

Anfangsunterricht im Fach Informatik

Datei anfangs2.doc

Einleitung

Der Beitrag beschäftigt sich mit dem Informatik-Anfangsunterricht, insbesondere nach dem neuen Berliner Lehrplan (1994). Nach einigen Anmerkungen zum Anfangsunterricht überhaupt, wird eine Unterrichtsreihe dargestellt, die sich um ein in der ITG verwendetes Softwareprodukt MUENZEN rankt.

Mehrere der sich in der 2.Phase der Lehrerausbildung befindlichen Referendare haben eigenverantwortlichen Unterricht in Anfängerkursen. Einige von ihnen haben kurze Erfahrungsberichte mit der Grobgliederung

- Unterrichtsvoraussetzungen
- Knappe Schilderung der Unterrichtsreihe
- Resumee

geschrieben. Diese Berichte sind insofern von besonderem Interesse, weil sich das Lehrplankonzept hier bei im Unterricht noch unerfahrenen Kollegen bewähren muß. Allerdings sind in der Seminararbeit immer wieder Diskussionen zu diesem Konzept erfolgt.

1. Bemerkungen zum Anfangsunterricht in Informatik, Klasse 11

Informatikunterricht beginnt häufig erst in Klasse 11, so daß der Start in dieses Fach besonderen Bedingungen unterliegt:

- Sehr inhomogene Lerngruppen wegen
 - möglicherweise vorhergehenden Unterrichts zur Informationstechnischen Grundbildung (ITG)
 - unterschiedlicher privater Vorerfahrungen der Schüler
 - eventuellen vorhergehenden Computereinsatzes in anderen Fächern
- falsche Vorstellungen der Schüler vom Fach Informatik

Nun sind inhomogene Lerngruppen noch nichts Negatives für den Unterricht, denn Inhomogenität schafft dem Lehrer die Möglichkeit sehr unterschiedliche Unterrichtsformen und -methoden einzusetzen. Jeder gute Schüler ist ein potentieller "Hilfslehrer". Die unterschiedlichen Schülerkompetenzen müssen nur ausgenutzt werden.

In Informatik allerdings potenzieren sich die Probleme, die hohen Zahlen das Fach abwählender Schüler sind bekannt.

So kommt gerade dem Anfangsunterricht besondere Bedeutung zu. Der häufig praktizierte Einstieg über die Erstellung vieler kleiner, meist unnützer Programme

ist ein Abschreckungsgrund für echte Anfängerschüler, denn hier sind die Schüler mit Vorerfahrungen den anderen sofort um Längen voraus.

In verschiedenen Bundesländern werden daher neuerdings andere Ansätze verfolgt.

Der Unterricht beginnt mit komplexen Systemen. Das können u.a. sein
 - professionelle Anwendersoftware,
 - umfangreichere Softwareprodukte, die vorher z.B. in Projektarbeit an Schulen erstellt wurden.

Die Vorzüge der Arbeit mit komplexen Systemen werden an anderer Stelle dieses LOGIN-Heftes gewürdigt (Lehmann: Komplexe Systeme als fundamentale Idee im Informatikunterricht).

Diese Ansätze haben auch den Vorteil, an Vorkenntnisse der Schüler aus der ITG anzuknüpfen.

Der Berliner Informatik-Rahmenplan sieht ab 1994 für den Anfangsunterricht (Unterrichtseinheit *Benutzung und Analyse eines dokumentierten Systems*, 10 Stunden) folgende Inhalte vor:

- "Bestimmung automatisierter Arbeitsschritte
- Zerlegung in geeignete Teilprobleme und Formulierung der Anforderungen an das System
- elementare Bedienung des Computers und Benutzung des Systems
- Mensch-Maschine-Schnittstellen, Funktionen des Systems
- Vergleich der von den Schülern formulierten Anforderungen mit der Realisierung durch das System
- Umgang mit der Benutzerdokumentation des Systems
- Bestandteile des Systems: Ausführbares Programm, Quellcode, zugehörige Dateien, Benutzerdokumentation
- Programm, Hauptprogramm, Prozeduren, Deklarationsteil, Blockstruktur, Eingaben, Ausgaben, Verarbeitung
- kleine Programmodifikationen, elementare Arbeiten im Editor, compilieren
- Wiederverwendbarkeit von Programmbausteinen "

Das Konzept besteht also in der Benutzung, Analyse, und (geringfügigen) Wartung der benutzten Software. Da diese Software in einen größeren (Anwendungs-) Zusammenhang eingebettet ist, erlebt der Schüler von Anfang an Datenverarbeitung so, wie sie auch in der Praxis vorkommen kann. Diese "Umgebung" soll dann auch in den Unterricht einbezogen werden.

