



Chefredakteur (V.i.S.d.P.):
StD Dr. Ludger Humbert
Redaktion: StD Christian F. Görlich
& Fachseminare Informatik Hamm und Arnsberg
Layout: Ludger Humbert (Vorlage von Torsten Bronger)
Copyright: Für namentlich gekennzeichnete Artikel übernimmt die Autorin die Verantwortung.



SOME RIGHTS RESERVED
Der Inhalt unterliegt der [Creative Commons License](http://creativecommons.org/licenses/by-nc-sa/3.0/deed.de-Lizenz)
by-nc-sa/3.0/deed.de-Lizenz ...
If Fase ... auch im Netz (humbert.in.hagen.de/iffase)
ISSN 1861-0498 – urn:nbn:de:0043-iffase-1

If Fase

IF FASE: INFORMATIKFACHSEMINARE HAMM ARNSBERG

Bildung
Fachseminare
Veranstaltungen
Informatiksysteme
Ausbildungsschulen

If Fase

Termine



Samstag, 24. November 2007

Treffen aller Fachseminargruppen Informatik

9:30 – 14:00
Willy-Brandt-Gesamtschule – Räume F009, F010 – Gebäude: Am Friedriehsberg, Bergkamen
Vorstellung des Projekts **Mobil Programmieren** durch die Schülerinnen und Schüler eines Informatikgrundkurses des elften Jahrgangs im Rahmen des *Tages der offenen Tür*.

INFOS 2007 – Informatiktag

2007/2008 – (Humbert)

(von Dr. Ludger Humbert) Wie an dem Umfang dieser **If Fase** abgelesen werden kann, enthält diese Ausgabe eine Reihe von Beiträgen, die sich mit Veranstaltungen und Workshops der INFOS 2007 und dem dort integrierten sechsten nordrhein-westfälischen Informatiktag 2007 beschäftigen.

Verantwortung der Redaktion

Nach der Veröffentlichung der **If Fase** Nr. 21 mit den ersten Berichten über die INFOS 2007 mit dem integrierten Informatiktag 2007 wurde der Redaktion mitgeteilt, dass einige Beiträge nicht der erwarteten, journalistischen Sorgfalt für eine objektive Berichterstattung entsprächen. Dabei ging es um den Duktus einzelner Berichte und um einige wertende Formulierungen. Daher sei mir gestattet, auf die folgende Punkte hinzuweisen: die Beiträge werden von Berichterstatern/innen erstellt, die ihre individuelle Sicht auf einen Vortrag oder einen Workshop öffentlich mitteilen. Jeder Beitrag wird mit einer entsprechenden Erklärung eingeleitet, damit klar ist, dass es sich um eine durchaus mit eigenen Positionen angereicherte kommentierende Berichterstattung handelt. Jede Leserin und jeder Leser hat die Möglichkeit, einen Leserbrief an die Redaktion zu schreiben und damit eine eigene Meinung kundzutun, die ggf. veröffentlicht wird.

Aufbau der Titel – Bezeichnungswiese Um eine sinnvolle Navigation auf den Artikelwebseiten zu ermöglichen, wird folgende Namensgebung gewählt: »INFOS 2007 – « leitet den Beitragstitel ein – <Bezeichnung für die Veranstaltung/den Bericht> (<Berichterstatler/in>). Im Untertitel finden Sie die Nennung derjenigen, die die Veranstaltung durchgeführt haben.

Informatiktag NW

Eine Übersicht zu den Informatiktagen in Nordrhein-Westfalen (dieses Bundesland wird hier, wie national vereinbart, mit **NW** abgekürzt) findet sich unter: www.nw.schule.de/gi/informatiktag/index.html

Informatiktag 2008

Der kommende, siebte Informatiktag findet 2008 – wie gewohnt – am letzten Montag vor den Osterferien (= 10. März 2008) statt. »Austragungsort« wird die Universität Münster sein; »Austragender« ist der Arbeitsbereich Didaktik der Informatik. Als Schwerpunktthema wurde »Informatik in der Sekundarstufe I« gewählt.

L^AT_EX – Teil 22: PSfrag

In einer Reihe von Artikeln in der **If Fase** werden nützliche Elemente von **L^AT_EX** vorgestellt, die erprobt sind und bei der Arbeit der Informatiklehrerin eingesetzt werden.

(von Thomas Arbeiter, Ludger Humbert) In den bisher vorgelegten zweiundzwanzig Teilen der Artikelserie – Ausgaben 0 ... 21: humbert.in.hagen.de/iffase/Archiv – finden Sie Hinweise und Anmerkungen zu den Themen: Installation, grundlegende Arbeitsweisen, Quellen zu Dokumentationen, Arbeit mit KOMA-Script, PSTricks, Erstellung von Arbeitsblättern, Struktogrammen, Automatengraphen, Elemente von UML, Barcodes, Formularerstellung, Zitieren, Abbildungen, ER Diagramme (mehrfach), Fragen der [Mikro-]Typografie, Setzen von Briefen Graphiken mit **L^AT_EX** erstellen, sowie Einbinden von Lizenzbedingungen in **L^AT_EX**-Quelltexte sowie in PDF-Dokumente. Mit der hier vorliegenden Ausgabe 22 wird thematisiert, wie dafür Sorge getragen werden kann, dass eingebundene Vektorgrafiken innerhalb der Grafik so beschriftet werden, dass die in **L^AT_EX** übliche, hervorragende Qualität auch in den Grafiken einen guten Textsatz erlaubt.

Besonders hervorzuheben ist, dass der Beitrag wesentlich von einem Referendar gestaltet wurde, dem die Vorstellung dieses Themas ein Anliegen ist.

PSfrag – Einsatzgebiete

Mit **L^AT_EX** lassen sich Dokumente erzeugen, die optisch hervorragend aussehen und professionell wirken. Dieser Eindruck kann jedoch schnell zunichte gemacht werden, wenn »LaTeX-fremde« Elemente im Dokument angezeigt werden sollen oder müssen. Bilder, Zeichnungen und Diagramme sind nur einige Beispiele, die zu Problemen in der Darstellung führen können. Dabei sind im Allgemeinen weniger die Grafiken selbst das Problem, sondern Beschriftungen innerhalb der Zeichnungen, die sich vom »normalen« Text des Dokumentes mitunter stark unterscheiden. Stammen die Zeichnungen dann noch aus verschiedenen Quellen oder wird die Größe geändert, so ergibt sich ein unansehnliches Durcheinander von Schrifttypen und -größen.

Abhilfe kann hier das **L^AT_EX**-Paket **PSfrag** schaffen. Es ersetzt Textelemente in EPS-Grafiken durch beliebige **L^AT_EX**-Elemente, so dass für Beschriftungen innerhalb eingefügter Grafiken nicht nur die gewünschte Schrift zur Verfügung steht, sondern komplexe **L^AT_EX**-Konstrukte eingebettet werden können. Dabei ist die Handhabung dieses Paketes kinderleicht.

PSfrag – Voraussetzungen

Zunächst muss die einzufügende Zeichnung als Vektorgrafik im EPS-Format vorliegen. Außerdem müssen in der Grafik an der Stelle, an der später **L^AT_EX**-Elemente auftauchen sollen, Zeichenketten stehen. Wichtig ist dabei, dass das Programm, welches die Vektorgrafik erstellt hat, die Zeichenketten als Zeichen und nicht als Bitmap speichert. Nur dann kann **PSfrag** die Zeichenketten in der Grafik finden und geeignet ersetzen.

Neben den beschriebenen Anforderungen an die einzufügende Grafik muss gewährleistet sein, dass das Paket **PSfrag** in die vorhandene **L^AT_EX**-Distribution korrekt eingebunden ist. In weit verbreiteten Distributionen MikTeX, TeX live und TeX ist **PSfrag** bereits integriert. Andernfalls sind die entsprechenden Dateien unter www.ctan.org/tex-archive/help/Catalogue/entries/psfrag.html zu finden.

PSfrag – Benutzung

Zunächst wird **PSfrag** durch `\usepackage{psfrag}` in der Präambel in das **L^AT_EX**-Dokument eingebunden. Mit `\includegraphics{grafik}` kann nun – wie gewohnt – jede beliebige Postscript-Grafiken eingefügt werden. Wird jedoch vor `\includegraphics` die Anweisung `\psfrag{irgendwas}{\sum\limits_{i=1}^n i}` geschrieben, so wird in der Abbildung die Zeichenkette »irgendwas« durch die **L^AT_EX**-Anweisung in der zweiten geschweiften Klammer ersetzt – im Beispiel also durch das Summenzeichen. **PSfrag** wird die entsprechenden Zeichenketten in sämtlichen nachfolgenden Abbildungen ersetzen, sofern diese in der gleichen geschlossenen Umgebung wie der `\psfrag`-Befehl – oder darin eingebetteten Umgebungen – stehen. So wird **PSfrag** bei folgendem Programmcode

```
\begin{figure}
\psfrag{tag1}{\textbf{Hier steht irgendetwas}}
\includegraphics{grafik1}
\includegraphics{grafik2}
\end{figure}
\includegraphics{grafik3}
```

in den Abbildungen `grafik1.eps` und `grafik2.eps` die Zeichenkette »tag1« durch den fettgedruckten Text »Hier steht irgendetwas« ersetzen; `grafik3.eps` erscheint hingegen – da außerhalb der `figure`-Umgebung – unverändert. Selbstverständlich können auch mehrere verschiedene Zeichenketten durch entsprechend viele `\psfrag`-Befehle in einer Abbildung ersetzt werden.

