

## Rezensionen

**Balzert, Heide: Lehrbuch der Objektmodellierung – Analyse und Entwurf. Heidelberg: Spektrum Akademischer Verlag, 1999. ISBN 3-8274-0285-9. 575 S.; EUR 59,95 (mit 1 CD-ROM).**



Das Buch bietet eine leicht verständliche Einführung in das objektorientierte Modellieren. Dabei orientiert es sich an objektorientierten Konzepten und methodischen Vorge-

hensweisen und nicht an den Diagrammen der *Unified Modeling Language* (UML) oder Schlüsselwörtern einer objektorientierten Programmiersprache. UML wird aber durchgängig zur Veranschaulichung herangezogen. Programmtext-Fragmente in den implementierungsnahen Teilen sind in JAVA bzw. C++ abgefasst. Auf akademischem Niveau erklärt die Autorin kompakt und klar – von Fachbegriffen bis hin zu Konzepten – Schritt für Schritt den Entwurf objektorientierter Software.

Inhaltliche Schwerpunkte des Buches bilden: Konzepte und Notationen der objektorientierten Analyse (Basiskonzepte, statische und dynamische Konzepte), Analysemuster, Prüflisten zur Erstellung eines OOA-Modells, Konzepte und Notationen des objektorientierten Entwurfs, Entwurfsmuster, Datenbanken, verteilte objektorientierte Anwendungen, Erstellung eines Entwurfsmodells mittels Drei-Schichten-Architektur. Für Anfänger sind besonders die Lehreinheiten zu den Grundkonzepten (Objekt, Klasse, Attribut, Operation), den statischen Konzepten (Assoziation, Vererbung, Paket) und den dynamischen Konzepten (Anwendungsfall, Botschaft, Szenario, Zustandsautomat) geeignet, in denen sehr ausführlich und anschaulich in wesentliche Konzepte der Objektorientierung eingeführt wird.

Die Lehreinheiten zu Analyse- und Entwurfsmustern tragen deren erheblichen Stellenwert in der Softwaretechnik Rechnung. Muster stellen wiederverwendbare Teilmodelle für wiederkehrende Problemstellungen bei Analyse und Entwurf dar, Aufgabenstellungen, die auch für den schulischen Kontext in gewissem Umfang relevant sind. Die Prüflisten zur Erstellung eines OOA-Modells können in einer didaktisch entsprechend reduzierten Fassung auch Schülerinnen und Schülern bei der Gestaltung eigener objektorientierter Modelle helfen.

Durch die Gliederung in Lehreinheiten mit jeweils vorangestellten Lernzielen, nachgelagertem Kapitelglossar, Zusammenfassung und Lernerfolgskontrollen mit Lernziel- und -zeitangaben ist das Buch gut für das Selbststudium, insbesondere auch für Anfänger geeignet. Vorkenntnisse zur Objektorientierung sind nicht erforderlich. In den implementierungsnahen Abschnitten sind aber Programmierkenntnisse (nicht notwendigerweise zur objektorientierten Programmierung) für das Verständnis hilfreich. Ähnliches gilt für die Lehreinheiten, die sich mit Datenbanken und verteilten Anwendungen befassen. Hier erleichtern diesbezügliche Vorkenntnisse das Verständnis erheblich.

Das Buch ist, nicht zuletzt auch aufgrund seiner aufwändigen Gestaltung, sehr gut lesbar. Ein umfangreiches Gesamtglossar am Ende erleichtert das Nachschlagen von Fachbegriffen. Insgesamt handelt es sich bei dem Buch um eines der gelungensten Lehrbücher zur Thematik, das dem Rezensenten bekannt ist.