2. Die Unterrichtsreihe MUENZEN

Das Programm MUENZEN wurde auf Anregung von Lehrern geschrieben, die an der FU Berlin an einem Fortbildungskurs zur ITG teilnahmen. Die ursprüngliche Zielsetzung bestand darin, in Anknüpfung an die Benutzung dieses Programms eine Unterrichtsreihe

zum Thema "Spielleidenschaft" zu konstruieren. Diese wurde für die ITG dann auch entworfen. Später wurde das Programm "zweckentfremdet" und für den Anfangsunterricht in Informatik eingesetzt, also mit den oben geschilderten (anderen) Intentionen.

Unterrichtsvoraussetzungen:

Die Schüler hatten in Klasse 9 ITG-Unterricht und sind somit mit den grundlegenden Bedienungselementen eines Computers vertraut. Hinzu kommen die individuellen Erfahrungen einzelner Schüler. - Beide Aspekte führen dazu, daß der Lehrer diese Kenntnisse und Fähigkeiten in den ersten Unterrichtsstunden in ihrem Umfang feststellen und reaktivieren muß.

Stunden 1/2: Die erste Doppelstunde wurde wie folgt durchgeführt:

- a1. "Münzen" - was fällt Ihnen dazu ein? (Brainstorming, u.a. wird sicher auch Münzwurf genannt)
- a2. Aspekte ordnen und Auswahl für die mittelfristige Planung der Unterrichtseinheit treffen
- b. Auslösen der Protokolle m.H. von Zufallszahlen und Taschenrechner
- c. Ablauf von (b) im Struktogramm darstellen
- d. Spiel MUENZEN vom Server laden
- e. Spielregeln aufrufen, Spiel von Hand spielen
- f. HA: Welche Begriffe aus der DVA wurden genannt?

Ziele: Die Schüler lernen einige Algorithmen kennen (Protokollvergabe mit Zufallszahlen, Spielregeln für Münzspiel). Sie können das Münsprogramm vom Server laden, die Spielregeln aufrufen und danach das Spiel von Hand spielen.

Im folgenden werden jeweils nur ausgewählte Phasen dargestellt!

b) Protokolle: Da die Lerngruppe (9 Schüler) auch als Seminarklasse dient, sollten die ersten Stunden des Anfangsunterrichts von den Schülern protokolliert werden. Hierzu wurde im Unterricht eine "gerechte" (zufällige) Protokoll-Reihenfolge entwickelt.

1. Eine Schülerreihenfolge wird festgelegt (Vornamen alphabetisch)
2. Solange noch Schüler ohne Protokollnummer vorhanden sind:
 3. Am Taschenrechner Taste RND drücken
 4. erste Nachkommastelle ablesen
 5. wenn diese Nachkommastelle schon vorkam oder gleich 0 ist, zu (3) gehen,
 6. andernfalls die Nachkommastelle dem jeweils aktuellen Schüler der Schülerreihe als Protokollnummer zuordnen

Damit haben die Schüler gleichzeitig den ersten Algorithmus kennengelernt, hier in einer verbalen Formulierung.

d) Das Spiel MUENZEN: Das Programm soll geladen werden.

- Das von uns zu benutzende Programm liegt auf Diskette vor. Wie bekommen wir es auf alle Rechner?
- Das Programm befindet sich auf dem Server. Wie bekommen wir es auf alle Rechner?

