte Anweisungen sind
Implementierung	Der Arbeitsvorgang, durch den eine Komponente mittels gegebener Bausteine realisiert wird. Z. B. Implementierung einer (abstrakt entworfenen Programmiersprache [z. B. Modula-2]) in einen gegebenen Computer (wodurch ein Programmiersystem entsteht [z. B. Stony Brook Modula-2])
informationstechnische Grundbildung (ITG)	Eine der in den Lehrplänen verpflichtend festgelegten fächerübergreifenden Bildungs- und Erziehungsaufgaben; (s. http://www.stmukwk.bayern.de/schule/itg/index.html)
Internet	weltweites öffentliches Netz von verbundenen Computern
ISO-OSI Schichtenmodell	Referenzmodell für Rechnernetze mit 7 funktionalen Schichten (Anwendungsschicht, Anpassungsschicht, Verbindungsschicht, Transportschicht, Netzwerkschicht, Streckenschicht, Physikalische Schicht) (s. http://131.188.160.241/rechner/netzwerk/vortrag/11Nov1996/Vortrag_Netzwerk_S001.html)
ITG	→informationstechnische Grundbildung
Kellerspeicher	linearer Speicher nach dem LIFO-Prinzip (last in, first out) (im Gegensatz zur →Warteschlange) (s. http://www.darmstadt.gmd.de/schulen/LICHTENBERGSCHULE/projekte/lkinfr/lkinf15.htm)
Kontrollstrukturen	Mechanismen um das funktionale Zusammenwirken verschiedener Teile zu bestimmen.
Methoden	In der →objektorientierten Programmierung Bezeichnung für die Möglichkeiten, wie Objekte aufeinander einwirken können.
Modellbildung	Entwerfen eines (abstrakt symbolischen) Abbildes eines Teiles der Realität. Im Modell können dann durch Verfahren der Mathematik oder Informatik Problemlösungen gefunden werden, die nach Rücktransformation in die Realität auch die realen Probleme lösen.
Modellierungstechniken	Verfahrensweisen, um von einem realen Problem zu einer abstrakten (und maschinell verarbeitbaren) Darstellung zu kommen

nebenläufiger Prozess	(syn.: paralleler Prozess) Prozesse können nebenläufig sein, wenn sie nicht miteinander in Konflikt stehen und keine unmittelbare kausale Abhängigkeit besteht (wenn sie keine gemeinsamen Vorbedingungen und Nachbedingungen haben).
Netzwerkprotokolle	Damit verschiedene Geräte in einem Netzwerk miteinander arbeiten können, müssen sie ihre Aktivitäten aufeinander abstimmen; diese Abstimmungen werden in Protokollen festgelegt (→ISO-OSI-Schichtenmodell). Bekannteste Familie von Netzwerkprotokollen ist die →TCP/IP-Protokollfamilie, die im Internet verwendet wird.
nichtdeterministischer Automat	→Automat mit nicht eindeutiger Transitionsfunktion, d. h. durch Istzustand und Eingabe ergibt sich die Ausgabe nicht eindeutig
Normalformen 1 – 3	Genügen die Tabellen einer →relationalen Datenbank bestimmten Bedingungen, so stehen sie in Normalform. (s. http://www.fsai.fh-trier.de/%7Eholert/dv/dvglossar.html)
objektorientierte Modellierung/ Programmierung	→Modellierung, bei der Daten und Methoden in Objekten zusammengefasst werden (→Datenkapselung, →Polymorphie, →Vererbung) (s. http://www.bg.bib.de/baeumle/texte/cppbuchauszug.html#abs13) Folien zum Thema Objektorientierte Modellierung (Prof. Magenheim, Universität Paderborn): http://ddi.uni-paderborn.de/didaktik/Veroeffentlichungen/mnukiel/sld001
Parallele Prozesse	→nebenläufige Prozesse
Petriernetz	Darstellung für →nebenläufige Prozesse
Polymorphie	→objektorientierte Programmierung: Eine Funktion erhält eine einzige Bezeichnung, die in der gesamten Objekthierarchie Verwendung findet. Jedes Objekt in der Hierarchie →implementiert die Funktion in der jeweils erforderlichen Art und Weise.
Rechnerarchitektur	Struktur des Aufbaus der Hardware aus den einzelnen Hardware-Betriebsmitteln
Redundanz	bei Datenbanken: Mehrfachspeicherung von Daten