Auf der beigefügten CD-ROM befindet sich u. a. ein kleines, handliches UML-Werkzeug namens *Generating Objects* – GO, das leicht zu bedienen ist, die Erstellung von Objekt-, Klassen-, Sequenz- und Zustandsdiagrammen ermöglicht und die Generierung von JAVA-Programmrahmen unterstützt (bei einer Verzeichnisgröße von unter 1,5 MByte, was auch für weniger leistungsstarke Schulrechner kein Problem darstellen sollte). Erwähnt werden sollte allerdings, dass nicht alle verfügbaren Gestaltungsfunk-

tionen zu Sequenzdiagrammen (z. B. Objektkonstruktion- und -destruktion zur Laufzeit) auch aktivierbar sind.

Torsten Brinda

**Barnes, David J.; Kölling, M.: Objektorientierte Programmierung mit Java – Eine praxisnahe Einführung mit BlueJ. Reihe „Pearson Studium“. München u. a.: Addison-Wesley, 2003. ISBN 3-8273-7073-6. 461 S.; EUR 34,95 (mit 1 CD-ROM).**



Das Buch ist die deutsche Übersetzung des englischen Titels *Objects First with Java: A Practical Introduction Using BlueJ*. Bei BLUEJ handelt es sich

um eine Lernumgebung für Objektorientierung und JAVA. In ihr können an UML angelehnte Klassendiagramme erzeugt werden. Die einzelnen Klassen lassen sich mit Programmtexten hinterlegen; es können per Mausklick Objekte erzeugt und Methoden aufgerufen werden.

Das Motto der Autoren von BLUEJ lautet: „So wenig wie möglich!“ – und die Realisierung dieser Absicht scheint durchaus geglückt. Dennoch lassen sich kleinere und mittlere Projekte problemlos verwirklichen. Die Schülerinnen und Schüler werden jedoch nicht von einem Funktionsumfang wie bei professionellen Entwicklungsumgebungen verwirrt und überfordert. Daher stellt das Kennenlernen der Umgebung keine Hürde bei der Aneignung der Lerninhalte dar (s. a. den Beitrag von Leipholz-Schumacher, in diesem Heft, S. 32–39).

Das Buch gliedert sich in zwei Teile: *Grundlagen der Objektorientierung* und *Anwendungsstrukturen*. Im ersten Teil werden zunächst Objekte und Klassen eingeführt sowie die Interaktion zwischen Objekten betrachtet. Sehr zügig erfolgt eine Überleitung zu anspruchsvolleren

Themen, wie z. B. der Nutzung von Bibliotheksklassen, einschließlich des Lesens der JAVA-Klassendokumentation. Ferner werden Themen der Softwaretechnik diskutiert, wie das Vermeiden von Fehlern und der Klassenentwurf. Im zweiten Teil folgen Vererbung einschließlich Schnittstellen (interfaces), Fehlerbehandlung und OOA/OOE-Techniken.

Das Vorwort verspricht, im Buch objektorientierte Programmierung anfängergerecht darzustellen: Viele Konzepte sollen zunächst „flach“ eingeführt werden, um sie später zu vertiefen und zu Beginn würde so weit wie möglich von der Programmiersprache abstrahiert.

Letzteres fällt positiv auf. Die ersten Seiten beschäftigen sich nicht mit dem JAVA-Programmtext, sondern nur mit der grafischen Benutzeroberfläche von BLUEJ. In diesem Zusammenhang werden zunächst Klassen und Objekten erklärt; in den dazugehörigen Übungsaufgaben werden intuitiv Objekte (von vorgegebenen Klassen) erzeugt und Methoden aufgerufen. Beim Einstieg in den Programmtext werden zunächst vorhandene Methoden nur leicht abgeändert, sodass der Leser den Programmtext nur in kleinen Ausschnitten verstehen muss.

Auch das schrittweise Vertiefen ist deutlich zu erkennen: Das Thema *Objekte* wird mithilfe des Erzeugens neuer Objekte durch die Programme selbst weiter vertieft, woran sich Objektsammlungen (Listen, Mengen etc.) anschließen. Ähnliches gilt für andere Themen, wie z. B. den Entwurf. Zunächst wird der Klassenentwurf eingeführt, später vertieft und in den Entwurf ganzer Anwendungen eingebettet.