Hierbei lernen (wiederholen) die Schüler die Funktion des Servers im Rechnernetz:

- login leh-inb (damit wird das Kursverzeichnis lokal und gleichzeitig auf dem Server bereitgestellt)
- copy l:\muenzen.* c:

Wir stellen fest:

Das Programmsystem besteht aus den Dateien

- muenzen.bal (Datei mit dem Menü)
- muenzen.exe (das eigentliche Programm in kompilierter Form)
- muenzen.pas (das Quellprogramm)
- muenzen.mas (Bildschirmdatei mit grundlegenden Arbeitsmaske)
- muenzen.reg (Datei mit den Spielregeln)

e) Spielregeln, von Hand spielen

Regeln für das Münzspiel

- 1) Beteiligt sind drei Spieler.
- 2) Alle drei Spieler zahlen einen CHIP in den TOPF.
- 3) Nun werden zwei Münzen geworfen.
- 4) Spieler 1 gewinnt bei zwei Wappen (WW)
Spieler 2 gewinnt bei zwei Zahlen (ZZ)
Spieler 3 gewinnt in den Fällen WZ und ZW.
- 5) Der Gewinner entnimmt dem TOPF die drei Chips.
- 6) Es geht weiter bei Nr.2, es sei denn, man beendet das Spiel.

Diese Spielregeln können als ALGORITHMUS aufgefaßt werden, nach dem das Spiel abläuft und auch programmiert werden kann.

Stunde 3

Die Schüler benutzen das Programm MUENZEN mit bestimmten Arbeitsaufträgen und werden so mit der Programmoberfläche und den Programmfunktionen vertraut.

a) Informieren Sie sich durch Aufruf des Programms MUENZEN.EXE über die Arbeitsweise des Programms.

b) Welche Aufgaben lassen sich mit dem Spiel bearbeiten? Notieren!

Ziel: Mit Anwendersoftware kann man Probleme bearbeiten/lösen (hier ein wahrscheinlichkeits-theoretisches Problem)

b1) Chip-Anzahl zu Beginn jeweils = 10. - Wie lange (Anzahl der Spiele) dauert es, bis einer der Spieler seine Chips verloren hat? - Simulieren Sie das 5 mal! --> Statistik

b2) Notieren Sie die Ergebnisse von drei Spielen mit jeweils 100 Würfeln, Chip-Anzahl zu Beginn jeweils = 10. Hinweis: Achten Sie auf die Zwischenergebnisse!

b3) Verlauf in 10 er Schritten graphisch darstellen.

HA: Zeichnungen fertigstellen, auswerten

Ausschnitte aus der Durchführung:

MÜNZSPIEL

- 1 Spielen 1 (für Anfänger)
- 2 Spielen 2 (Fortgeschrittene)
- 3 Spielregeln
- 4 Textdatei zur freien Verfügung
- 5 Hinweise für den Lehrer
- 6 Programmende

SIMULATION EINES MÜNZSPIELS

```
-----
Anzahl von Chips   Anzahl der Gewinnspiele
Spieler 1           000           000
Spieler 2           000           000
Spieler 3           000           000
-----
```

Der nächste Münzwurf (j,n) = ? ---> ??

```
| WW: Spieler 1 gewinnt | ZZ: Spieler 2 gewinnt |
| ZW oder WZ: Spieler 3 gewinnt |
```

Spielprotokoll

```
Anzahl der Chips von Spieler 1 am Anfang = 10
Anzahl der Chips von Spieler 2 am Anfang = 10
Anzahl der Chips von Spieler 3 am Anfang = 10
```

Stunde 4: Auswertung der erzeugten Ergebnisse

Die Schüler können die mit dem Programm erzeugten Daten auswerten. Sie erkennen den Einfluß der Parameter (Anzahl der Chips zu Beginn).

Die vorgelegten Häufigkeitsdiagramm wurden ausgewertet.

HA: Quelltext geben - das Programm besteht aus Teilen, welche?

