

# Von der Funktion zum Objekt – Informatik für die Sekundarstufe 1

Peter Hubwieser

Fakultät für Informatik  
Technische Universität München  
Boltzmannstr. 3  
D-85748 Garching  
<http://ddi.in.tum.de>  
[Peter.Hubwieser@in.tum.de](mailto:Peter.Hubwieser@in.tum.de)

**Abstract:** Ausgehend von einigen grundsätzlichen Postulaten für einen allgemein bildenden Informatikunterricht in der Sekundarstufe 1 wird ein möglicher Lernweg skizziert, der von einem ersten Funktionsbegriff (Black-Box-Sicht) ausgehend über Datenmodellierung, Zustandskonzepte und Algorithmusbegriff zu einer Glass-Box-Sicht auf Funktionen führt und damit alle Voraussetzungen für die Beherrschung objektorientierter Konzepte schafft. Alle erlernten Konzepte werden dabei möglichst früh in praktischer Arbeit mit passenden Werkzeugen (aus dem Standardsoftware- bzw. Programmiersprachenbereich) veranschaulicht, angewandt, erprobt und vertieft. Der Schwerpunkt der Darstellung liegt dabei auf den funktionalen Konzepten, die sich derzeit in einer ersten Erprobungsphase für den flächendeckenden Einsatz (ab 2007 im regulären bayerischen Pflichtfach Informatik in den Jahrgangsstufen 8–10) befinden.

## 1 Einleitung

Von einem Hauptvortrag auf einer bedeutenden Fachkonferenz erwartet man grundsätzliche Aussagen zu aktuellen Fragestellungen der jeweiligen Fachgemeinde. In diesem Beitrag möchte ich versuchen, diesem Anspruch in zweifacher Hinsicht gerecht zu werden: Einerseits werde ich eines der seltenen Konzepte für ein Pflichtfach Informatik in der Sekundarstufe 1 (Jgst. 6–10) beschreiben, das den Weg in die regulären Stundentafeln der Gymnasien gefunden hat. Andererseits will ich typische Arbeitsmethoden einer praxisorientierten Fachdidaktik beschreiben, indem ich über ein soeben abgeschlossenes Projekt zur Erprobung der o. g. Unterrichtskonzepte für unser neues bayerisches Pflichtfach berichte.

Der Anfangsunterricht (Jgst. 6 und 7) befindet sich bereits im Alltagsbetrieb. Er wurde ausführlich erprobt und evaluiert, wie wir das an anderer Stelle bereits beschrieben haben (siehe [Fr01]). Derzeit erproben wir daher vor allem die Mittelstufe, die im Herbst 2007 mit der 8. Jahrgangsstufe in Betrieb gehen wird. Bis dahin muss die geplante Unterrichtskonzeption mit Material (u. a. mit Unterrichtswerken) hinterlegt, getestet und evaluiert werden. In diesem Vortrag soll der Schwerpunkt auf die Erprobung eines Moduls zur funktionalen Modellierung gelegt werden, da wir hierzu die jüngsten und umfangreichsten Ergebnisse vorliegen haben.

## 2 Das Pflichtfach Informatik in Bayern

### 2.1 Das neue achtjährige Gymnasium

Nach einigen heftigen Turbulenzen, die vor allem durch die handstreichartige Verkürzung der Ausbildungsdauer ausgelöst wurden, ist das neue 8-jährige bayerische Gymnasium (kurz: G8) ab dem Schuljahr 2004/05 in den Jahrgangsstufen 5 und 6 in Betrieb gegangen (siehe <http://www.g8-in-bayern.de>). Dazu wurde der gesamte Lehrplan, der unmittelbar vor der Verkürzung bereits (für das herkömmliche 9-jährige Modell) in allen Fächern und Jahrgangsstufen völlig neu konzipiert worden war, nochmals gestrafft und intensiv überarbeitet. Die endgültige Form der Oberstufe ist allerdings derzeit noch nicht vollständig festgelegt. Für das neue Pflichtfach Informatik (siehe [Hub03a], [Hu03c]) brachte diese Verkürzung einige wesentliche Veränderungen mit sich:

- Der ursprünglich als selbstständiges, zweistündiges Fach in der 6. Jahrgangsstufe geplante Anfangsunterricht wurde (inhaltlich unverändert) jeweils einstündig auf die 6. und 7. Jahrgangsstufe des neu geschaffenen Fächerverbundes „Natur und Technik“ (aus Physik, Biologie und Informatik) aufgeteilt. Der Unterricht soll allerdings nur von Lehrkräften mit voller Lehrbefähigung im jeweiligen Fachgebiet erteilt werden.
- Die ursprünglich für die Jahrgangsstufen 9–11 der mathematisch-naturwissenschaftlichen Ausbildungsrichtung geplante Mittelstufeninformatik wurde inhaltlich angepasst und auf die Jahrgangsstufen 8–10 vorgeschoben. Sie bildet dort als zweistündiges Vorrückungsfach zusammen mit Chemie das Profil des nun in „naturwissenschaftlich-technologisch“ umbenannten Zweiges.