Eine anfängergerechte Darstellung ist dem Buch allerdings nur bedingt zu bescheinigen. Die Strukturierung des Stoffs fällt unter diesem Aspekt zunächst positiv auf. Die didaktischen Möglichkeiten von BLUEJ werden gut genutzt, um die Leser an die Objektorientierung und an JAVA heranzuführen. Allerdings wird eine Fülle von Themen – auf 450 Seiten komprimiert – dargeboten, wodurch das Buch für Anfänger sehr anspruchsvoll wird. Daher ist es als Lehrbuch für die

Hand der Schüler nicht zu empfehlen. Für die Lehrenden kann es jedoch eine große Hilfe bei der Nutzung von BLUEJ im Unterricht sein. Allein die Übungsaufgaben (es gibt deren mehrere hundert) werden eine willkommene Hilfe bei der Unterrichtsvorbereitung sein. Die Übungsaufgaben selbst sind einleuchtend strukturiert und bauen schön aufeinander auf. Als einziger kleiner Wermutstropfen muss hier die fehlende Kennzeichnung der Aufgaben mit Schwierigkeitsgraden oder Zeitangaben genannt werden.

Insgesamt hätte das Buch davon profitiert, wenn einige vertiefende Themen ausgelassen und andere dafür breiter behandelt worden wären. Beispielsweise wird es einem Leser wenig nützen, eine halbe Seite über Kooperation und Teamarbeit zu lesen, ebenso wenig wie eine ganze Seite über die Entwicklung von Prototypen. Auf sechs Seiten werden Entwurfsmuster allgemein eingeführt, und vier Muster werden exemplarisch dargestellt. Dieser Platz reicht jedoch in anderen Büchern zurecht nicht aus, um auch nur den Sinn von Entwurfsmustern darzulegen.

Andererseits fehlt dem Buch eine Hinführung des Lesers zu einer Programmierung außerhalb von BLUEJ. Die Autoren begnügen sich mit drei Seiten im Anhang. Hier wäre mindestens eine Erklärung typischer Möglichkeiten der Ein- und Ausgabe zu erwarten gewesen. Es fehlt selbst ein kurzer Hinweis auf einschlägige Pakete der Klassenbibliothek.

Zusammenfassend kann gesagt werden, dass das Buch aufgrund der äußerst knappen Darstellung vieler Themen als Schülerbuch zwar ungeeignet ist, dafür jedoch eine wertvolle Unterstützung für die Lehrenden sein kann. Die sehr durchdachte Gliederung des Stoffs und die vielen Übungsaufgaben sind eine große Hilfe bei der Unterrichtsvorbereitung. Durch die Fülle und Tiefe der Themen kann das Buch auch zur Begleitung für einen Leistungskurs empfohlen werden.

Matthias Krings

**Schubert, Sigrid; Schwill, Andreas: Didaktik der Informatik. Heidelberg; Berlin: Spektrum Akademischer Verlag, 2004. ISBN 3-8274-1382-6. 372 S.; EUR 29,95.**



Es ist hoch erfreulich und begrüßenswert, dass zwei Informatik-Didaktiker, die durch zahlreiche einschlägige Publikationen hervorgetreten sind und damit

den Informatikunterricht entscheidend mitprägten, ihre gesammelten Ansichten nun in einer Monografie (als Gemeinschaftswerk) vorgelegt haben.

Nachdem eine Übersicht über den Inhalt des Buches gegeben wird, wird in der vorliegenden Rezension von folgenden Fragen bzw. Gesichtspunkten ausgegangen:

- ▷ Wie wird Informatik als Schulfach legitimiert?
- ▷ Welches Verständnis von einer Fachdidaktik wird aufgezeigt?
- ▷ Wie sehen die Autoren die Beziehungen der Informatik zu anderen Schulfächern?
- ▷ Welche Unterrichtsmittel (Programmiersprachen, Softwarewerkzeuge, Lernumgebungen) werden empfohlen?