Weiterhin ist zu erwähnen, dass die `psfrag`-Anweisung parametrisiert werden kann. Die vollständige Befehlsyntax lautet

```
\psfrag{tag}[posn][psposn][scale][rot]{Ersetzungstext}
```

Dies ermöglicht eine genaue Positionierung des Ersetzungstextes. Es gibt außerdem noch eine Stern-Version von **PSfrag**, mit der Textelemente in einer Postscript-Grafik mehrfach ersetzt werden können.

PSfrag – Probleme mit pdf_latex

Dokumente, in denen mit dem Paket **PSfrag** gearbeitet wird, können prinzipiell nicht mit `pdflatex` übersetzt werden, da `pdflatex` keine Schnittstellen zu Postscript besitzt. In dieser Situation werden Nutzer, die solche Anforderungen formulieren, häufig aktiv und erstellen zumindest für ihren Anwendungsbereich Lösungen in Form von kleine Programmen, die die auftretenden Schwierigkeiten lösen. Hier wird auf zwei dieser Möglichkeit hingewiesen: Um PDF-Dokumente zu erzeugen, kann `ps2pdf` oder eine (etwas umständlichere) Möglichkeit genutzt werden, die unter www.tat.physik.uni-tuebingen.de/~vogel/fragmaster/main.html de.

`\tex{[]}{The \LaTeX\ package PSfrag}`

x4			
	x3		
	x2	x5x6	
	x1		

Die zu bearbeitende EPS-Datei: Vorher

```
Die Bearbeitung
\includegraphics[width=.47\textwidth]{psf-demo}
\hfill
\begin{psfrags}
\psfragscanon
\psfrag{x1}[br][ ]{\LaTeX} \psfrag{x2}[br][br]{\LaTeX}
\psfrag{x3}[br][t]{\LaTeX} \psfrag{x4}[br][Br]{\LaTeX}
\psfrag{x5}[Br][ r][1.15][45]{\Huge\LaTeX}
\psfrag{x6}[t][ l][1.15][45]{\Huge\LaTeX}
\includegraphics[width=.47\textwidth]{psf-demo}
\end{psfrags}
```

The **L^AT_EX** package PSfrag

L ^A T _E X			
L ^A T _E X			
L ^A T _E X	L ^A T _E X	L ^A T _E X	
L ^A T _E X			

Die bearbeitete EPS-Datei: Nachher

Fazit

Mit **PSfrag** ist es möglich, optisch ansprechende Grafiken in **L^AT_EX**-Dokumente einzufügen. Dabei ist die Handhabung kinderleicht und sollte auch **L^AT_EX**-Anfängern sofort gelingen. Lösbare Probleme treten auf, wenn man auf `pdflatex` angewiesen ist. In diesem Fall müssen die Änderungen außerhalb direkt auf den EPS-Dateien vorgenommen werden, die in PDF-Dateien umgewandelt werden müssen, damit sie von `pdflatex` eingebaut werden.

PSfrag – Verweise

Dokumentation und Downloadmöglichkeit: www.ctan.org/tex-archive/help/Catalogue/entries/psfrag.html
PSfrag mit `pdflatex`:
www.tat.physik.uni-tuebingen.de/~vogel/fragmaster/main.html

Wettbewerbe

Informatik Wettbewerbe

(von Dr. Ludger Humbert)

Biber Informatik

Vom 5. bis zum 9. November 2007 findet die **Biberwoche** statt. www.informatik-biber.de

In Biberwoche ist die Teilnahme am Wettbewerb jeweils zwischen 7:30 Uhr und 18 Uhr möglich. Innerhalb von 40 Minuten können die teilnehmenden Schülerinnen und Schüler 15 Aufgaben bearbeiten.

Voraussetzung: die Schule hat einen Koordinator/eine Koordinatorin, der/die die teilnehmenden Schülerinnen und Schüler anmeldet.



Logo des Biber-Wettbewerbs <http://www.informatik-biber.de/>

Bundeswettbewerb Informatik

Für die erste Rundes des laufenden 26. Bundeswettbewerbs Informatik ist am 12. November 2007 Einsendeschluss.

Einige Fragen zu den Aufgaben und Antworten finden sich auf den Webseiten des Bundeswettbewerbs Informatik www.bwinf.de

KurzNotiert



(von Dr. Ludger Humbert)

Vorratsdatenspeicherung stoppen!

Noch 0 Tage bis zur Einführung des neuen E-Reisepasses ...

Noch 8 Tage bis zur Abstimmung im Deutschen Bundestag ...

Noch 61 Tage bis zur totalen Protokollierung der Telekommunikation ...

Für Datenschützer brechen harte Zeiten an. Unterstützen wir sie in ihrem Bemühen, die letzten Reste der Freiheit zu verteidigen: Am Dienstag, 6. November finden bundesweite dezentrale Kundgebungen statt.

Weitere Hinweise: www.vorratsdatenspeicherung.de

www.FreiheitStattAngst.de
www.foebud.org



INFOS 2007 – Workshop: Roboter im Informatikunterricht (Pieper)

R. Greb (Privates Gymnasium
Marienstatt) und M. Hufnagel
(Gymnasium Delbrück)

(von Johannes Pieper) Dieser Workshop im Rahmen der Infos 2007 gab Einblick in verschiedene Möglichkeiten, Roboter im Informatikunterricht einzusetzen. Neben einem Theorieteil gab es ausreichend Zeit, um selber den Einsatz von Robotern in der Praxis zu testen. Dabei standen nicht nur die bekannten neuen LEGO-NXT bereit, sondern auch die älteren Modelle aus dem selben Haus. Von Fischertechnik standen neben Modellen von älteren und neueren Baureihen auch verschiedene passende Programmieroberflächen zur Verfügung. Bei großer Auswahl wurde dafür gesorgt, dass man sich mit einem Modell näher beschäftigen konnte und zusätzlich Information und Beispiele der anderen Modelle erhalten konnte.

Einführung

Der erste Teil des Workshops war eine Einführung in das Thema. Dabei wurde aufgezählt, aus welchen Bestandteilen, wie z. B. Aktoren und Sensoren, sich generell Roboter zusammensetzen. Die Roboter, die man in der Schule bzw. im privaten Umfeld mit Kindern einsetzen kann, sind meist aus Produkten der Industrie entstanden. Meist basiert daher auch die Programmierung der Roboter auf solchen Produkten. Neben den Produktreihen der beiden Firmen, von denen Modelle vorhanden waren, wurden auch noch weitere genannt, die mögliche Modelle anbieten.

Praxisphase

In kleinen Gruppen mit zwei bis drei Personen hatten dann alle Teilnehmer die Chance, ein Robotermodeil selber zu testen und es nach ihren Wünschen umzuprogrammieren. Damit auch alle von jedem Modell etwas hatten, wurde es so geregelt, dass mindestens einer aus der Gruppe bei dem Roboter geblieben ist, um Interessierten dieses zu erklären, während der Rest der Gruppe sich umgesehen hat. Da es von Seiten einiger Teilnehmer Interesse gab, wurde auch während dieser Phase versucht, einen der LEGO-Modelle mit LegoKara anzusteuern.

Konkrete Tipps für den Unterricht

Zum Abschluss wurden noch ein paar Tipps zum Unterricht gegeben. Dabei wurde unter anderem erwähnt, dass sich Gruppen aus höchstens drei Schülern am besten eignen. Zusätzlich sollte bei den Modellen mit möglichst wenig Sensoren und Aktoren begonnen werden, die dann später erweitert werden. Da die Bauphase auch eine längere Zeit einnimmt, wurde vorgeschlagen, dieses nur mit Teilen der Gruppe außerhalb des Unterrichtes durchzuführen. Dabei hätte es sich gezeigt, dass es zu empfehlen ist, Bauanleitungen zu folgen. Selbst erfahrene LEGO-Bauer wären bisher noch nicht zu geeigneten Modellen gekommen.