Das Quellprogramm (Anlage für die Hausarbeit)

```

PROGRAM muenzwurf_simulation;
USES Crt, Ino_u, Bild_u, Balk_u;
VAR wurf: CHAR;
    ergebnis: STRING[2];
    gewinn_spieler1,gewinn_spieler2,gewinn_spieler3: INTEGER;
    chips_spieler1,chips_spieler2,chips_spieler3 : INTEGER;
{-----}
PROCEDURE chips_am_anfang;
BEGIN
    Textcolor(lightgray);
    Input_integer(1,18,'Anzahl der Chips von Spieler 1 am Anfang = ', chips_spieler1,2);
    Input_integer(1,19,'Anzahl der Chips von Spieler 2 am Anfang = ', chips_spieler2,2);
    Input_integer(1,20,'Anzahl der Chips von Spieler 3 am Anfang = ', chips_spieler3,2);
END;
{-----}
PROCEDURE chips_einzahlen;
BEGIN
    chips_spieler1:=chips_spieler1-1;
    chips_spieler2:=chips_spieler2-1;
    chips_spieler3:=chips_spieler3-1;
END;
{-----}
PROCEDURE initialisieren;
BEGIN
    ergebnis:='??';
    gewinn_spieler1:=0; gewinn_spieler2:=0; gewinn_spieler3:=0;
    chips_spieler1 :=10; chips_spieler2 :=10; chips_spieler3 :=10;
END;
{-----}
PROCEDURE muenzwurf;
{Münzen werfen und Ergebnisse berechnen}
VAR x,y: REAL;
    spielanzahl: INTEGER;
BEGIN
    x:=Random; y:=Random;
    IF (x<0.5) AND (y<0.5) THEN
    BEGIN ergebnis:='WW'; gewinn_spieler1:=gewinn_spieler1+1;
        chips_spieler1 :=chips_spieler1+3;
    END
    ELSE
    IF (x>=0.5) AND (y>=0.5) THEN
    BEGIN ergebnis:='ZZ'; gewinn_spieler2:=gewinn_spieler2+1;
        chips_spieler2 :=chips_spieler2+3;
    END
    ELSE
    IF (x>=0.5) AND (y<0.5) THEN
    BEGIN ergebnis:='ZW'; gewinn_spieler3:=gewinn_spieler3+1;
        chips_spieler3 :=chips_spieler3+3;
    END

```

```

END
ELSE
IF (x<0.5) AND (y>=0.5) THEN
BEGIN ergebnis:='WZ'; gewinn_spieler3:=gewinn_spieler3+1;
      chips_spieler3 :=chips_spieler3+3;
END;

{Ergebnisse eintragen}
Textcolor(yellow);
Gotoxy(41,13);WRITE(ergebnis); Textcolor(lightgray);
Gotoxy(25,6);WRITE(chips_spieler1:3);
Gotoxy(25,8);WRITE(chips_spieler2:3);
Gotoxy(25,10);WRITE(chips_spieler3:3);
Gotoxy(44,6); WRITE(gewinn_spieler1:3);
Gotoxy(44,8); WRITE(gewinn_spieler2:3);
Gotoxy(44,10);WRITE(gewinn_spieler3:3);
spielanzahl:=gewinn_spieler1+gewinn_spieler2+gewinn_spieler3;
Gotoxy(1,25); WRITE('Spielanzahl = ', spielanzahl:6);
Textcolor(lightgray);
END;
{-----}
PROCEDURE spielen_anfaenger;
BEGIN
Textcolor(lightgray);
Randomize; Initialisieren;
dateiname:='muenzen.mas'; textbild_aus(dateiname);
chips_am_anfang;
REPEAT
chips_einzahlen;
wurf:='?';
Input_zeichen(30,13,"wurf,['j','n','J','N']);
IF wurf IN ['j','J'] THEN muenzwurf;
UNTIL wurf IN ['n','N'];
END;
{-----}
PROCEDURE spielen_fortgeschrittene;
VAR c: CHAR;
BEGIN
WRITELN('Hier kann sp.,ter eine zweite Spielversion aufgerufen werden. ');
WRITELN('In dieser kann man z.B. die M□nze "zinken"');
c:=Readkey;
END;
{-----}
PROCEDURE spielregeln;
VAR c: CHAR;
BEGIN
Output_txtdatei(1,1,80,25,'muenzen.reg','');
END;
{-----}