Ferner werden die entsprechenden Stellungnahmen der Autoren mit denen einer neueren informatik-didaktischen Publikation verglichen, und zwar mit derjenigen von Peter Hubwieser aus dem Jahr 2000 (die 2., überarbeitete Auflage von 2004 dieses Didaktik-Werkes wird demnächst gesondert rezensiert).

### *Zum Inhalt*

Das Buch ist in 11 Hauptkapitel eingeteilt: 1. Was ist Informatik? 2. Grundmodell für Ziele, Inhalte und Lehrmethoden, 3. Theoretische Fundierung der Schulinformatik, 4. Problemlösen im Informatikunterricht, 5. Didaktisches System, 6. Informatisches Modellieren und Konstruieren, 7. Objektorientierung, 8. Interaktion, 9. Informatiksysteme, 10. Anfangsunterricht, 11. Projekte. Darüber hi-

naus gibt es ein 8-seitiges Literaturverzeichnis, vier Anhänge (A. Kompetenzen und Lernziele, B. Programmbeispiele, C. Informatiklabor, D. Geschichte der Informatik) und ein 12-seitiges Register.

## *Wie wird Informatik als Schulfach legitimiert?*

Sowohl Schubert/Schwill wie auch Hubwieser gehen – um einseitigen Festlegungen und Missverständnissen bezüglich der Wörter „Computer“ und „Programm(ien)“ vorzubeugen – vom Begriff des Informatiksystems aus: „Als *Informatiksystem* bezeichnet man die spezifische Zusammenstellung von Hardware, Software und Netzverbindungen zur Lösung eines Anwendungsproblems“ (Schubert/Schwill, S. 5) und folgern daraus: „*Informatik* ist in diesem Sinne dann die Wissenschaft von Entwurf und Gestaltung von Informatiksystemen“ (Schubert/Schwill, S. 6). Nach Hubwieser werden Informatiksysteme durch die Merkmale (a) automatische Verarbeitung von Daten, (b) Vernetzung und (c) Interaktion mit menschlichen Benutzern gekennzeichnet (Hubwieser, S. 43). Nimmt man an, dass dieser letzte Punkt (Interaktion) bei Schubert/Schwill in der Wendung „Lösung eines Anwendungsproblems“ enthalten ist, stimmen beide Begriffsbestimmungen im Wesentlichen überein (vgl. Schubert/Schwill, S. 46 f.). Diese pragmatische, d. h. sich auf menschliches Handeln beziehende Bestimmung der Informatik schafft beste Voraussetzungen für die Konzeption eines handlungsorientierten Unterrichts (zu dem sich sowohl Schubert/Schwill als auch Hubwieser bekennen), der auch auf Fragen der persönlichen und gesellschaftlichen Verantwortung eingeht.

Zur Lösung des Legitimationsproblems legen Schubert/Schwill den Begriff der *informatischen Bildung* zugrunde (S. 55 f.), untermauern diese Lösung durch „fundamentale Ideen“ und konkretisieren sie durch drei Leitlinien (S. 46 ff.), nämlich:

- ▷ *Pläne* (Leitfrage: „Wie können durch planvolle Entwicklung, Gestaltung und Anwendung von Informatiksystemen Probleme

der Lebenswelt gelöst werden?“).

- ▷ *Sprachen* (Leitfrage: „Welches sind die Möglichkeiten und wo liegen die Grenzen formal-sprachlicher Wissensverarbeitung?“).
- ▷ *Systeme* (Leitfrage: „Wie sind Informatiksysteme aufgebaut, welches sind die Prinzipien des Zusammenwirkens ihrer Komponenten und wie ordnen sie sich in größere Systemzusammenhänge ein?“).

Hubwieser sieht den allgemein bildenden Wert des Informatikunterrichts in folgenden – von Bußmann und Heymann (1987) formulierten und auch in dieser Zeitschrift für die Informatik fruchtbar gemachten (Lehmann, 1992) – Bildungszielen: Vorbereitung auf zukünftige Lebenssituationen, Stiftung kultureller Kohärenz, Aufbau eines Weltbilds, Anleitung zum kritischen Vernunftgebrauch, Förderung von Phantasie und Kreativität, Weckung von Verantwortungsbereitschaft und Befähigung zum Tragen von Verantwortung (Hubwieser, S. 62).