Fazit

Es war ein Workshop, der offen war für Anregungen und Austausch von Erfahrungen. Ob man Roboter in seinem Unterricht einsetzen will, bleibt danach weiterhin jedem selber überlassen. Es ist eine Möglichkeit, Programme auf eine andere Weise erfahrbar zu machen, als nur auf dem Bildschirm. Die Nachteile des Einsatzes sind aber auch klar geworden: Die Anschaffung von Robotern in Klassenstärke ist mit einem hohen finanziellen Aufwand verbunden, der getätigt werden muss. Ein weiterer Nachteil liegt darin, dass die Möglichkeiten der Robotermodelle sehr stark eingeschränkt sind. Dieses zeigte sich auch im Workshop, dass man meist nur eine einzige Tätigkeit machen konnte. Eine Erweiterung hätte einen entsprechenden Umbau mit sich gebracht.

INFOS 2007 – Vortrag »Das Projekt Infopraktiker – Ein neuer Informatikberuf für schulschwache Jugendliche« in der Schweiz (Wolf)

Dr. C. Thomann

(von Christian Wolf) Herr Dr. Christoph Thomann, Abteilungsleiter der Technischen Berufsschule Zürich, erarbeitet zusammen in einer 11-köpfigen Arbeitsgruppe als deren Vorsitzender die Einführung eines neuen Informatikberufs namens »Infopraktiker« in der Schweiz. Der neue Ausbildungsberuf richtet sich in erster Linie an praktisch begabte und kontaktfreudige Jugendliche, die jedoch Schwächen in den schulischen Fächern haben.

Im Rahmen des Vortrages stellte Herr Dr. Thomann zunächst das aktuelle Ausbildungskonzept der modularisierten Informatiklehre in der Schweiz vor (weitere Hinweise hierzu sind in dem Bericht von Thomas Hammersen (humbert.in.hagen.de/iffase/Artikel/Informatiklehre_Schweiz_2005-10-03.html) in der If Fase Ausgabe Nr. 5 nachzulesen). Angesprochen wurden häufige Probleme bei der Informatiklehre im Ausbildungsbetrieb, dass z. B. die übertragenen Aufgaben und Arbeiten oftmals zu einfach und monoton sind. Ein weiterer Anstoß für die Einführung des Berufs Infopraktiker besteht darin, ein Lehrstellenangebot für praktisch begabte Jugendliche mit Lernschwächen aufzubauen. Nach Meinung von Herrn Dr. Thomann leisten diese Jugendlichen oft gute praktische Arbeit und führen auch Tätigkeiten gerne aus, die sich über einen längeren Zeitraum wiederholen. Als Grund wurde hierfür genannt, dass die Jugendlichen sich in ihrer Tätigkeit bestärkt und sinnvoll gebraucht fühlen und somit in dem Beruf aufgehen (oder wie die Schweizer sagen: »den Knopf auf tun«).

Mögliche Tätigkeitsfelder:

- Routinemäßige HW- und SW-Installationen
- Verkauf von IT-Geräten mit Beratung
- First Level Support für Benutzer
- Wartung und einfache Fehlerbehebung: PC und Peripherie

- Praktisches Handeln mit wenig Theorie
 - Soziale Kompetenzen vor Mathematik-Kenntnissen
- Mögliche Einsatzorte:
- Schwerpunkt nicht bei Informatikbetrieben
 - Betriebe mit größerer IT-Infrastruktur und Usern
 - Handel und Verkauf von IT-Geräte
 - Betriebe für den PC-Support
- Die Ausbildungsdauer beträgt zwei Jahre und schließt mit dem Eidgenössischen Berufsattest (EBA) ab. Die duale Ausbildung besteht aus einem Schultag pro Woche zuzüglich der überbetrieblichen Kurse.

Um einen neuen Beruf in der Schweiz einzuführen, müssen bestimmte Phasen durchlaufen werden. Das Projekt Infopraktiker hat bereits die notwendigen Analysephasen abgeschlossen (Analyse der Arbeitssituation und Festlegung des Berufsprofils) und befindet sich nun in der mittleren so genannten Vor-Ticketphase. Eine Umfrage bei 200 teilnehmenden Betrieben ergab, dass 24,8% der Betriebe sich vorstellen können, Infopraktiker/innen mit abgeschlossener Ausbildung in Ihrem Betrieb einzusetzen, 29,1% wollen es noch prüfen. Bei der Fragestellung, ob der Betrieb auch selbst einen Infopraktiker-Ausbildungsplatz anbieten würde, ist das Ergebnis ähnlich: 19,3% bejahten die Frage und 29,3% wollen es noch prüfen.

Seit dem Sommer 2007 gibt es insgesamt zwei Pilotklassen: eine Klasse an der Gewerblich-Industriellen Berufsschule Bern (GIBB) und eine Klasse an der Wirtschaftsinformatikschule Schweiz (WISS) in Zürich. Bei den beiden Pilotklassen wurden vorsätzlich eher schulschwache Schüler ausgewählt. Die Schüler empfinden die Teilnahme an dem Pilotprojekt selbst als »Glücksfall« und waren schon vor der Ausbildung, wie die Schweizer sagen, »angefressen« von der Informatik.

Auf Nachfrage in einem kurzen persönlichen Gespräch, welche kritischen Stimmen es denn von anderen Personen zu dem Infopraktiker Projekt gibt, nannte Herr Dr. Thomann das Arbeitsplatzangebot sowie die Meinung, dass intellektuelles Wissen ein Muss für den Informatiker sei.

Aktuell hat das Schweizer Bundesamt für Berufsbildung und Technologie (BBT) nun zugestimmt, dass die Pilotklassen in Zürich und Bern weitergeführt und im Anschluss auch neue Klassen wiederum gebildet werden können. Bzgl. der anderen Schweizer Kantone erwartet der Bund eine Evaluation der beiden Pilotklassen.

Für detaillierte Angaben zu den einzelnen Projekten stehen folgende Ansprechpartner zur Verfügung:

- Berufsbild Infopraktiker/in: Dr. Christoph Thomann (christoph.thomann@tbz.ch)

- Pilotversuch Kanton Zürich: Candide Waldispühl (cwaldispuehl@wiss.ch)

- Pilotversuch Kanton Bern: Matthias Moser (matthias.moser@i-bern.ch)

Endlich einmal ein Konzept, welches sich an den Menschen orientiert. Ich freue mich über die Energie, die die Arbeitsgruppe in dieses Projekt investiert und hoffe, dass das Schweizer BBT das zukunftssträchtige Projekt unterstützt und die lange Einführungsphase eventuell verkürzt. Denkbar wäre ein ähnlicher Ansatz auch in anderen Ländern wie z. B. Deutschland. Ich vertrete persönlich auch die Meinung, dass schulschwache Jugendliche durchaus ein großes Talent und Interesse für die Informatik mitbringen können. Gerade im Hinblick auf die aktuelle Diskussion um die Blue Card könnten solche Maßnahmen national sehr fruchtbar sein. Ein persönliches Dankeschön geht an Herrn Dr. Thomann für die Versorgung mit aktuellen Materialien und Hinweisen.

INFOS 2007 – Workshop »KOMA: Modellierung von Klassen, Objekten, Methoden und Attributen in der Sekundarstufe I« (Oprisch, Pieper)

H. Fischer, T. Knapp, S. Friedrich, H. Neupert, K. Thuß

(von Anca Oprisch und Johannes Pieper) Der Workshop sollte den Tagungsbeitrag »KOMA – Das Konzept einer Fortbildung« praktisch demonstrieren. Dabei wurde versucht, den Ablauf der eintägigen Fortbildung auf die Zeit des Workshops zu komprimieren. In diesem Konzept geht es um die einheitliche Darstellung von »Objekten« und »Klassen«.

Mögliche Darstellungsarten sind:

- Umgangssprache
- Wissensnetze (Die Autoren bezeichnen diese mit »Mindmaps«. Weil diese Bezeichnung copyright-geschützt ist, wird im Folgenden die Bezeichnung Wissensnetze benutzt.)
- UML-Standards (?)
- Punktnotation

Die »Regeln für die Darstellung« finden sich unter www.sn.schule.de/~fischer/gw_neu/download/darstellung_k_o_a_aw.pdf Sie wurden den Teilnehmern zur Verfügung gestellt.

Darstellung mit Wissensnetzen

Hier wurden »Objekte« und »Klassen« sehr ähnlich dargestellt. Name der »Objekte« und »Klassen« werden in gerundeten Rechtecken eingetragen, von denen die Attribute an schwarzen verengenden Zweigen abgehen. An diesen Zweigen sind bei Klassen mögliche Attribute angeben. Sowohl bei den Objekten als auch bei den Klassen enthalten die Zweige auch die Methoden, mit denen das entsprechende Attribut geändert werden kann. Der einzige gut sichtbare Unterschied zwischen Klassen und Objekten in dieser Darstellung ist darin zu finden, dass

Klassen zusätzliche Methoden an roten Zweigen haben können, die unabhängig von Attributen sind.