### 2.2 Postulate für ein Pflichtfach Informatik

Die Durchsetzung des neuen Faches Informatik gelang vor allem deshalb, weil wir im Vorfeld einen relativ stabilen Konsens über die folgenden Postulate hergestellt hatten [Hu03a]:

- Der Informatikunterricht an weiterbildenden Schulen muss möglichst früh einsetzen, um die ansonsten sehr unterschiedliche Entwicklung der IT-Kompetenz (Jungen vs. Mädchen, begütert vs. armes Elternhaus, Einwanderer vs. Ansässige etc.) noch ausgleichen zu können. Ein Informatik-Anfangsunterricht, der erst in der gymnasialen Oberstufe einsetzt, ist nach unseren Erfahrungen nur sehr schwer für alle Teilnehmer nutzbringend zu gestalten.
- Modellierung vor Programmierung: Der Bildungswert der Modellierung ist in der Fachdidaktik seit langem anerkannt. Allerdings ermöglichte erst die Entwicklung standardisierter grafischer Modellierungstechniken, die sich mittlerweile in der UML stabilisiert hat, einen schülergemäßen, handlungsorientierten Unterricht auf der Basis von Modellierung. Programmierung im Sinne von Codierung dient nur noch zur Umsetzung bzw. Realisierung von sorgfältig entwickelten Modellen, um

diese zu simulieren und damit zu testen und zu validieren. Damit wird auch sichergestellt, dass grundlegende informatische Konzepte nicht mehr nur auf der Ebene einer speziellen Programmiersprache vermittelt, sondern auf einer modellbasierten Ebene sprachübergreifend verstanden werden.

- Tiefgreifendes Verständnis statt „Benutzerfertigkeit“: Im Gegensatz zu „Führerscheinkonzepten“ wie dem ECDL geht es im Informatikunterricht nicht vorrangig um die Bedienung konkreter Softwaresysteme (oder auch Programmiersprachen), sondern um übertragbare, langlebige Konzepte, die (zumindest voraussichtlich) für viele konkrete Systeme auch in fernerer Zukunft Bedeutung haben werden. Der Unterricht muss sich daher (vor allem sprachlich) immer auf einem gewissen Abstraktionsniveau bewegen.
- Werkzeug in Reichweite: Neben dem Verständnis grundlegender Konzepte der Informatik muss der Informatikunterricht auch die *Beherrschung* (d. h. hier die gedankliche Durchdringung) von modernen Softwaresystemen fördern. Daher muss der praktischen Arbeit am Rechner ein erheblicher Teil der Unterrichtszeit gewidmet werden. Die Erstellung von Dokumenten aller Art wie Grafiken, Texte oder auch Rechenblätter, Datenbanksysteme und Programme fördert auch die Kreativität und die Kooperationsfähigkeit der Schülerinnen und Schüler sowie moderne Unterrichtsmethoden. Insbesondere sollte abgesichert werden, dass alle theoretischen Konzepte möglichst sofort in Software umgesetzt und so angewandt oder veranschaulicht werden können.
- Projektorientierung: Soweit irgend möglich, sollen alle Lernvorgänge innerhalb von Projekten stattfinden. Lehrervorträge sollten nur in sehr kurzer Form und nur an absolut unvermeidlichen Stellen angesetzt werden.

### 2.3 Der Anfangsunterricht

In den Jahrgangsstufen 6 und 7 beschäftigen sich die Schülerinnen und Schüler vor allem mit der objektorientierten Sicht auf grundlegende Datenstrukturen von Standardsoftwaresystemen. Dies geschieht in altersgemäßer, spielerischer Weise unter intensivem, kreativem Einsatz dieser Software. Nach der Einführung der Begriffe „Objekt“, „Klasse“, „Attribut“, „Methode“ anhand von Grafiken lernen sie an Texten, Ordnern und Hypertexten typische Beziehungen zwischen Objekten wie die Aggregation oder „verweist auf“ kennen. Den Abschluss des Kurses bildet eine Einführung in die Algorithmik unter Verwendung von virtuellen oder realen Robotersystemen (siehe <http://www.schule.bayern.de/karol>). Eine umfassende Darstellung des zweijährigen Kurses findet sich in unserem dafür verfassten Unterrichtswerk [FHW04], ein kurzer Überblick in [Hu03b].

### 2.4 Objektorientierung als Einstieg?

Gegenwärtig findet man oft objektorientierte Konzepte als Einstieg in den Informatikunterricht. Auch wir setzen ja im Anfangsunterricht auf die objektorientierte Sichtweise auf

Datenstrukturen bei der Modellierung von Standardsoftware (siehe [Hu03b]). Allerdings sind wir nicht der Meinung, dass sich objektorientierte *Programmierung* für den Anfang des Informatikunterrichtes eignet. Dabei werden die Schüler bereits in ihrem ersten Programm gleichzeitig mit einer Vielzahl von schwierigen Konzepten (Variable, Funktion, Verbund, Zeiger/Referenz) konfrontiert, was sie oft überfordert oder zumindest das Verständnis erschwert. Andere Autoren wie [Bö01] haben diese Problematik bereits im Detail diskutiert. Auch weisen unsere Erfahrungen mit objektorientierten Konzepten im Grundstudium Informatik in diese Richtung (siehe [Sc04]). Für den Informatikunterricht am Anfang der Mittelstufe gewinnen diese Argumente nochmals an Gewicht. Wir setzen daher eher auf einen synthetischen Weg, indem wir versuchen, die grundlegenden Konzepte einzeln nacheinander einzuführen und auch einzüben, bevor sie (erst im dritten Jahr des Mittelstufenkurses) in der Objektorientierten Modellierung und Programmierung zusammengeführt werden. Als wichtigstes Konzept hat sich dabei der Funktionsbegriff herausgeschält. Er wird später u. a. im Datenbankenbereich für die Semantik der Abfragen, in der imperativen Programmierung für die Modularisierung von Programmen, für Komplexitätsbetrachtungen und zum Verständnis von Rekursion sowie im OO-Bereich für die Definition von Methoden benötigt. Da er im neuen Mathematiklehrplan der 8. Jahrgangsstufen erst im späteren Verlauf des Schuljahres vorgesehen ist, kommt erstmals der Informatik die Aufgabe zu, diesen Begriff einzuführen. Wir sind der Meinung, dass die Informatik mit ihrer operationalen Sichtweise dies auch gut und nachhaltig zu leisten vermag.