Als „inhaltlichen Kern“ des Informatikunterrichts beschränkt Hubwieser sich auf die Modellierung (Hubwieser, S. 85). Zwar spielen auch Schubert/Schwill kurz mit dem Gedanken der Informatik als „Wissenschaft von der Modellbildung“ (Schubert/Schwill, S. 99), besinnen sich dann jedoch auf deren Aufgabe als „Entwurf und Gestaltung von Informatiksystemen“ (s. o.). Die Lösung dieser Aufgabe besteht eben nicht nur aus der Teilphase der Modellierung, sondern auch aus den übrigen Entwicklungsphasen. „Informatisches Modellieren ist eine wichtige Komponente der informatischen Bildung, darf aber weder inhaltlich noch zeitlich dominieren“ (Schubert/Schwill, S. 254). Schon andere Autoren – wie auch beispielsweise die Rezensenten im didaktischen *„Inf-Phasen-Schema“* (vgl. Koerber/Peters, 1989) – haben diese Einsicht bzw. Forderung vertreten.

## *Fachdidaktik*

Verdienstvoll an dem Werk von Schubert/Schwill ist, dass die Auto-

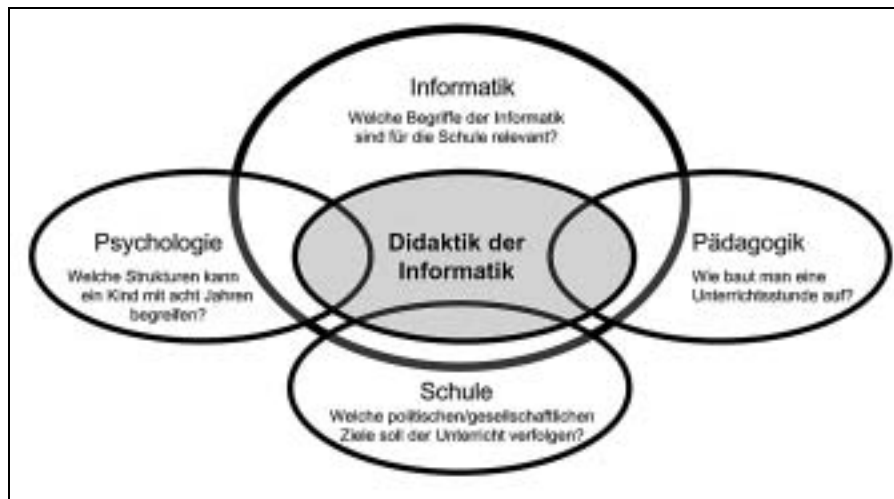
ren die von ihnen vertretenen Grundsätze deutlich benennen und nicht – wie vielfach in solchen Werken üblich – verschleiern. So wird durch die Abbildung auf Seite 18 das Selbstverständnis der Autoren von der Fachdidaktik Informatik insofern deutlich, als sie die Didaktik der Informatik ausschließlich in die Wissenschaft Informatik einbetten (s. Bild, nächste Seite). Zwar gibt es Schnittstellen (bzw. Schnittmengen) zur Psychologie, zur Pädagogik und zur Institution Schule, aber diese drei Bereiche werden hier zum Teil von der Fachwissenschaft Informatik selbst überdeckt und zum geringeren Teil dann von der Fachdidaktik.

Nach Auffassung der Rezensenten ist aber Fachdidaktik mehr: Zum einen ist sie eine Konkretisierung der (allgemeinen) Didaktik als Wissenschaft vom Lehren und Lernen für ein bestimmtes Fach bzw. für eine Fächergruppe. Zum zweiten steht Fachdidaktik in engem Zusammenhang mit der (allgemeinen) Didaktik als Unterdisziplin der Erziehungswissenschaft. Und zum dritten umfasst die Fachdidaktik damit auch Aspekte, die über die ausschließliche Einbettung in die Bezugswissenschaft hinausragen.