Darstellung mit UML

Eine an UML angelehnte Darstellung wurde präsentiert. Diese wird nach Regeln aufgebaut. Bei dieser Darstellung ist die Unterscheidung in Klassen und Objekte etwas besser zu sehen als in den Wissensnetzen. Dieses liegt in der Schreibweise der Namen begründet. Objekte führen den Namen der Klasse mit.

Didaktische Umsetzung

Begonnen hat der Workshop mit einem kurzen Vortrag. An dessen Ende wurden die Darstellungen von Objekten und Klassen besprochen und erklärt. Anhand eines Aufgabenbeispiels aus dem konkreten Unterricht begann die Praxisphase. Dabei ging es um ein Beispiel aus einer Tabellenkalkulation, in dem eine falsche Darstellungsform gewählt wurde. Durch die Veränderung von Attributwerten bestimmter Objekte, sollen die Schüler diese Fehler beheben.

Das zweite Beispiel wurde durch die Teilnehmer selber durchgeführt. In drei verschiedenen Darstellungsformen sollte ein Rezept bearbeitet werden. Diese Formen waren: eine Bildschirmpräsentation aus Bildern, einen Text sowie eine Tabellenkalkulation. Jede Gruppe hatte die Aufgabe, das Rezept entsprechend umzusetzen und anschließend drei Objekte mit ihren Attributwerten heraus zu schreiben. Bei der anschließenden Präsentation zeigte sich deutlich, dass selbst vielen

Teilnehmern nicht der Unterschied zwischen Objekten und Klassen bekannt war, da mehrere Gruppen Klassen präsentiert haben. Nach Angabe der entsprechenden Gruppen hatten sie wissentlich nach Objekten gesucht.

Im letzten Teil des Workshops sollte für das bekannte Spiel Minesweeper mögliche Klassen heraus gesucht werden. Dieses wurde in der großen Runde durchgeführt. Da die Wahl von Objekten nicht eindeutig war, gab es mehrere richtige Lösungen für dieses Problem. Hier zeigte sich bei einigen Teilnehmern, dass sie das Konzept nicht richtig verstanden haben und durch die fehlende Ausführlichkeit mehr verwirrt wurden, als dass sie neue Erkenntnisse gewonnen haben.

Fazit

Man hat dem Workshop den Zeitdruck angemerkt. Bei dem beginnenden Vortrag wurden die Folien zu schnell umgeblättert. Es wurden viele Dinge sehr kurz abgehakt und es blieb kaum Zeit für eine ausführliche Diskussion. Dadurch bedingt gab es zahlreiche Konfusionen.

Für den Unterricht ist das Konzept an einigen Stellen noch zu überdenken. Gerade die nicht klare Trennung von Objekten und Klassen kann sehr schnell dazu führen, dass die Begriffe von Schülern vertauscht werden. Es würde sich daher anbieten, die Darstellungen zu überarbeiten, so dass Objekte und Klassen graphisch deutlich voneinander zu unterscheiden sind. Auch eine große zeitliche Trennung der Einführung der beiden Begriffe könnte hier Abhilfe schaffen.

INFOS 2007 – Workshop: »Projekte im Informatikunterricht – Modellieren und Implementieren von diskreten Simulationsmodellen« (Arbeiter)

Dr. H. Herper (Uni
Magdeburg)

(von Thomas Arbeiter) Diskrete Simulationsmodelle im Informatikunterricht zu behandeln, scheint auf den ersten Blick nicht ganz einfach zu sein. Dass es jedoch möglich ist, zeigte Herr Dr. Herper in seinem Workshop. Bereits beim Betreten des Raumes fiel mein Blick sofort auf ein dekoratives Lego-Modell, welches einen Straßenschnitt mit einer Tankstelle nebst Zufahrten erkennen ließ und in mir den Eindruck erweckte, dass der Workshop interessant zu werden verspricht.

Zu Beginn stellte Herr Dr. Herper klar, dass die Beschäftigung mit Simulationsmodellen sehr gut zu den Richtlinien des Informatikunterrichtes in der Sekundarstufe II passt. Den Teilnehmern wurde eine Situation vorgestellt, die grundlegend für diesem Workshop die Modellbildung, Prozessbeschreibung und Erarbeitung von Fragestellungen, sowie deren Analyse erlaubt. Wie man sich aufgrund des bereits erwähnten Tankstellenmodells denken kann, sollte es um die Simulation einer Tankstelle gehen.

Auch wenn dieses konkrete Thema nur begrenzt zur Alltagswelt von Schülern gehört, dürfte die Untersuchung dieses Problems dennoch interessant und spannend sein. Im Workshop wurden nun alle Phasen des Modellierens durchlaufen: Es standen mögliche Fragestellungen im Fokus, welche mithilfe einer Simulation beantwortet werden könnten. Ferner wurden Systemgrenzen festgelegt; außerdem Objekte klassifiziert, die in das Modell einfließen und diskutiert, wie detailliert die Objekte beschrieben werden müssen. Die Beschreibung der Objekte erfolgte dabei in UML-Notation. All dies kann im Unterricht sehr gut zusammen mit Schülerinnen und Schülern erarbeitet werden.

Schließlich wurde die Simulation trocken – sprich ohne Rechner – durchgeführt. Zwei zehneckige Würfel erzeugten Zufallszahlen, die darüber entschieden, wann ein Auto zur Tankstelle einbiegt, wie lange der Tankvorgang dauert und ob die Person in Eile schnell bezahlt oder noch einen größeren Einkaufsbummel im Tankstellenshop vornimmt. In einer Tabelle wurden dabei die Zeitpunkte, zu denen eine neue Phase begann, eingetragen. Im Workshop machte sich an dieser Stelle Murphys Gesetz bemerkbar, welches für ungeeignete Zufallszahlen sorgte und den (uninteressanten) Fall erzeugte, dass ein neues Auto immer erst dann an die Tankstelle fuhr, wenn das vorherige die Tankstelle bereits verlassen hatte. Ein wenig »menschliche Nachhilfe« sorgte dann aber doch noch für eine betriebsame Tankstelle :-)

Die – vermutlich für Schüler interessanteste – Phase begann dann mit der Visualisierung des erwählten Ablaufs. Dafür wurde das Animationswerkzeug »Proof Animation« benutzt, welches leicht zu bedienen ist und recht ansprechende Animationen erzeugt.

Auch wenn das Würfeln gut geeignet ist, um Schülern die Abläufe des Modells zu verdeutlichen, ist eine Übertragung in ein Computermodell wünschenswert. Dies geschah im Workshop unter Verwendung der Simulationssprache WinGPSS, mit deren Hilfe Animationsablaufprotokolle erzeugt werden, die Proof Animation dann darstellt.

Fazit: Der Workshop war sehr interessant. Die vorgestellte Unterrichtsreihe halte ich für durchaus realisierbar und interessant für Schülerinnen und Schüler. Probleme sehe ich höchstens in der Simulationssprache WinGPSS: ich hatte den Eindruck, dass man dafür Einarbeitungszeit benötigt. Da ich mit WinGPSS keine eigenen Erfahrungen habe und im Workshop nur ganz kurz darauf eingegangen wurde, kann ich die Komplexität dieses Programms aber nicht genau beurteilen.

INFOS 2007 – Workshop »Einsatz von Flash im Informatikunterricht der Mittelstufe« (Schultz) M. Pesch (Sankt Augustin)

(von *Torsten Schultz*) Der Titel dieses Workshops versprach, dass aufgezeigt wird, wie sich informatische Inhalte mit Hilfe von Flash im Unterricht der Sek I, also in NRW im Wesentlichen in der Differenzierung der neunten und zehnten Klasse, vermitteln lassen. Flash assoziiert man mit bunten Internetseiten, was eine schülerorientierte und motivierende Unterrichtsreihe verspricht.

Im Workshop wurde zuerst als Werkzeug das Flash-Produkt des hier führenden Anbieters vorgestellt. Hier gibt es eine 30-Tage-Testversion, so dass im Workshop gearbeitet werden konnte. Gleichzeitig wird an dieser Stelle schon deutlich, dass an einen Einsatz in der Schule kaum zu denken ist, eine Einzellizenz des Produktes kostet etwa so viel, wie ein heute handelsüblicher einfacher Laptop. Das Arbeiten mit Testversionen ist unpraktisch und vermutlich auch verboten. Alternative, u. U. freie Produkte wurden weder vorgestellt noch erwähnt. Das vorgestellte Werkzeug ist ein sogenanntes professionelles Produkt, es bietet eine Überfülle von Funktionen und hat eine entsprechend unübersichtliche Benutzungsoberfläche. Für Schule und Schüler muss es als überfrachtet bezeichnet werden.

Informatische Bildung?