relationale Datenbank	<p>Nach dem von Codd u. a. 1970 vorgestellten Relationenmodell aufgebaute Datenbank (im Gegensatz zum hierarchischen Datenbankmodell): Gleichartige Datenmengen (Entitäten) werden in Tabellen zusammengefasst, deren Spalten die Eigenschaften (Attribute) und deren Zeilen die einzelnen Datensätze bilden. Jeder Datensatz erhält zur Unterscheidung einen eindeutigen Schlüssel zugeordnet (z. B. laufende Nummer); mehrere Tabellen können über ihre Schlüssel miteinander kombiniert werden (Relationen).</p> <p>[Foliensatz zu relationalen Datenbanken: http://www-dbs.ethz.ch/~infosys/folien_online/kap2/sld001.htm]</p>
Schlüsselattribute	→relationale Datenbank
Schutzrechte im Internet	<p>Aufgrund der weltweiten Ausdehnung des Internets ergeben sich derzeit noch ungelöste Rechtsfragen z. B. bezüglich Datenschutz, Jugendschutz, Urheberrechtsschutz ("Das Recht welchen Landes gilt?")</p> <p>(s. http://www.dfn.de/service/ra/aktuelles/SchuleUndInternet.html)</p>
Semaphore	Das Semaphor-Konzept dient zum Steuern →paralleler Prozesse; wurde von Hoare weiterentwickelt zum Monitor-Konzept
Struktogramm	Symbolische Darstellung der Struktur eines Programmes
TCP/IP	<p>TCP: Transport control protocol; IP: Internet protocol. Die Protokollfamilie im Internet</p> <p>(s. http://www.tu-chemnitz.de/urz/netz-kurs/script.html)</p>
Top-Down-Analyse	Problemanalyse vom Allgemeinen zum Besonderen (im Gegensatz zur bottom-up-Analyse)
Turing-Maschine	→Automat
Vektor- und Pixelgraphik	<p>Bei einer Vektorgraphik werden die graphischen Objekte (z. B. Linie, Fläche) mit allen Eigenschaften (z. B. Anfangs- und Endpunkt; Strichstärke, Farbe) einzeln gespeichert und dann aufgrund dieser Angaben auf der Zeichenfläche jedes Mal neu dargestellt. Vorteil: wenig Speicherplatz nötig, alle Objekte können einzeln bearbeitet werden. Nachteil: Bild muss jedes Mal neu berechnet werden.</p> <p>Bei einer Pixelgraphik werden nur die Farben aller Pixel der Bildfläche gespeichert. Es wird nicht gespeichert, "warum" ein Pixel eine bestimmte Farbe hat (z. B. weil es zu einer Kreislinie gehört). Vorteil: Einfache Speicherung und Darstellung. Nachteil: Nachträgliche Änderungen an "Objekten" nur schwer möglich; hoher Speicherplatzbedarf</p>

Vererbung	(→objektorientierte Programmierung) Eigenschaften eines Objektes werden auf davon abgeleitete Objekte übertragen.
von-Neumann-Architektur	Eine Rechnerarchitektur, die nach dem von-Neumann-Operationsprinzip der Verarbeitung unstrukturierter Daten arbeitet. Die Betriebsmittel bestehen aus einer Zentraleinheit, einem Speicher und einem oder mehreren Ein-/Ausgabeprozessoren.
Warteschlange	Zwischenspeicher nach dem FIFO-Prinzip (first in first out); (im Gegensatz zum →Kellerspeicher)