## **2.5 Die inhaltliche Konzeption der Mittelstufe (Jahrgangsstufe 8–10)**

Während der Konzeption des Anfangsunterrichtes sah sich die Lehrplankommission einem steigenden Druck zur Integration von Tabellenkalkulationssystemen ausgesetzt. Um diese jedoch auf einem ausreichend abstrakten Niveau behandeln zu können, wäre entweder das Vererbungskonzept (für eine objektorientierte Sicht, siehe [Vo05]) oder der Funktionsbegriff (siehe [Hu04]) notwendig. Beide sind in der Unterstufe unserer Meinung nach nicht zu vermitteln, weshalb wir dort auf Tabellenkalkulationen verzichten mussten. Um diese Lücke in der Palette von Standardsoftware möglichst bald zu schließen, sollten diese Systeme daher gleich am Anfang der 8. Jahrgangsstufe behandelt werden. Andererseits wollten wir dort auch mit dem Funktionsbegriff beginnen, so dass sich die Verbindung dieser beiden Themen geradezu aufdrängte. Als Beschreibungstechnik nutzen wir (eingeschränkte) Datenflussdiagramme, mit denen Terme und Rechenblätter strukturiert werden. Die Funktionen werden überwiegend in einer Black-Box-Sicht betrachtet.

Wie eine nähere Untersuchung der unterrichtlichen Möglichkeiten zeigte, führte diese Verbindung letztlich zu einem originellen und fruchtbaren Einstieg in den Informatikunterricht (siehe [Hu04], [Sc05]). Im Abschnitt 4 dieses Beitrags werden wir uns ausführlich mit unseren jüngsten Unterrichtskonzepten und -erfahrungen hierzu beschäftigen.

Der zweite Abschnitt der 8. Jahrgangsstufe ist der Datenmodellierung und den Datenbanksystemen gewidmet. Wir schlagen vor, mit der Erstellung und Abfrage einer Tabelle zu beginnen, bevor mittels Entity-Relationship-Modellierung komplexere Datenstruk-

turen beschrieben und auf relationale Datenbanken umgesetzt werden. Die größte didaktische Herausforderung stellen dann Abfragen über mehrere Tabellen dar. Die Anforderungen an ein Datenbankschema, Betrachtungen zu Datensicherheit und Datenschutz sowie ein komplexes Anwendungsbeispiel (Projekt) beschließen dann diese 8. Jahrgangsstufe.

Die 9. Jahrgangsstufe beginnt mit der Beschreibung von Abläufen (z. B. von realen Automaten wie Fahrscheinautomaten oder Videorekordern) mit Hilfe von Zustandsmodellierung und Algorithmen. Mithilfe des Zustandskonzeptes kann dann im Rahmen der imperativen Programmierung sowohl der Variablenbegriff sauber eingeführt als auch die Semantik der Programme exakt beschrieben werden. Schließlich sollen die Schülerinnen und Schüler in der Lage sein, ihre Zustandsmodelle durch schematisch erstellte imperative Programme zu simulieren.

Anhand der Gliederung von komplexeren Systemen in Teilsysteme nach dem Prinzip „teile und herrsche“ und der Abbildung dieser Teilsysteme in Programmmodule werden die funktionalen Konzepte aus der 8. Jahrgangsstufe wieder aufgegriffen und vertieft, wobei wir die verwendeten Funktionen und Prozeduren nun allerdings in Glass-Box-Sicht behandeln. Hier begegnen die Schülerinnen und Schüler erstmals auch rekursiven Algorithmen.

Eine wichtige Rolle spielt an dieser Stelle die Spezifikation von Schnittstellen und die arbeitsteilige Entwicklung von Programmen. Mithilfe dieser Kenntnisse können nun auch umfangreichere Algorithmen entwickelt und eingehend untersucht werden.

In der 10. Jahrgangsstufe steht die Vertiefung der Objektorientierten Sichtweise (aus der Unterstufe) und die Umsetzung der Modelle in objektorientierte Programme im Vordergrund. Nach der Zusammenfassung und Festigung der bisher erlernten objektorientierten Konzepte lernen die Schülerinnen und Schüler, Interaktion zwischen Objekten zu beschreiben und zu programmieren. Das Konzept der Vererbung erlaubt es ihnen, komplexere Objektstrukturen zu beschreiben und effizient zu programmieren. Schließlich sind sie in der Lage, größere Systeme zu modellieren und durch Programme zu simulieren. Dabei werden sie gezwungen, kooperative Arbeitsabläufe zu organisieren und einzuüben.

Am Ende der Jahrgangsstufe bietet ein größeres Softwareprojekt (z. B. Buchungssystem, Geschäftsabläufe einer Bank, Simulation eines idealen Gases) den Jugendlichen Gelegenheit, alle bisher erlernten Modellierungstechniken zusammenzuführen und zur Beschreibung eines komplexeren Systems aus mehreren Sichten zu nutzen.

### **3 Unterrichtserfahrungen**

#### **3.1 Bayerische Schulversuche**

Die geplanten Inhalte des neuen Faches Informatik wurden auf breiter Front seit dem Schuljahr 2000/2001 erprobt. Der Großteil der Probeläufe fand im Rahmen der Schul-

versuche „Europäisches Gymnasium Typ 3“ und „Mathematisch-naturwissenschaftliches Gymnasium Typ 2“ statt, an denen jeweils ca. 20 Gymnasien beteiligt waren. Die Ergebnisse wurden u. a. von der Universität Dortmund (siehe [Fr01]) evaluiert.