Inhaltlich wurde den Teilnehmern im Verlauf des Workshops vermittelt, wie mit dem eingeführten Werkzeug einfache Flash-Filme erstellt werden können. Es wurde aber kaum auf allgemeinbildungsrelevante informatische Inhalte hingewiesen, die im Zuge der Reihe vermittelt werden können. Solche Inhalte lassen sich sicherlich ausmachen, ob hier aber der optimale Weg aufgezeigt wurde, diese zu vermitteln, bleibt fraglich. Parallel wurden einige Schülerlösungen präsentiert. Diese konnten aber auch nicht deutlich machen, welche informatischen Inhalte im Unterricht zentral waren. Sie zeigten vielmehr, dass das Werkzeug missbraucht werden kann, um Szenen darzustellen, die in der Schule nicht zeigenswert sind. Im weiteren Verlauf wurde in die Programmierung von Flash-Filmen mit Hilfe von ActionScript eingeführt. Dabei handelt es sich um eine Java-Script ähnliche Sprache. Hier wäre sicherlich eine Möglichkeit gegeben, Inhalte wie Sequenz, Verzweigung und Schleife einzuführen. Diese wurden aber im Workshop nicht als zentral und bildungsrelevant herausgestellt.

Fazit

Der Workshop muss somit in weiten Teilen als Produktschulung für das angesprochene Flash-Produkt bezeichnet werden. Dies kann so nicht in einem anspruchsvollen Informatikunterricht stattfinden. Möglicherweise ist der Bereich Flash geeignet, informatische Inhalte zu transportieren, dazu müssten aber alternative Konzepte entwickelt werden. Auch müsste es für den Einsatz in der Schule ein Werkzeug geben, dass für Schulen finanzierbar ist und das nur die wichtigen Funktionen zur Verfügung stellt, um Schüler nicht mit unwichtigen Details zu überfordern.

Grundsätze und Standards für die Informatik in der Schule

Bildungsstandards Informatik
Entwurfsvfassung für
Empfehlungen der Gesellschaft für Informatik e. V.
(Stand: August 2007)

Arbeitskreis »Bildungsstandards«
des Fachschulusses »Informatische Bildung in Schulen« (FA IBS)
und der Fachgruppe »Didaktik der Informatik« (FG DDI)

INFOS 2007 – Workshop »Objektorientierte informatische Grundbildung« (Löffler und Dr. Linke)

G. Kubitz (Hannah-Arendt-Gymnasium Lengerich)

(von *Susanne Löffler und Dr. Thomas Linke*) Die folgende zusammenfassende Darstellung stellt die subjektive Sicht eines Besuchers und einer Besucherin dar. Sie dient primär dazu, im Fachseminarzusammenhang eine Diskussion zu ermöglichen.

In diesem Workshop stellte G. Kubitz einen objektorientierten Ansatz zur informatischen Grundbildung in den Jahrgängen 6 und 7 im Rahmen der Ergänzungsstunden des achtjährigen Gymnasiums vor. Dabei wurden besonders die allgemeinbildenden Aspekte des Fachs Informatik betont, die in die Forderung mündeten, den Empfehlungen der GI nachzukommen und Informatik als eigenständiges Pflichtfach in der Sekundarstufe I einzuführen, das von Fachlehrerinnen und Fachlehrern unterrichtet wird.

Begründung des Vorgehens

Zunächst hob G. Kubitz den Beitrag hervor, den das Fach Informatik zur Allgemeinbildung leisten kann. Ausgehend von der Frage »Was ist Informatik?« und der Antwort »Systematisches Lösen von Problemen« zeichnete er folgendes Vorgehen nach:

- Problem: Teilsystem der realen Welt
- geistiges Modell: Aus der Analyse des Problems und der Klassifizierung der vorkommenden Objekte entsteht das geistige Modell.
- informatisches Modell: Aus dem geistigen Modell wird mithilfe der objektorientierten Modellierung ein informatisches Modell erstellt.
- lauffähiges Softwaresystem: Die Umsetzung des geistigen Modells erfolgt durch die Codierung/Programmierung und mündet in ein lauffähiges Softwaresystem.

Dieses Vorgehen und die Einführung der informatischen Begriffe »Objekt« und »Klasse« tragen nach Ansicht von Kubitz zur Allgemeinbildung der Schülerinnen und Schüler bei, weil bereits Kinder mithilfe von Eigenschaften und Fähigkeiten Kategorien bilden und in Klassen denken (z.B. das Objekt »Bello« und die Klasse »Hund«). Daher ist die objektorientierte Analyse für die informatische Grundbildung geeignet. Die objektorientierte Programmierung betreffend kann in den Klassen 6 und 7 ohne wirklich zu programmieren eine Grundvorstellungen über die Arbeitsweise von Softwaresystemen entwickelt werden.

Das obige Vorgehen beinhaltet zudem eine ganze Reihe allgemeinbildender Aspekte, wie die sogenannten Softskills (z.B. Teamfähigkeit), die Modellierung mit Objekten, Klassen und deren Beziehung, die Organisation und Darstellung von Information, die Algorithmen und das fachübergreifende Denken.

Als positives Beispiel wählte Kubitz die Situation in Bayern, wo Informatik in den Jahrgangsstufen 5, 6 und 7 des Gymnasiums als Pflichtfach im Rahmen des Fächerverbundes »Natur und Technik« und an naturwissenschaftlich-technischen Gymnasien in den Jahrgangsstufen 9 und 10 zudem als eigenständiges Pflichtfach unterrichtet wird. Den Stellenwert des Faches Informatik in NRW stellte er drastisch durch den Vergleich der Zahlen der Abiturientinnen und Abiturienten in den naturwissenschaftlichen Fächern im Jahr 2007 dar. Das Fach Informatik liegt hier weit abgeschlagen auf dem letzten Platz.

Unterrichtsgänge

Im Folgenden wurden Unterrichtsgänge beschrieben, die an der Schule von Herrn Kubitz im Rahmen der Ergänzungsstunden des achtjährigen Gymnasiums in den Jahrgangsstufe 6 und 7 einstündig durchgeführt werden und mit denen gute Erfahrungen gemacht wurden. Ausführlich behandelte Kubitz den Einstieg in den Unterricht zur informatischen Grundbildung anhand von Objekten in Grafiken.

Unterrichtsgang 1

Objekte in Grafiken

Der Zugang erfolgt spielerisch über die Verwendung von Grafikprogrammen. Zu einer vorgegebenen einfachen Zeichnung (Haus, mit je einem Baum halb davor und halb dahinter, bestehend aus Rechtecken und Kreisen) wird folgende Aufgabe gestellt:

- Abzeichnen mit Paint (Pixelgrafik),
- Abzeichnen mit OpenOffice.org DRAW (Vektorgrafik),
- Ändern der Bauvorschrift z. B. für die Fenster
- Mit welchem Programm war die Aufgabe leichter? Warum?

Anhand dieses Beispiels werden die Begriffe »Objekt« und »Attribut« eingeführt. Da die Attributwerte im Fall der Änderung der Bauvorschriften angepasst werden müssen, stellen die Schülerinnen und Schüler fest, dass die Objekte Dienste anbieten müssen, damit eine Veränderung der Werte vorgenommen werden kann. Weiter geht es in Gruppenarbeit z. B. mit der Aufgabe eine Dampflok mit genau fünf Objekten zu zeichnen und diese so genau wie möglich zu beschreiben, so dass eine andere Gruppe sie nachzeichnen kann. Die Ergebnisse werden ausgedruckt und verglichen, wodurch die Notwendigkeit der genauen Angabe von Objektattributen und deren Werten offensichtlich wird. Danach kann die Punktnotation als Beschreibungsmittel für die Objekteigenschaften eingeführt werden.

Es liegt nahe, neben der Beschreibung der Objekte durch Attribute auch eine Klassifizierung vorzunehmen, die im Klassenbegriff mündet. Zudem werden auch die Beziehungen von Objekten und Klassen thematisiert und Objekt- und Klassendiagramme erstellt.

Objekte in Texten

Objekte in Präsentationen

Unterrichtsgang 2

Baumstruktur (als allgemeines dem Menschen eigenes Werkzeug der Strukturierung)
Netzstrukturen

Unterrichtsgang 3

E-Mail und Algorithmen

Diskussion und Fazit

Die anschließende Diskussion sollte zum Erfahrungsaustausch über den Informatikunterricht in der Sekundarstufe I dienen. Es zeigte sich, dass nur an wenigen Schulen Informatikunterricht in den unteren Klassen stattfindet. Einige Teilnehmerinnen und Teilnehmer zeigten sich skeptisch, ob die Schülerinnen und Schüler, aber auch die Eltern eine Einführung der Textverarbeitung mithilfe des gewählten objektorientierten Ansatzes überhaupt akzeptieren und verstehen. Herr Kubitz konnte hier auf die erfolgreiche Umsetzung des Unterrichtsganges verweisen.