```

```

PROCEDURE lehrer_hinweise;
VAR c: CHAR;
BEGIN
  WRITE('Die Lehrerhinweise sind noch nicht formuliert. ');
  c:=Readkey;
  Output_textdatei(1,1,80,25,'muenzen.leh','');
END;
{-----}
PROCEDURE textdatei_zur_freien_verfuegung;
VAR c: CHAR;
  dateiname: STRING[12];
BEGIN
  WRITELN('An dieser Stelle können Sie eine Textdatei mit eigenen');
  WRITELN('Texten aufrufen. Die Datei heißt MUENZEN.AUF - sie muß');
  WRITELN('als ASCII-Datei erstellt werden,z.B. im Turbo-Pascal-Editor. ');
  WRITELN('In diese Datei können Sie z.B. Aufgaben oder Informationen');
  WRITELN('für die Schüler hineinschreiben. ');
  c:=Readkey;
  dateiname:='muenzen.auf';
  Output_textdatei(1,1,80,25,dateiname,'');
END;
{-----}

BEGIN
REPEAT
  Clrscr;
  Textcolor(magenta); Gotoxy(10,2);
  WRITE('MÜNZSPIEL          Autor: Lehmann,E. - Berlin,d.8.5.94');
  Textcolor(lightgreen); Gotoxy(10,25);
  WRITE('Zum Auswählen Cursor auf/ab bewegen. Dann ENTER-Taste drücken. ');
  dateiname:='muenzen.bal';
  Menue_wahl(lightgreen,yellow,dateiname,option);
  Clrscr;
  CASE option OF
    1: spielen_anfaenger;
    2: spielen_fortgeschrittene;
    3: spielregeln;
    4: textdatei_zur_freien_verfuegung;
    5: lehrer_hinweise;
    6: ;
  END;
UNTIL option=6;
Clrscr;
END.

```

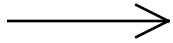
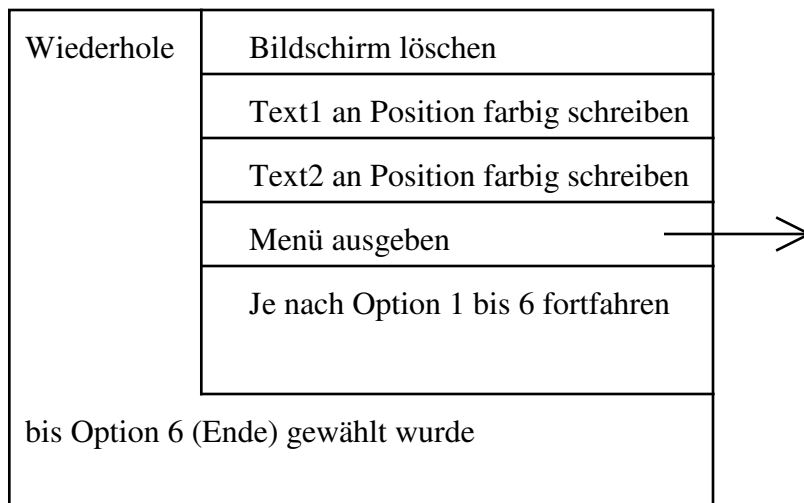

Stunden 5/6

Ziele: Erkennen der internen Programmstruktur, Prozedurbegriff kennenlernen, Kennenlernen des Turbo-Pascal-Editors und Arbeit im Editor üben.

- a) Besprechung der Hausarbeit - Prozeduren notieren
- b) Wie kann man ein (PASCAL-)Programm analysieren und damit verstehen?
- c) Das Programm auf dem Bildschirm ansehen, den Editor benutzen lernen, Leerzeilen zwischen den Blöcken einschieben.
- d) Sie haben bei der Programmbenutzung bemerkt, daß einige Menüpunkte noch nicht bearbeitet sind! - Entwerfen Sie eine Information für den Lehrer (Menüpunkt 5), was er mit dem Programm anfangen kann.
- e) Das Spiel wird erneut aufgerufen (nun ist Menüpunkt 5 sinnvoll belegt mit der in c) geschriebenen Datei).

b) Programmanalyse

Der folgende Text stammt aus dem Schülerprotokoll: "Man sucht sich das Hauptprogramm und versucht es zu analysieren. Für das Beispiel MUENZEN.PAS ergab sich der folgende Algorithmus in Struktogrammform:



Dieser Programmschritt beinhaltet einen sogenannten Baustein. Die Anweisungen für diesen Baustein sind in einer anderen Datei (MUENZEN.BAL) gespeichert. Folgende Fragen interessieren uns zu diesem Baustein:

- Wo befindet er sich?
- Welche Funktion erfüllt er?
- Wie kann man ihn in das Programm integrieren?
- Wie konstruiert man solch einen Baustein?

Während die ersten drei Fragen eher den Benutzer des Bausteins interessieren, ist die letzte Frage eher für den Programmierer von Bedeutung."