Leider waren in den Stundentafeln dieser Schulversuche für die Informatik in der Mittelstufe nur die Jahrgangsstufen 10 und 11 vorgesehen. Die 9. Jahrgangsstufe wurde erst nach der Entscheidung der Bayerischen Staatsregierung, die Informatik flächendeckend (zunächst für das damalige 9-jährige Gymnasium) einzuführen, hinzugenommen. Dies hatte natürlich eine Erweiterung des Lehrplans zur Folge, in deren Rahmen u. a. der hier schwerpunktmäßig besprochene Themenbereich „Funktionale Modellierung“ entstand. Folglich konnte dieser Teil des Lehrplans bisher nicht in den Schulversuchen erprobt werden.

### **3.2 Rahmenbedingungen der Erprobung**

Die Erprobung verspricht nur dann einigermaßen valide Ergebnisse, wenn sie im Rahmen eines regulären Pflichtfaches stattfinden kann. Die einzige Möglichkeit dafür bot sich uns im Rahmen des so genannten „Additums“ aus dem Mathematikunterricht der 9. bis 11. Jahrgangsstufen des mathematisch-naturwissenschaftlichen Zweigs. Da einerseits der Mathematikunterricht in diesem Zweig umfangreicher ist als in den anderen Zweigen, andererseits aber die Schülerinnen und Schüler aller Zweige in der Kollegstufe wieder in denselben Grund- und Leistungskursen zusammengeführt werden, bietet dieses Additum zusätzlichen Stoff an, der später aber nicht vorausgesetzt werden darf. Wir ließen uns daher vom Bayerischen Staatsministerium für Unterricht und Kultus die Genehmigung erteilen, im einstündigen Additum der 9. Jahrgangsstufe Teile des zukünftigen Lehrplans der 8. Jahrgangsstufe zu erproben, da keine Möglichkeit für eine Erprobung in der 8. Jahrgangsstufe in Sicht war. Bei den Ergebnissen ist also zu bedenken, dass die Schülerinnen und Schüler in der flächendeckenden Umsetzung ein Jahr jünger sein werden.

Markus Schneider führte im Schuljahr 2003/04 mit einer Klasse bereits Vorversuche durch (siehe [Sc05]). Im Schuljahr 2004/05 unternahm ich dann selbst an einem Gymnasium in einer südbayerischen Kleinstadt (der Name soll hier aus Datenschutzgründen nicht genannt werden) in zwei Klassen einen umfangreicheren Lehrversuch über 13 Unterrichtsstunden (je Klasse). Zusätzlich wurde jeweils eine Stunde für die Einführung sowie für den Abschlusstest verbraucht. Im Lehrplan sind für das Themengebiet 18 Stunden vorgesehen, so dass man davon ausgehen kann, dass alle im Versuch erprobten Inhalte auch flächendeckend durchgeführt werden können. Die beiden Klassen bestanden annähernd zu gleichen Teilen aus Jungen und Mädchen (9-I: 15 Mädchen und 16 Jungen, 9-II: 14 Mädchen und 13 Jungen).

Die Erprobung musste in zwei Phasen aufgeteilt werden: Vom 24.11. bis zum 2.2. führte ich 8 Stunden im einstündigen und vom 12.4. bis zum 27.4. dann im zweistündigen Wochenrhythmus durch. Der Grund für die Umstellung war die Erfahrung, dass durch einzelne (meinerseits berufsbedingte) Unterrichtsausfälle und Ferienzeiten die Stundendichte oft unter ein erträgliches Maß absank.

### **3.3 Unterrichtsmethodik**

Bei der Konzeption des neuen Faches und während der gesamten Lehrplanarbeit war es uns ein großes Anliegen, das Fach so anzulegen, dass es schülerzentriert, handlungsorientiert und mit möglichst wenig Lehrervorträgen umgesetzt werden kann. Daher achtete ich auch bei der Erprobung darauf, möglichst alle Lernvorgänge durch konkrete Aufgabenstellungen anzustoßen. Die Lehrervorträge beschränkten sich auf maximal 5-8 Minuten pro Unterrichtsstunde. Daneben wurden vor allem Gruppenarbeiten und Partnerarbeiten durchgeführt.

Um den Schülerinnen und Schülern möglichst viel für den weiteren Mathematikunterricht mitzugeben, wählte ich möglichst mathematische Beispiele und Aufgaben.

Als roter Faden diente die schrittweise Erstellung eines Rechenblattes für Zeugnisnoten in allen Fächern. Diese Aufgabe wurde durch einen Eingangstest (Berechnung des Durchschnittswertes von fünf Schulaufgaben) eingeleitet und fand unter Berücksichtigung der unterschiedlichen Berechnungsvorschrift je nach Anzahl der Schulaufgaben eines Faches seinen Abschluss.

### **3.4 Die Technik**

Da die Rechanlage der Schule kurz vor meinen Versuchen auf eine Terminalserverstruktur umgestellt worden war, kam mir die Aufgabe zu, diese Technik im ersten harten „Officeinsatz“ zu erproben. Die Terminals liefen dabei unter Linux, für den Server stand eine Windows- und eine Linux-Lösung zur Verfügung. Dabei stellte sich heraus, dass der Windows-Terminalserver nicht stabil betrieben werden konnte. Insbesondere das gleichzeitige Speichern von Dateien durch mehrere Schüler führte regelmäßig zu Abstürzen. Das war sowohl mit MS-Office als auch mit Star-Office zu beobachten. Wesentlich stabiler lief dann der Linux-Terminalserver mit Star-Office, der allerdings ebenfalls vor jeder Sitzung neu gestartet werden musste, da er offensichtlich nicht mit einer größeren Anzahl von parallelen Prozessen zurechtkam.