In dem Workshop wurde ein einfacher und einleuchtender Unterrichtsgang vorgestellt, der den objektorientierten Einstieg in das Fach Informatik in der Sekundarstufe I ermöglicht. Alle für die nachfolgende objektorientierte Programmierung wichtigen Begriffe wurden auf verständlichem Niveau besprochen. Hier ist besonders die objektorientierte Sicht auf Standardanwendungen hervorzuheben, weil sie ein anwendungsunabhängiges Verständnis wichtiger Datenverarbeitungsaufgaben gestattet.

Hilfreich war auch der Hinweis von G. Kubitz, dass man Kolleginnen und Kollegen leichter überzeugen kann, dass eine informatische Grundbildung in der Sekundarstufe I im Rahmen der Ergänzungsstunden stattfinden sollte, wenn man die Vorteile für andere Fächer hervorhebt. Wenn die Einführung der Textverarbeitung im Informatikunterricht erfolgt, kann z. B. die Deutschlehrerin/der Deutschlehrer auf diese Fähigkeiten zurückgreifen.

Der Forderung von Kubitz, Informatik als Pflichtfach in der Sekundarstufe I einzuführen, können wir nur zustimmen. Der Workshop zeigte uns eine Möglichkeit auf, Informatikunterricht in der aktuellen Situation auch in der Sekundarstufe I umzusetzen.

INFOS 2007 – Workshop »Internetworking und Informatiksystemverständnis« (Löffler)

W. Kempf (Gymnasium auf der Morgenröthe, Siegen), H. Koch (Fürst-Johann-Moritz-Gymnasium, Siegen), S. Freischlad, P. Stechert, S. Schubert (alle Universität Siegen)

(von *Susanne Löffler*) Die folgende zusammenfassende Darstellung stellt die subjektive Sicht einer Besucherin dar. Sie dient primär dazu, im Fachseminarzusammenhang eine Diskussion zu ermöglichen.

Der Workshop »Internetworking und Informatiksystemverständnis« befasste sich zum einen mit einem didaktischen System zum Thema Internetworking und zum anderen mit dem Ansatz des Verstehens von Informatiksystemen nach Stechert/Schubert, wobei die praktische Umsetzung und Erprobung durch Lehramtsstudierende an zwei Siegener Gymnasien im Vordergrund standen. Die Universität Siegen kooperierte zu diesem Zweck mit dem Gymnasium auf der Morgenröthe (vertreten durch den Schulleiter und Informatiklehrer W. Kempf) und dem Fürst-Johann-Moritz-Gymnasium (vertreten durch den Informatiklehrer H. Koch). Eine Diskussion über die Möglichkeiten der Verzahnung von universitärer Forschung und praktischer Umsetzung an der Schule sollte den Workshop abschließen.

Zu Beginn berichteten Herr Kempf und Herr Koch ausführlich über die Rahmenbedingungen an ihren Schulen und die Erfahrungen, die im Verlauf des Projekts gemacht wurden. Nach den Erfahrungsberichten der Informatiklehrer erfolgte der Übergang zur Diskussion, die jedoch eher stockend verlief. Einige Teilnehmerinnen und Teilnehmer des Workshops fragten explizit nach den theoretischen Grundlagen und der Konzeption des Projekts. Sie wurden auf die Vorträge verwiesen, die P. Stechert und S. Freischlad am Vormittag gehalten hatten. Eine Einführung in die theoretischen Grundlagen der Gesamtkonzepte, die den Ausgangspunkt für die Unterrichtsentwürfe bildeten, die in den beiden Gymnasien umgesetzt wurden, und eine nähere Erläuterung des Vorgehens im Rahmen des Projekts erfolgten nur in Ansätzen.

Fazit

Ausgehend von der Zusammenfassung zu dem Workshop hatte ich erwartet, dass neben den Erfahrungsberichten auch die Theorie und die Gesamtkonzeption der Projekte näher beleuchtet werden. Zudem ging aus der Zusammenfassung nicht hervor, dass eine starke Kopplung zwischen den Vorträgen am Vormittag und dem Workshop bestand. Da mir die notwendigen grundlegenden Kenntnisse über die didaktischen Ansätze und die Konzeption der Projekte fehlten, konnte ich die Erfahrungsberichte nicht in die Gesamtkonzeption einbinden und mich an der Diskussion nicht sinnvoll beteiligen.

Damit ein Workshop sowohl für die Teilnehmerinnen und Teilnehmer als auch für die Veranstalterinnen und Veranstalter zu einem fruchtbaren Ergebnis führt, sollte deutlich sein, dass der Besuch bestimmter Vorträge vorausgesetzt wird. Denn nur dann ist es möglich, dass sich alle in den Workshop einbringen können und davon profitieren. Außerdem wäre es sinnvoll, zu Beginn eines Workshops darzulegen, welche Themen behandelt werden sollen und dass der Workshop auf Vorträgen aufbaut, sodass sich die Teilnehmerinnen und Teilnehmer noch entscheiden können, einen anderen Workshop oder Vortrag zu besuchen, falls sie falsche Erwartungen haben oder die Vorträge nicht besucht haben.

Fachleitungen NW fordern aktive Mitgestaltung der Lehrplanentwicklung

(von *Dr. Ludger Humbert*)

Anlässlich des landesweiten Treffens aller Informatikfachleitungen für allgemein bildenden Schulen berieten nach Ende der offiziellen Tagung die Anwesenden über aktuelle Entwicklungen bezogen auf die Weiterentwicklung der Richtlinien und Lehrpläne für die gymnasiale Oberstufe.

(von *Dr. Ludger Humbert*) Als Ergebnis dieser Beratungen wurde die Vorsitzende Bärbel Kibben von uns beauftragt, eine offizielle Eingabe vorzubereiten, uns zukommen zu lassen und diese nach Bestätigung durch alle Fachleitungen Gymnasium/Gesamtschule an das Bildungsministerium NW weiterzugeben.

Inhalt des Schreibens an das Ministerium

Vom Sprecher der GI-Fachgruppe »Informatische Bildung in NRW«, Herrn David Tapaße, haben wir erfahren, dass der

neue Lehrplan Informatik SII für NRW von Herrn LRSD Klaus Dingemann, dem Fachdezernenten für Informatik bei der Bezirksregierung Münster, und bisher drei weiteren, uns nicht bekannten Personen entwickelt werden soll. Wir halten es für unabdingbar, dass der neue Lehrplan von weitaus mehr aktiven Informatiklehrerinnen und Informatiklehrern erstellt wird. Als Fachleiter sind wir in besonderer Weise an der Weiterentwicklung des Informatikunterrichts interessiert und beteiligt und fordern Sie daher auf, den Kreis der an der Entwicklung des Lehrplans Beteiligten mindestens um einen Vertreter aus unserer Gruppe zu erweitern. Außerdem möchten wir Sie bitten, uns im Verlauf der Lehrplanentwicklung die Gelegenheit zu Rückmeldungen zu Entwurfsfassungen zu geben und bieten an, unsere bereits bestehende Organisation der Fachleiter Informatik NRW für die konstruktive Mitarbeit am Lehrplan aktiv einzubringen.

Bärbel Kibben
Sprecherin der Fachleiter Informatik
für das Lehramt SII (GYMGE)

An das
Ministerium für
Schule und Weiterbildung
des Landes Nordrhein-Westfalen
Herr Burkhard (Referat 525)

26. Oktober 2007

Sehr geehrter Herr Burkhard,

Vom Sprecher der GI-Fachgruppe »Informatische Bildung in NRW«, Herrn David Tapaße, haben wir erfahren, dass der neue Lehrplan Informatik SII für NRW von Herrn LRSD Klaus Dingemann, dem Fachdezernenten für Informatik bei der Bezirksregierung Münster, und bisher drei weiteren, uns nicht bekannten Personen entwickelt werden soll.

Wir halten es für unabdingbar, dass der neue Lehrplan von weitaus mehr aktiven Informatiklehrerinnen und Informatiklehrern erstellt wird. Als Fachleiter sind wir in besonderer Weise an der Weiterentwicklung des Informatikunterrichts interessiert und beteiligt und fordern Sie daher auf, den Kreis der an der Entwicklung des Lehrplans Beteiligten mindestens um einen Vertreter aus unserer Gruppe zu erweitern. Außerdem möchten wir Sie bitten, uns im Verlauf der Lehrplanentwicklung die Gelegenheit zu Rückmeldungen zu Entwurfsfassungen zu geben und bieten an, unsere bereits bestehende Organisation der Fachleiter Informatik NRW für die konstruktive Mitarbeit am Lehrplan aktiv einzubringen.