Hinweis: Die Schüler können schon im Anfangsunterricht mit Hilfe des Programms MENUE auf einfache Weise Menüs konstruieren. Diese werden dann mit Hilfe des Bausteins "menue_wahl(...)", der sich in einer Bausteinbibliothek (Unit) befindet, in das Pascal-Programm eingebunden.

Stunde 7: Bausteine

Die Schüler erkennen die Bedeutung der Verwendung von Programmbausteinen.

Der Blick der Schüler wird nun auf die Bausteine gelenkt.

a) Bei der Programmierung wurden diverse Bausteine verwendet. - Markieren Sie die entsprechenden Aufrufe im Programmtext, und versuchen Sie diese zu erläutern.

Die Schüler stellen u.a. fest:

```
Input_integer(1,18,'Anzahl der Chips von Spieler 1 am Anfang = ', chips_spieler1,2);
Menue_wahl(lightgreen,yellow,dateiname,option);
Output_textdatei(1,1,80,25,dateiname,"");
Input_zeichen(30,13,',wurf,['j','n'],'J','N');
```

b) Welche Vorteile bietet die Verwendung dieser Bausteine?

c) Wie bindet man Bausteine aus Modulbibliotheken (Units) in sein Programm ein?

Und unabhängig vom gegenwärtigen Thema, aber dennoch sehr wichtig:

d) Inzwischen haben sich einige Protokolle angesammelt. Schwächen und Stärken werden besprochen. Es wird zusammengestellt, welche Punkte ein Protokoll enthalten sollte.

Stunde 8: Programmänderungen

Ziel: Die Schüler studieren die Auswirkungen kleiner Programmänderungen. Sie lernen ein Programm compilieren.

Für kleine Programmänderungen gibt es viele Möglichkeiten. In diesem Fall erhielten sie im Rahmen einer Lehrprobe den Auftrag, Farben anders zu setzen und eine Gewinnmöglichkeit (Wappen,Zahl) aus dem Programm herauszunehmen. Diese Änderung hat dann Auswirkungen auf die Gewinnwahrscheinlichkeiten.

Stunde 9: Grundsätzlicher Aufbau eines PASCAL-Programms

Die Schüler begreifen den Aufbau eines PASCAL-Programms.

In Stunde 9 wurde der grundsätzliche Aufbau eines Pascal-Programms betrachtet. Ausgehend von MUENZEN.PAS wurden die Anweisungen soweit gelöscht, daß nur noch das Programmgerippe übrigblieb.

```
PROGRAM muenzwurf_simulation;
USES Crt, Ino_u, Bild_u, Balk_u;
VAR wurf: CHAR;
    ergebnis: STRING[2];
```

```

gewinn_spieler1,gewinn_spieler2,gewinn_spieler3: INTEGER;
chips_spieler1,chips_spieler2,chips_spieler3 : INTEGER;
option: INTEGER;
{-----}
PROCEDURE chips_am_anfang;
BEGIN
END;
{-----}
PROCEDURE chips_einzahlen;
BEGIN
END;
{-----}
PROCEDURE initialisieren;
BEGIN
END;
{-----}
PROCEDURE muenzwurf;
BEGIN
END;
{-----}
PROCEDURE spielen_anfaenger;
BEGIN
END;
{-----}
PROCEDURE spielen_fortgeschrittene;
BEGIN
END;
{-----}
PROCEDURE spielregeln;
VAR c: CHAR;
BEGIN
END;
{-----}
PROCEDURE lehrer_hinweise;
BEGIN
END;
{-----}
PROCEDURE textdatei_zur_freien_verfuegung;
BEGIN
END;
{-----}
BEGIN
CASE option OF
1: spielen_anfaenger;
2: spielen_fortgeschrittene;
3: spielregeln;
4: textdatei_zur_freien_verfuegung;
5: lehrer_hinweise;
6: ;
END;
END.

```

Durch eine Umkehrung der Sichtweise wurde dadurch auch klar, wie die frühere Programmkonstruktion stattgefunden haben kann.

Zusammenfassung: Damit sind die Lernziele für die erste Unterrichtseinheit im wesentlichen erfüllt. Die nun folgende Unterrichtseinheit "Teilalgorithmen zu Anwendungsfällen" geht weiterhin davon aus, daß die betrachteten Algorithmen in größere Zusammenhänge eingebettet werden, was den Schülern auch bewußt zu machen ist.