Leider trafen die technischen Probleme überwiegend dieselbe Klasse 9-II. Dabei konnte die fehlende Rechenzeit nicht etwa für eine Verstärkung der theoretischen Arbeit herangezogen werden, weil sie meist für die Ursachenforschung bzw. -behebung der technischen Probleme verbraucht wurde.

### **3.5 Der Eingangstest**

Aufgrund der Vorerfahrungen aus den Schulversuchen stellten wir die Hypothese auf, dass die Schülerinnen und Schüler kaum Probleme mit der bloßen Bedienung der Tabellenkalkulationssysteme haben würden. Zur Prüfung dieser Hypothese ließ ich sie in der ersten Stunde ohne jegliche Vorbereitung in Partnerarbeit (mit jeweils einem Rechner pro Schülerpaar) folgende Aufgabe bearbeiten: „Erstelle eine Tabelle, die mit Hilfe der Grundrechenarten die Durchschnittsnote aus 5 Schulaufgabennoten ausrechnet“.

Als Hilfestellung hatte ich zwei Informationsblätter vorbereitet, die in zwei Stufen zur Lösung führen sollten:

- Stufe 1: Lösung in Formelansicht ohne jeglichen Kommentar
- Stufe 2: Hinweise zur Bedienung (Dateneingabe, Gleichheitszeichen als Einleitung einer Formel, etc.)

Die Erfolgsquote konnte nur am Bildschirm und daher nur arbeitsplatzweise festgestellt werden. Sie war wie erwartet in beiden Klassen sehr hoch und annähernd gleich: An insgesamt 29 Arbeitsplätzen (15 in 9-I bzw. 14 in 9-II) waren nach 5 Minuten ohne jegliche Hilfestellung 13 Rechenblätter fertig gestellt, nach weiteren 5 Minuten und Hilfe-stufe 1 auch die restlichen 16. Die schnellsten Paare waren in jeder Klasse bereits nach ca. 1 Minute fertig. Zudem übertrafen in jeder Klasse jeweils 5 Lösungen die Anforderungen. Da Hilfestufe 2 in keinem Fall benötigt wurde, erschien eine Schulung von reinen Bedienerfertigkeiten wie erwartet völlig überflüssig.

### **3.6 Die Vorkenntnisse**

Die betrachteten 9. Klassen hatten bereits im Rahmen des Mathematikunterrichtes des vorausgegangenen Schuljahres den Funktionsbegriff kennen gelernt. Allerdings zeigte sich während der Unterrichtsfolge, dass sie kaum in der Lage waren, diesen ausreichend genau wiederzugeben. Parallel mit der Informatik beschäftigten sie sich im Mathematikunterricht mit quadratischen Funktionen und Gleichungen sowie später mit Quadratwurzeln. Im Gegensatz dazu werden die 8. Klassen in der flächendeckenden Umsetzung im G8 vorher noch nichts von Funktionen, quadratischen Gleichungen oder gar Wurzeln gehört haben.

### **3.7 Die Lernschritte**

In diesem Abschnitt soll kurz die Abfolge der wesentlichen Lernschritte und ihrer methodischen Umsetzung skizziert werden. Weitere Informationen zur Vorgehensweise finden sich in [Hu03b], [Sc05].

Nach dem Eingangstest war klar, dass eine Benutzerschulung an der Tabellenkalkulationssoftware nicht notwendig war. So konnten wir sofort anhand der Aufgabenstellung des Eingangstests die zentralen Begriffe der Tabellenkalkulation klären: Rechenblatt, Zellen und Zelleninhalt (Konstante oder Formel), Datentypen, Zellbezüge.

Nun war der Funktionsbegriff an der Reihe. Eine Gruppenarbeit zeigte, dass aus der Mathematik nur noch vage Vorstellungen davon existierten. In einer weiteren Gruppenarbeit lotete ich die Möglichkeiten aus, zur Abstützung des Funktionsbegriffs den Relationsbegriff einzuführen. Dies gelang relativ gut, nahm aber zwei Stunden in Anspruch, die in der flächendeckenden Umsetzung wohl nicht zur Verfügung stehen werden. Nach diesem Exkurs führten wir die Begriffe *Funktion* (hier als eindeutige Relation), *Argument* und *Funktionswert* ein. Als sehr geeignetes Anfangsbeispiel, das insbesondere die

Black-Box-Sicht gut unterstützte und von Anfang an die Existenz nichtnumerischer Funktionen (die Werte sind Texte) klarstellte, erwies sich die Funktion RÖMISCH (siehe Abbildung 1).

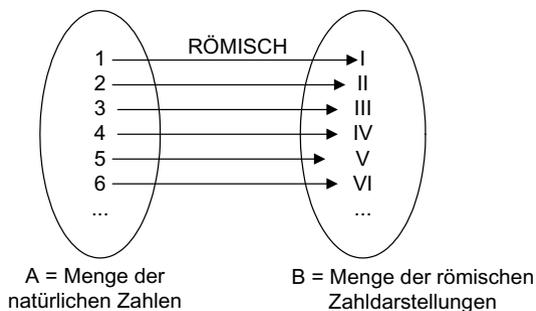


Abbildung 1: Die Funktion RÖMISCH im Mengendiagramm

Nach weiteren Beispielen für Funktionen folgten Überlegungen zur graphischen Darstellung unter ausgiebiger Verwendung der diesbezüglichen Möglichkeiten der Tabellenkalkulation (Linien- bzw. Balkendiagramme etc.). Relativ überraschend war für die Schülerinnen und Schüler dann die Betrachtung der konkreten Berechnungsvorschriften für Funktionen, hier insbesondere der Wurzelfunktion. Dies führte uns zum Algorithmusbegriff.