Im Namen aller Fachleiter Informatik für das Lehramt SII (GYMGE) in NRW

Bärbel Kibben (Sprecherin)

Brief an das Ministerium im Faksimile

INFOS 2007 – Workshop »NP-schwere Probleme im Informatikunterricht« (Arbeiter)

M. König (Carl-Zeiss-Gymnasium Jena)

(von Thomas Arbeiter) Die theoretische Informatik ist ein Teilgebiet, welches im Informatikunterricht sehr stiefmütterlich behandelt wird. Dies ist nicht weiter verwunderlich, wenn man bedenkt, dass sich bereits ein großer Teil der Informatikstudenten mit diesem Gebiet sehr schwer tut. Automaten, Grammatiken, Komplexitätsanalysen, ... - all diese Begriffe fordern vom Lernenden hohe abstrakte Denkleistungen und führen bei SchülerInnen schnell zu Frust und Resignation.

Herr König wollte in diesem Workshop jedoch zeigen, dass es durchaus möglich - ja, erforderlich - ist, dass SchülerInnen im Unterricht mit Grundlagen der theoretischen Informatik, insbesondere mit Begriffen aus dem Umfeld der NP-schweren Probleme, konfrontiert werden. Nach einer kurzen Einführung, in welcher Herr König erklärte, dass die NP-schweren Probleme durchaus in die Richtlinien und Rahmenpläne aller Bundesländer für den Informatikunter-

richt passen, stellte er eine mögliche Unterrichtsreihe recht detailliert vor.

Das Programm »NP-Schule«

Mittelpunkt der Unterrichtsreihe ist ein selbstentwickeltes Programm mit dem Namen »NP-Schule«, welches SchülerInnen helfen soll, einige NP-schwere Probleme kennenzulernen und eigenständig zu erforschen. Dieses Programm wird auf der Internetseite e4.mirz.uni-jena.de/np vorgestellt und steht dort zum Download bereit. Es stellt vier Probleme

- Rucksackproblem
- Behälterproblem
- Partitionsproblem
- Erfüllbarkeitsproblem

vor. Die Beschreibungen der einzelnen Probleme in diesem Programm sind in meinen Augen jedoch an vielen Stellen verbesserungswürdig: Einige Textteile sind für SchülerInnen vermutlich schwer

verständlich; manche etwas verwirrend: So wird beispielsweise das Rucksackproblem anhand der Planung eines Nachmittages eingeführt, an dem man nur zwei Stunden Zeit hat und in dieser Zeit gerne Fernsehen und Computer spielen möchte; nebenbei aber auch noch lästige Hausaufgaben zu bearbeiten hat. Wenn die einzelnen Aktivitäten mit Prioritäten und Zeiterfordernissen versehen werden, erhält man eine Optimierungsaufgabe, die nun zu lösen ist. Diese Einführung mag zwar leicht verständlich sein; was das Ganze allerdings mit »Rucksäcken« zu tun haben soll, bleibt schleierhaft.

Neben den Problembeschreibungen haben die SchülerInnen immer die Möglichkeit, das Problem näher kennenzulernen, indem sie interaktiv eine gestellte Aufgabe lösen und dabei schnell merken, wie aufwändig eine Lösung wird, wenn man die Anzahl der verwendeten Objekte nur leicht erhöht. Mit diesem Programmteil dürften die SchülerInnen

einigen Spass haben.

Im Workshop konnten die Teilnehmer mit diesem Programm ein wenig »herumspielen«, wobei sich gerade der interessante Teil, in dem Schüler interaktiv Aufgaben lösen, als fehleranfällig erwies. Bei ungünstigen Eingaben tauchten Fehlermeldungen und Programmabbrüche auf, die aber sicherlich leicht zu beheben sind. Im Großen und Ganzen machte das Programm aber einen guten Eindruck und scheint für den Unterricht durchaus geeignet zu sein.

Reduzierbarkeit

Nachdem sich die Teilnehmer des Workshops ausgiebig mit der Software beschäftigt hatten, stellte Herr König einen weiteren Teil der geplanten Unterrichtsreihe vor: Es ging um polynomiale Reduzierbarkeit. Hierbei wurde bewiesen, dass das Rucksackproblem polynomial auf das Partitionsproblems reduzierbar

ist. Der Beweis liegt in Form eines Flash-Filmes vor, in welchem die einzelnen Schritte ausführlich dargestellt und mit geeigneten Grafiken veranschaulicht werden. Trotzdem bleibt der Beweis sehr umfangreich und dürfte aus meiner Sicht die meisten SchülerInnen überfordern.

Fazit

Die Bemühungen von Herrn König, Grundlagen der theoretischen Informatik im Unterricht zu behandeln, kann ich nur unterstützen. Die Schwierigkeit liegt jedoch darin, die äußerst schwierigen und komplexen Inhalte schülergerecht zu reduzieren und aufzubereiten. Das vorgestellte Programm »NP-Schule« stellt dabei einen Ansatz dar, der mir sehr gut gefällt. Einige Teile der vorgestellten Unterrichtsreihe halte ich jedoch für kaum umsetzbar, da sie in der vorgestellten Form einfach zu abstrakt und schwer nachvollziehbar sind.

INFOS 2007 – Workshop »Sprachen und Sprachkonzepte – eine Unterrichtseinheit für die Sekundarstufe I nach dem informationsorientierten didaktischen Ansatz« (Löffler)

T. Hempel (Richard-Wossidlo-Gymnasium Ribnitz-Damgarten)

(von Susanne Löffler) Die folgende zusammenfassende Darstellung stellt die subjektive Sicht einer Besucherin dar. Sie dient primär dazu, im Fachseminarzusammenhang eine Diskussion zu ermöglichen.

In diesem Workshop stellte T. Hempel eine Unterrichtseinheit für die Sekundarstufe I zum Thema »Sprachen und Sprachkonzepte« vor, die i. d. R. in der Jahrgangsstufe 10 durchgeführt wird und mindestens 30 Unterrichtsstunden umfasst. Die Unterrichtseinheit setzt sich aus den drei Sequenzen »Von der natürlichen zu den künstlichen Sprachen«, »Computer und Sprache« und »Problemlösen mit dem Computer« zusammen. Sie wurde von Prof. Breier (Universität Hamburg) entwickelt, der einen Vortrag zu diesem Thema auf der INFOS 2005 gehalten hat.

Zunächst bettete Herr Hempel die Unterrichtsreihe in die Bildungsstandards und den Rahmenplan für das Fach Informatik in Mecklenburg-Vorpommern ein, wo Informatik in den Jahrgangsstufen 5 bis 10 unterrichtet wird, und stellte dann die Unterrichtseinheit vor, die hier kurz dargestellt werden soll.

Von der natürlichen zu den künstlichen Sprachen

Im Rahmen der ersten Unterrichtssequenz informieren sich die Schülerinnen und Schüler zunächst über das Thema »Sprache«. Der Schwerpunkt liegt hierbei auf

einer Definition für Sprache, den Grundelementen von Sprache (Alphabet, Syntax, Semantik etc.) und den Gemeinsamkeiten und Unterschieden von verschiedenen Sprachen. Die Schülerinnen und Schüler halten ihre Ergebnisse in Form von Wissensnetzen fest. Im Anschluss erfolgt die Analyse und Festigung der Begriffe. Hempel stellte interessante Einstiege vor, die genutzt werden können, um die Schülerinnen und Schüler zur Auseinandersetzung mit dem Thema zu motivieren. Die Sprachen der Welt können z. B. auf der Internetseite www3.germanistik.uni-halle.de/prinz/index.htm erforscht werden, auf der in verschiedenen Sprachen ein Auszug aus dem Buch »Der kleine Prinz« vorgelesen wird. Einen humorvollen Einstieg in Dialekte und Slangs bzw. in Grammatiken bieten die Beiträge »Lerne Ostdeutsch« und »Imbissdeutsch« der Sendung »Extra 3« des NDR-Fernsehens. Mithilfe von PROLOG kann eine einfache Grammatik festgelegt werden, sodass man überprüfen kann, ob ein Satz gemäß der Grammatik gebildet wurde, und Sätze ausgegeben werden können. Danach erfolgt der Übergang zu den künstlichen Sprachen. Die Schülerinnen und Schüler erstellen arbeitsteilig Präsentationen z.B. zur Blindenschrift, zu Esperanto oder zur Gebärdensprache. Zum Abschluss der Sequenz erfolgt eine Zusammenfassung der Ergebnisse, die zum Grundschemata der Verarbeitung und Vermittlung von Information führt.