Das bedeutet, dass in der Fachdidaktik nicht nur danach gefragt wird, welche aus der Fachwissenschaft ausgewählten Inhalte wie unterrichtet werden sollten, sondern die Fachdidaktik sich insgesamt damit beschäftigt, „wer was wann mit wem wo wie womit warum und wozu lernen soll“ (vgl. Jank/Meyer, <sup>4</sup>1997). Fachdidaktik ist deshalb eng mit gesellschaftlichen Bezügen verknüpft, die über die jeweilige Bezugswissenschaft hinausreichen. Daher forderte Klafki (<sup>5</sup>1996) beispielsweise, epochaltypische Probleme in den Mittelpunkt des Unterrichts zu stellen. Was „epochaltypisch“ ist, wird nicht nur durch die Lebenswelt der Schülerinnen und Schüler bestimmt, sondern auch – zeitgebunden – durch grundlegende gesellschaftliche Probleme, die damit in den Unterricht einzubeziehen und von Fachdidaktikern zu reflektieren sind.

## *Beziehungen der Informatik zu anderen Schulfächern*

Schubert/Schwill betonen die Wichtigkeit der Kooperation der



## Einbettung der Didaktik der Informatik gemäß Schubert/Schwill, 2004, S. 18.

senschaftlicher Herkunft zusammengewirkt und ihre Terminologie nicht überall angelegentlich haben. So wird beispielsweise der Begriff *Aggregation* auf S. 164 als „das gleichrangige Zusammensetzen mehrerer Datentypen zu einem einzigen (kartesischen Produkt)“, auf S. 203 dagegen als „Sonderform der Assoziation“ bezeichnet. Wie beides zusammenhängt, bleibt dem Scharfsinn des Lesers überlassen. Diese Uneinheitlichkeit bzw. Widersprüchlichkeit ist für zahlreiche objektsprachliche (z. B. Implementations, Schnittstelle, Konstruktor, Generalisation, polymorph usw.) und metasprachliche Termini (z. B. Idee, Prinzip, Methode, Paradigma, Abstraktion, Kategorie, Schema, Problem, Modell usw.) zu konstatieren. Zum Aufbau einer (didaktischen) Fachterminologie haben weder Schubert/Schwill noch Hubwieser Vorschläge formuliert. Hier liegt noch ein erhebliches Desiderat fachlicher und didaktischer Forschung (vgl. Glossar zur Objektorientierung, in diesem Heft, S. 93–99). Denn dies muss bei den sich in allen Bundesländern abzeichnenden zentral vorgegebenen Prüfungen und den sich entwickelnden Bildungsstandards nunmehr mit hoher Priorität gesehen werden, damit entsprechende Prüfungsaufgaben für alle Schülerinnen und Schüler terminologisch eindeutig sind.

Ähnliches gilt für einige Unterrichtsbeispiele. Auf S. 156 wird von Schubert/Schwill ein Beispiel zur Ermittlung kürzester Wege in einem bewerteten Graphen vorgestellt; es hätte sehr nahe gelegen, das Problem mit den kurz zuvor entwickelten prädikativen Lösungsmethoden (PROLOG) zu behandeln. Oder es wird einerseits auf ein für JAVA ungünstiges didaktisches Urteil verwiesen und damit offenbar von der Verwendung dieser Programmiersprache abgeraten (Schubert/Schwill, S. 175). Aber andererseits wird später ein – offenbar positiv bewertetes – Unterrichtsbeispiel in JAVA vorgestellt. Diese