Nun galt es, den Funktionsbegriff auf Funktionen mit mehreren Argumenten zu erweitern, wofür ich die Beispiele KGV und RUNDEN heranzog. Interessant war die Begegnung mit Parameterlisten variabler Länge bei den Funktionen SUMME und MITTELWERT. Die zweistelligen Rechenoperationen untersuchten wir danach in verschiedenen Darstellungsformen (Infix- bzw. Präfixnotation).

Die Einführung der Verkettung von Funktionen (siehe Abbildung 2) ermöglichte den Schülern schließlich den Aufbau von Termen, deren Implementierung im Rechenblatt durch (didaktisch reduzierte) Datenflussdiagramme dargestellt wurde. Zugelassen waren dabei die folgenden Elemente:

- *Rechtecke* für Ein- bzw. Ausgabezellen,
- *Linien* für Datenflüsse, die im Rechenblatt als Zellbezüge implementiert wurden,
- *Ovale* für informationsverarbeitende Prozesse, hier speziell für Funktionen (deren Verwendung war damit auf eine Datenausgangsleitung eingeschränkt),
- *Doppellinien* für Datenspeicher (in Zellen), die insbesondere für mehrfach zu verwendende Zwischenergebnisse benötigt wurden.

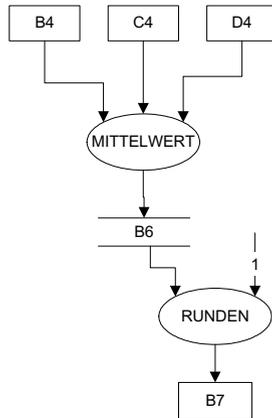


Abbildung 2: Verkettung von Funktionen im Datenflussdiagramm

Nach der anfänglich kleinschrittigen Berechnungsweise der Terme (mit vielen Zwischenergebnissen) wurde die Berechnung der Terme mehr und mehr auf wenige Zellen komprimiert. Dabei erkannten die Schülerinnen und Schüler, dass die Mehrfachverwendung von Zwischenergebnissen (zur Vermeidung von Mehrfachberechnung, siehe Abbildung 3) dieser Kompression natürliche Grenzen setzt.

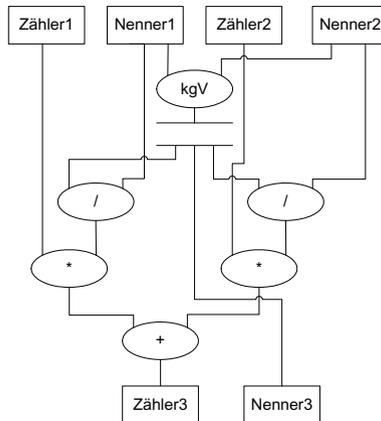


Abbildung 3: Mehrfachverwendung von Zwischenergebnissen am Beispiel der Bruchaddition

Praktisch für viele Berechnungen erwies sich die Verwendung des Bereichsoperators, der hier als zweistellige Funktion in Infixschreibweise (z. B. in A1:B7) betrachtet wurde.

Eine besondere didaktische Herausforderung war die Arbeit mit bedingten Termen unter Verwendung der dreistelligen WENN-Funktion, hier am Beispiel des Abstands zweier Zahlen am Zahlenstrahl (aus dem Abschlusstest):  $d = \text{WENN}(a-b > 0; a-b; -(a-b))$ .

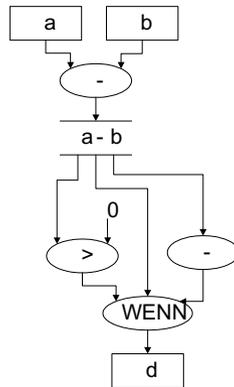


Abbildung 4: WENN-Funktion am Beispiel des Abstands zweier Zahlen

Besonders problematisch erwies sich dabei das Verständnis für den Ablauf der Auswertung dieser WENN-Funktion unter Anwendung des „lazy-evaluation“-Prinzips. Wie aus der funktionalen Programmierung gewöhnt, sind bedingte Terme ja nicht unbedingt strikt. Es kann durchaus vorkommen, dass einer der beiden Zweige der Fallunterscheidung (hier also das 2. oder 3. Argument) keinen definierten Wert liefert. Diese Problematik kann aufgelöst werden, indem jeweils nur der Zweig ausgewertet wird, der aufgrund des Wertes der Bedingung (1. Argument) für das Ergebnis tatsächlich benötigt wird.

Die folgende Übersicht zeigt den Ablauf der Unterrichtssequenz (mit der Nummer der jeweiligen Unterrichtsstunde):

0. Ablauf
1. Vorstellung, Umfrage
2. Test: Erstellung eines Rechenblattes zur Notenberechnung. Definitionen Rechenblatt, Zelle, Datentypen, Formel, Zellbezug
3. Übungen zu Rechenblättern und Datentypen
4. Gruppenarbeit „Funktionen“
5. Gruppenarbeit „Relationen“
6. Funktion als eindeutige Relation, Graphische Darstellung
7. Gruppenarbeit zur Berechnung von Funktionen, Algorithmusbegriff
8. Mehrstellige Funktionen: Postfix, Infixnotation
9. Wiederholung, Verkettung von Funktionen
10. Datenflussdiagramme, Kompression
11. Bereichsfunktion
12. Bedingte Terme
13. Aufgaben
14. Aufgaben
15. Test