Computer und Sprache

Als Grundlagen bringen die Schülerinnen und Schüler aus den anderen Schuljahren das Wissen über die Codierung von Information, über Informationsträger und über das EVA-Prinzip mit. Daher steht in dieser Sequenz zunächst die Verarbeitung binärer Daten durch den Computer im Mittelpunkt. Zu Beginn wird der von Neumann-Rechner und der Aufbau des Prozessors mit Registern behandelt. Zur Analyse der Arbeitsweise eines Computers wird ein Minimalrechner verwendet, auf dem die Schülerinnen und Schüler Programme ablaufen lassen. Da die Maschinensprache, aber auch Assembler für den Menschen nur schwer verständlich sind, erleben die Schülerinnen und Schüler die höheren Programmiersprachen als klare Verbesserung. Mithilfe eines Stammbaums der Programmiersprachen wird schnell deutlich, dass eine programmiersprachenunabhängige Darstellung, das Struktogramm, notwendig ist, um Algorithmen zu beschreiben.

Problemlösen mit dem Computer

In der letzten Unterrichtssequenz erlernen die Schülerinnen und Schüler ein Problem mithilfe des Computers zu lösen, indem sie die Problemstellung analysieren, ein Modell in Form eines Struktogramms erstellen und dann zur Implementation

übergehen. Zunächst erhalten sie ein Java-Programm, das das Nimm-Spiel umsetzt, aber noch Fehler enthält. Die Schülerinnen und Schüler nutzen und analysieren das Programm, indem sie den Ablauf protokollieren. Dem in der Umgangssprache festgehaltene Programmablauf wird das Java-Programm gegenübergestellt. Zudem erhalten die Schülerinnen und Schüler ein Arbeitsblatt mit einem Algorithmen-Baukasten, der erläutert, wie ein Struktogramm in Java-Code übersetzt wird. Aufbauend auf dieser Grundlage verbessern sie das Programm, sodass es fehlerfrei abläuft. Im Anschluss daran können weitere Aufgaben von den Schülerinnen und Schülern bearbeitet und in Programme überführt werden.

Die Schülerinnen und Schüler sollen im Rahmen dieser Unterrichtssequenz auch erfahren, dass das imperative Problemlösen nicht immer der optimale Weg ist und dass es Alternativen gibt. Dazu wird das deklarative Problemlösen betrachtet, indem eine Aufgabenstellung durch die Verwendung von PROLOG gelöst wird.

Die Unterrichtsreihe wird durch eine Zusammenfassung abgeschlossen, in der hervorgehoben wird, dass die Schülerinnen und Schüler von einem Problem ausgegangen sind, das in natürlicher Sprache formuliert ist, und mithilfe des imperativen oder des deklarativen Wegs zu einem Programm auf dem Computer gelangen.

Fazit

Die Unterrichtsreihe holt die Schülerinnen und Schüler in der eigenen Lebenswelt beim täglichen Umgang mit der natürlichen Sprache ab und führt sie zu den Programmiersprachen. Zudem verdeutlicht sie den Weg von der Problemstellung über die Problemlösung zum Programm auf dem Computer. Durch die Betrachtung des imperativen und deklarativen Problemlösens erfahren die Schülerinnen und Schüler, dass es verschiedene Wege gibt. Welcher Weg geeigneter ist, hängt dabei vom Problem ab.

T. Hempel stellte in seinem Workshop eine Unterrichtsreihe vor, die den Schülerinnen und Schülern viel Raum zur Selbsttätigkeit und zum Entdecken bietet. Seine Präsentation umfasste konkrete praktische Hinweise und Vorschläge, die man gut in den eigenen Unterricht einbauen kann, wenn man ein ähnliches Thema behandelt. Der Workshop wurde ergänzt durch kleine Übungen, die auch Teil der Unterrichtsreihe sind. Die praktische Tätigkeit lockerte den Workshop auf und verdeutlichte die vorgestellten Ideen.

Zum Weiterlesen

Auf der Homepage von Tino Hempel unter www.tinohempel.de befinden sich die Unterlagen und die Präsentation zum Workshop auf der INFOS 2007 und viele weitere Materialien.

INFOS 2007 – Workshop »Informatikunterricht mit der visuellen Programmiersprache Puck« (Dr. Linke)

Dipl.-Inf. L. Kohl

(von Dr. Thomas Linke) In dem Workshop wurde die visuelle Programmiersprache Puck www.uni-jena.de/puck.html vorgestellt.

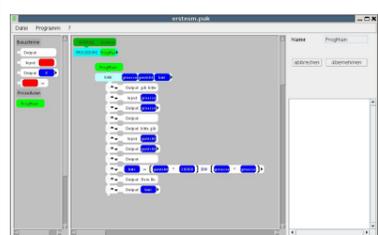
Übersicht – Zieldimension – Material

Puck ist eine visuelle Programmierumgebung, die es gestattet einfache imperativer Programme aus ca. 16 grafischen Modulen aufzubauen. Puck eignet sich besonders für den Einführungsunterricht zur imperativen Programmierung in der Unterstufe, beherrscht aber auch Prozeduraufufe mit Parametern und Rekursion. Den Teilnehmern wurde das Papier »Mit Puck einfach Programmieren lernen – Eine Einführung mit fünf Beispielen« zur Verfügung gestellt. Auf der Puck-Webseite www.uni-jena.de/puck.html finden sich weitere Materialien, Veröffentlichungen, Hilfen und eine ausgearbeitete Unterrichtsreihe zu Puck.

Funktionsumfang von Puck

In der nachfolgenden Grafik erkennt man die drei Bereiche des Puck-Interface: links

die Bausteinauswahl (momentan sind vier Bausteine aktiviert), in der Mitte das grafische Programm und links die Eigenschaften des gerade ausgewählten Elements:



Bildschirmfoto Puck

Puck stellt folgende Bausteine bereit:

- Textoutput, Input, Output und Zuweisung (im Bild von oben nach unten)
- Kontrollstrukturen: IF THEN, WHILE DO, REPEAT UNTIL, FOR
- Prozeduraufruf
- grafische Bausteine: Dot, Line, Color, Rect

- sonstige Bausteine: Sleep, Random, Sound

Fraglich ist, ob man wirklich drei verschiedene Schleifen oder Sound braucht. Das führt uns auch schon zu weiteren hilfreichen Funktionalitäten von Puck:

- Konfigurierbarkeit: Man kann Puck so konfigurieren, dass die Schülerinnen und Schüler nur die gewünschten Bausteine zur Auswahl haben. Dadurch kann man seinen Unterricht z.B. auf einen Schleifen-typ beschränken.
- Lehrermodul: Ferner gibt es noch ein Lehrermodul, bei dem der weitere Menüpunkt »Aufgabenkonstruktion« zur Verfügung steht. Dahinter verbirgt sich die Möglichkeit Pseudo-, Java- und Oberon2-Quelltext zu einem Puck-Programm zu generieren.

Das unterstützt zum Einen den Übergang zur quelltextbasierten Programmierung und gestattet zum Anderen die Konstruktion von Aufgaben.

Eine mögliche Aufgabe könnte

sein: Erstelle das Puck-Programm zu einem gegebenen Pseudoquelltext, oder integriere einen von Puck generierten Javaquelltext in dein Java-Programm.

Zur Erstellung von Arbeitsblättern lassen sich Bildschirmfotos erzeugen. Ein eher experimentelles Hilfsmittel ist die Vergleichsmöglichkeit der Komplexität von Lösungen. Mehr hierzu findet man auf der Puck-Webseite (www.uni-jena.de/puck.html).

Fazit

Der Workshop war ein echter und deshalb echt guter Workshop. Jede Teilnehmerin und jeder Teilnehmer musste(!) im Plenum kleine Aufgaben mit Puck vorführen. Dabei haben alle Teilnehmer schnell am eigenen Leib die motivierende, zum experimentieren anregende Wirkung des direkten Feedbacks der grafischen Programmierung erlebt. Ich würde Puck verwenden, um Schülerinnen und Schüler erste Erfahrungen mit Kontrollstrukturen sammeln zu lassen.

Zur fachlichen Einordnung

Aus meiner Sicht empfiehlt sich der Einsatz von Puck besonders zur Einführung von Kontrollstrukturen, des Variablenkonzeptes und Prozeduren mit Parametern, wobei die Möglichkeit zur Quellcodegenerierung (s.o.) aus grafischen Puck-Programmen den Übergang von Puck zur quellcodebasierten Programmierung sehr gut unterstützt. Man kann Puck sowohl in einer längeren, ausführlicheren Unterrichtsreihe in den Jahrgängen 7/8 oder in einer sehr kurzen Reihe in den Jahrgängen 9/10 für den schnellen, syntaxunabhängigen Zugang zur Einführung imperativer Programmstrukturen einsetzen. Der Vorteil liegt in beiden Fällen darin, dass sich die Schülerinnen und Schüler zunächst auf das Wesentliche an den Inhalten konzentrieren und schnelle Erfolgserlebnisse haben können, ohne dabei durch die Tücken der Codegenerierung behindert zu werden. Diese können später angegangen werden, wenn die Schülerinnen und Schüler z.B. schon wissen, welche Kontrollstrukturen es gibt, wie diese funktionieren und wozu man sie einsetzen kann.