Informatik mit anderen Schulfächern und weisen der Informatik diesbezüglich eine Sonderstellung ein, die in der Informatisierung aller Fachgebiete begründet sei (Schubert/Schwill, S. 67 ff., 133 f.). In diesem Zusammenhang betonen sie den Aspekt der Informatik als Basis einer neuen Kulturtechnik neben dem Lesen, Schreiben und Rechnen. Hubwieser geht hierauf (nur) indirekt insofern ein, als Wissen bzw. Information (Hubwieser, S. 135) und Modellbildung in allen Fachgebieten eine zentrale Rolle spielen und die Informatik in der Realisierung (auf Informatiksystemen) der von den Fachwissenschaften erstellten Modelle eine grundlegende Rolle zufällt. Das jeweilige Produkt schafft eine „künstliche Welt, die der durch ein Programm gesteuerte Prozess [erzeugt]“ (Schubert/Schwill, S. 155). Aus dem Scheitern der so genannten „Informationstechnischen Grundbildung“, bei der die informatischen Inhalte in andere Fächer integriert werden sollen und damit „wegintegriert“ wurden, ziehen Schubert/Schwill den berechtigten Schluss, dass nur mithilfe eines Unterrichtsfachs Informatik die Grundlagen gelegt werden können, die das Einbringen einsichts- und verständnisvoller Fähigkeiten und Fertigkeiten in die anderen Fächern möglich machen.

### Unterrichtsmittel (Programmiersprachen – Lernumgebungen)

Für „programmiersprachliches Denkschema“ verwenden Schu-

bert/Schwill die Bezeichnung „Paradigma“ und empfehlen, im Unterricht zwei Paradigmen, etwa das prozedurale (als erstes) und das prädikative (als zweites) vorzustellen (Schubert/Schwill, S. 213). „Auf den imperativen Baukasten kann man bis auf weiteres nicht verzichten. Jedoch sollte man dann stets die objektorientierten Erweiterungen imperativer Sprachen wählen“ (Schubert/Schwill, S. 173). „Der prädikative Baukasten eignet sich zum Einstieg in die Informatik vermutlich besser als der funktionale Ansatz. Auf höherem Niveau scheint jedoch der funktionale Baukasten überlegen“ (Schubert/Schwill, S. 174). Hubwieser weist ergänzend darauf hin, dass auch im Schulunterricht (sogar der Mittelstufe) funktional gedacht und modelliert werden kann, etwa bei der Verwendung von Tabellenkalkulationsprogrammen (Hubwieser, S. 192). Die funktionale Denkweise ist naturgemäß der Mathematik eng verhaftet.

Als innovativsten Beitrag zur Didaktik der Informatik sehen Schubert/Schwill vermutlich die Entwicklung so genannter „didaktischer Systeme“ und ihre Überlegungen zu „Entwurfsmustern für Lernprozesse“ an (Schubert/Schwill, S. 144; vgl. auch Brinda, in diesem Heft, S. 103 ff.). Was davon zu halten ist, wird die Zukunft zeigen.

### Inkonsistenzen

An etlichen Stellen ist zu erkennen, dass mit Schubert und Schwill zwei Autoren unterschiedlicher wis-

mangelnde Absprache zwischen den Autoren hat natürlich auch etwas Sympathisches insofern, als der gleiche Gegenstand aus unterschiedlicher Sicht betrachtet und bewertet wird; dem Leser wird keine abgeklärte Meinung aufgezwungen; es bleibt ihm vielmehr die Freiheit, sich das passende Argument für die eigene didaktische Entscheidung herauszusuchen.

## Fazit

Gerade weil die Informatik in der Schule derzeit ein Schattendasein fristet und offensichtlich noch weiter zurückgedrängt werden soll, ist dieses Buch bedeutsam, da es den Bildungswert des Informatikunterrichts aufzeigt und dessen Befürwortern gewichtige Argumente an die Hand gibt. Den Informatiklehrenden können die Vorschläge zur Konzeption von Unterrichtssequenzen und Stundenentwürfen wertvolle Hilfestellungen geben. Insgesamt handelt es sich bei diesem Buch um eine ideenreiche Neuerscheinung, mit der die didaktische Diskussion um ein gutes Stück voran gebracht wird.