### 3.8 Der Abschlusstest mit Auswertung

Unmittelbar nach Abschluss der Unterrichtsserie führte ich mit den beiden Klassen einen Abschlusstest durch. Um die Ernsthaftigkeit der Bearbeitung abzusichern, wurde angekündigt, den Test (ähnlich wie eine Stehgreifaufgabe) als mündliche Note zu werten. Es waren die folgenden Teilaufgaben (TA) zu lösen:

1. *Quadratische Gleichungen:* Hier war die Lösungsformel für quadratische Gleichungen in ein Datenflussdiagramm zu übersetzen. Dabei durften die Schülerinnen und Schüler davon ausgehen, dass für die Diskriminante  $D \geq 0$  gesichert war.
2. *Abstand zweier Zahlen am Zahlenstrahl:* Hier war das in Abbildung 2 gezeigte Datenflussdiagramm aus der allgemeinen Berechnungsvorschrift zu entwerfen.
3. *Zinsen auf Guthaben:* Hier sollte ein Datenflussdiagramm zur Berechnung des Zinsertrags (aus den Eingaben Kapitalbetrag, Zinssatz, Einzahlungs- und Abhebungsdatum) in einen Term übersetzt werden.

Die Arbeitszeit betrug 30 Minuten. Es nahmen insgesamt 55 Schülerinnen und Schüler aus den beiden Klassen teil.

Die drei Teilaufgaben wurden im Verhältnis 14:8:12 gewichtet und brachten ein unterschiedliches Ergebnis: die Schülerinnen und Schüler erreichten in TA 1 durchschnittlich 65%, in TA 2 nur 27% und in TA 3 64% sowie summiert über alle drei Aufgaben 56% der möglichen Punkte. Auffallend war die relativ schlechte Bearbeitung der TA 2, hier vor allem die „digitale“ Punkteverteilung: In 13 (23%) Arbeiten wurden 6-8 Punkte erzielt, in 34 (62%) Arbeiten dagegen 0 Punkte. Offensichtlich hatte nur ein kleiner Teil der Schülerinnen und Schüler gelernt, die WENN-Funktion richtig anzuwenden. Angesichts der sehr kurzen Unterrichtszeit von nur einer Stunde, die dafür aufgewendet werden konnte, war das nicht überraschend.

Der Test erbrachte erwartungsgemäß ein Notenergebnis, das durchaus im Bereich üblicher Mathematikschulaufgaben lag. Ein Vergleich mit den Mathematiknoten aus dem Zwischenzeugnis vom Februar 2005 (im Folgenden kurz als Mathematiknote bezeichnet) ergab eine mittelstarke bis starke positive Korrelation von +0,58.

Die beiden Klassen erzielten dabei unterschiedliche Teilergebnisse, wobei interessanterweise die Klasse 9-II mit den bei weitem häufigeren Ausfällen der Rechenanlage mit einem Notenschnitt von 3,12 (im Vergleich zu 3,60 in 9-I) das bessere Ergebnis erzielte. Dies liegt vermutlich *nicht* an einer eventuellen unterschiedlichen Leistungsfähigkeit der Klassen, da ihre Mathematiksnitte in umgekehrter Reihenfolge sehr nahe zusammen lagen (3,37 bei 9-II gegenüber 3,27 bei 9-I). Eine Erklärung könnte möglicherweise die Ablenkung durch die laufenden Computer liefern.

Leider konnten im Durchschnitt über beide Klassen die Mädchen (3,59) nicht ganz mit den Knaben (3,18) mithalten. Auch die sehr guten Arbeiten kamen mit einem Verhältnis von 5:3 überwiegend von den Jungen. Im Gegensatz dazu sind die Mathematikleistungen nahezu identisch (3,33 bei den Mädchen im Gegensatz zu 3,32 bei den Jungs). Betrachtet

man die Unterschiede jedoch klassenweise, so kommt man zu einem anderen Ergebnis (siehe Abbildung 5). Hier zeigt sich, dass dieser Effekt ausschließlich von den (auch in Mathematik) relativ schwachen Mädchen der Klasse 9-I ausgelöst wurde, während in Klasse 9-II die Mädchen im Test (3,08) sogar etwas besser waren als die Jungs (3,17).