Rüdeger Baumann  
Bernhard Koerber

## Literatur

Bußmann, H.; Heymann, H.-W.: Computer und Allgemeinbildung. In: Neue Sammlung, 27. Jg. (1987), H. 1, S. 2–39.

Hubwieser, P.: Didaktik der Informatik – Grundlagen, Konzepte, Beispiele. Berlin; Heidelberg: Springer-Verlag, 2000.

Jank, W.; Meyer, H.: Didaktische Modelle. Berlin: Cornelsen/Scriptor, 1997.

Klafki, W.: Neue Studien zur Bildungstheorie und Didaktik – Zeitgemäße Allgemeinbildung und kritisch-konstruktive Didaktik. Weinheim; Basel: Beltz, 1996.

Koerber, B.; Peters, I.-R.: Grundzüge einer Didaktik der Informatik. In: Senatsverwaltung für Schule, Berufsbildung und Sport (Hrsg.), Berlin 1989.

Lehmann, G.: Ziele im Informatikunterricht. In: LOG IN, 12. Jg. (1992), H. 1, S. 26–30.

## Info-Markt

### Mediale Bilderwelten im Wettbewerb

7. bundesweiter  
Multimediawettbewerb  
ausgeschrieben

*MB21 – Mediale Bilderwelten* ist ein bundesweiter Multimediawettbewerb für Jugendliche bis 21 Jahre. Dieser Wettbewerb ist in diesem Jahr zum siebenten Mal ausgeschrieben und mit dem *Δ deutschen Jugend-Multimedia-Preis* verbunden worden. Der Ausschreibungszeitraum läuft von

**Juli 2004 bis Februar 2005,**

wobei die eingesandten Beiträge nicht vor dem Jahr 2004 produziert sein dürfen.

Teilnehmen können alle Schülerinnen, Schüler und Jugendliche im Alter von 10 bis 21 Jahren, wobei das Alter der Erstellung des eingesandten Wettbewerbsbeitrags zählt. Zugelassen sind Arbeiten von Einzelpersonen, aber auch von Gruppen. Maximal dürfen drei Beiträge pro Gruppe oder pro Einzelteilnehmerin bzw. -teilnehmer eingesandt werden. Es gibt zwei Wettbewerbsbereiche:

▷ *MB21 – Mediale Baustellen:* Angenommen werden Konzepte für

multimediale Projekte aller Art. Das bedeutet, dass auch Ideen oder unfertige Produktionen eingesandt werden können. Die Gewinner dieses Wettbewerbsbereichs erhalten die Möglichkeit, mit Profis ihre Konzepte zu diskutieren und sie in Workshops zu bearbeiten oder zu realisieren.

▷ *MB21 „ex-tra“:* Dies ist der Bereich für „Experten“. Es können fertige Beiträge in folgenden drei Sparten in den Alterskategorien I (10–16 Jahre) und II (17–21 Jahre) eingereicht werden:

- *Animationsfilm:* Computeranimationen, die mit 2-D- oder 3-D-Software hergestellt wurden und deren Laufzeit höchstens 15 Minuten beträgt.

- *Webseiten:* Online vorzufindende Webseiten, die die Möglichkeiten des Internets ausschöpfen. Sie sollen vor allem die Nutzer zum Eingreifen und zur Kommunikation auffordern.

- *Interaktive Medien:* Interaktive Multimedia-Präsentationen, die eine gute Mischung von Text, Bild, Ton, Animation und filmischen Elementen bieten. Eingereicht werden können nur selbstlaufende Programme auf CD-ROM, DVD o. Ä.

Weitere Informationen sind zu finden bei

<http://www.mb21.de/bund/>

Insgesamt stehen Sach- und Geldpreise sowie die kostenfreie Teilnahme an Workshops im Wert von 20000 Euro zur Verfügung.

koe

**Für Schülerinnen, Schüler und Jugendliche im Alter von 10 bis 21 Jahren wird der Multimediawettbewerb bundesweit ausgeschrieben.**