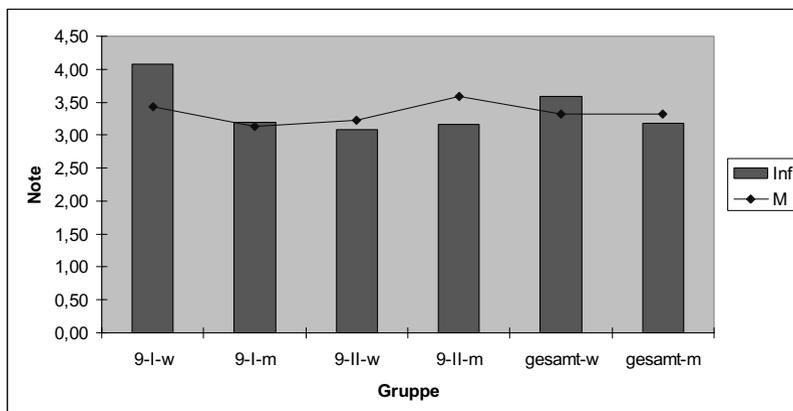


Abbildung 5: Notendurchschnitte in Abhängigkeit vom Geschlecht

### 3.9 Problemstellen des Unterrichtskonzeptes

Um spezielle Problemstellen des Lehrplankonzeptes ausfindig zu machen, habe ich mir die Mühe gemacht, alle Fehler möglichst präzise zu bezeichnen, durchzunummerieren und einen festen Punktabzug pro Fehler festzulegen. Daraufhin wurde für jede Arbeit die jeweils aufgetretene Fehlernummer vermerkt und in ein Rechenblatt eingetragen. Damit konnte die Berechnung der jeweils erreichten Punktzahl je Aufgabe (abgesehen von einem frei einzugebenden „pädagogischen“ Punktezuschlag) vollständig automatisiert werden. Auf diese Weise war ich gezwungen, jeden Fehler genau zu erfassen und zu kategorisieren. Allein die Mühe der Bezeichnung lohnt sich dabei schon, da man sich so weitaus intensiver mit den einzelnen Fehlern beschäftigt. Dem Praktiker kann dieses Vorgehen sehr empfohlen werden, allerdings wegen des enormen Arbeitsaufwandes nur *gelegentlich*. Die häufigsten sieben Fehler waren folgende (Anzahl in Klammern):

1. Einer der alternativen Zweige der WENN-Funktion fehlt. (30)
2. Mehr als eine Ausgabeleitung bei Funktionen. (23)
3. Keine Kästen um Ein-/Ausgaben. (23)
4. Berechnung der Zweige NACH der WENN-Funktion. (15)
5. Überflüssige Instanz der WENN-Funktion. (14)
6. Falsche Reihenfolge der Operatoren (Argumente). (13)
7. Kein eigener Eingang für das zweite Argument des Vergleichs. (13)

Auch hier zeigte sich also, dass die WENN-Funktion die weitaus größten Probleme bereitete.

## 4 Zusammenfassung und Ausblick

In diesem Beitrag wurde vor allem die Erprobung des Anfangsmoduls unseres Mittelstufenkonzeptes beschrieben. Hinsichtlich dieses Moduls erzielten wir dabei zufriedenstellende Ergebnisse, wobei allerdings auch einige besondere didaktische Herausforderungen wie die WENN-Funktion zutage traten. Besonders für diese Stellen werden wir die Lehrkräfte mit umfangreichem und sorgfältig entwickeltem Material unterstützen müssen. Der erste Schritt dazu ist das Unterrichtswerk für die Jahrgangsstufe 8, das in meiner Arbeitsgruppe soeben am Entstehen ist.

Die weiteren Module des Unterrichtskonzeptes sollen in den folgenden Jahren ausführlich erprobt und getestet werden, wobei das Datenmodellierungsmodul (2. Teil der 8. Jahrgangsstufe) am wenigsten Probleme bereiten dürfte, da es in zahlreichen Jahrgangsstufen im Rahmen von Wahl- oder Wahlpflichtkursen erprobt wurde.

Im nächsten Schuljahr werde ich das Konzept für die 9. Jahrgangsstufe (Zustandsmodellierung, Algorithmen und imperative Programmierung) wiederum im Rahmen des Mathematik-Additums einer 9. Jahrgangsstufe testen. Die Ergebnisse sollen ebenfalls in ein Unterrichtswerk einfließen. Für das Schuljahr 2007/08 ist dann die Erprobung des Konzeptes für die 10. Jahrgangsstufe geplant.

## Literaturverzeichnis

- [Bö01] Böszörményi L.: Java für Anfänger? LOG IN 21 (2001) Heft 1; S. 14–19.
- [FHW04] Frey E., Hubwieser P., Winhard F.: Informatik 1. Objekte, Strukturen, Algorithmen. Unterrichtswerk für die 6./7. Jahrgangsstufe, Klett, Stuttgart, 2004.
- [Fr01] Frey E., Hubwieser P., Humbert L., Schubert S., Voß S.: Informatik-Anfangsunterricht. Erste Ergebnisse aus dem Informatik-Anfangsunterricht in den bayerischen Schulversuchen. LOG IN 21 (2001) Heft 1; S. 20-32.
- [Hu03a] Hubwieser P.: Didaktik der Informatik. Grundlagen, Konzepte und Beispiele. Springer Verlag, Berlin, Heidelberg, 2. Überarbeitete Auflage, September 2003.
- [Hu03b] Hubwieser P.: Object Models of IT-Systems Supporting Cognitive Structures in Novice Courses of Informatics. In (van Weert T., Munro R. Eds.): Informatics and The Digital Society: Social, Ethical and Cognitive Issues. IFIP Conference Proceedings 244 Kluwer 2003; S. 129-140.
- [Hu03c] Hubwieser P.: Muss es gleich ein Pflichtfach sein? Die Konzeption des neuen Pflichtfachs Informatik an Bayerns Gymnasien. CD Austria, Sonderheft 10/03 des österr. Bundesministeriums für Unterricht, Wissenschaft und Kultur; S. 4–7.
- [Hu04] Hubwieser P.: Functional Modelling in Secondary Schools Using Spreadsheets. Education and Information Technologies 9 (2) June 2004; 175-183.

- [Sc04] Schneider M.: An empirical study of introductory lectures in informatics based on fundamental concepts; In (Magenheim J., Schubert S. Eds.): Informatics and student assessment; Concepts of Empirical research and Standardisation of Measurement in the Area of Didactics of Informatics; Lecture Notes in Informatics, Seminars, Volume 1, Bonn, 2004; S. 123–133.
- [Sc05] Schneider M.: A Strategy to Introduce Functional Data Modeling at School Informatics. In (Mittermeir, R. ed.): Informatics for Secondary Schools - Evolution and Perspectives: From Computer Literacy to Informatics Fundamentals; Proceedings of ISSEP 2005, Klagenfurt; S. 130–144.
- [Vo05] Voß S.: Informatische Bildung in Anwenderschulungen. Erscheint in diesem Tagungsband.