

Programmieren spielend gelernt mit dem Java-Hamster Modell

12.11.09 14.30-17.00Uhr (150min)

120min Vortrag
mit Einführung in das Programm
unterbrochen durch selbstständigen Umgang mit dem Programm
20min Erste Eindrücke, Anregungen, Ausblick OOP

Unterrichtsgang im Fach Informatik (2 WS) in der Klasse 11_2

I Grundlagen der Programmierung

- 1) Programmierung: Ziele, Algorithmen, Formulierung von Algorithmen
- 2) Programmiersprachen: Klassifikation, Syntaxdarstellung
- 3) Programmentwicklung, Entwicklungswerkzeuge
- 4) Boolesche Aussagen

9 Kurzreferate zu PAP/ Flussdiagramm, Struktogramm
 imperative, funktionale, prädikative, regelbasierte, objektorientierte
 Programmiersprachen
 Syntaxdiagramm, BNF

II Imperative Programmierung

- 5) Grundlagen Hamstermodell
- 6) Anweisungen und Programme
- 7) Prozeduren
- 8) Auswahlanweisungen (if else)
- 9) Wiederholungsanweisungen (while, do while)
- 10) Boolesche Funktionen
- 11) Programmentwurf (Kapitel aus Buch kopieren Inhalt Thema der Klausur)
- 12) Boolesche Variablen
- 13) Zahlen, Variablen und Ausdrücke Ende 11_2
- 14) Prozeduren und Funktionen (int Funktionen, Verallgemeinerung des Funktionskonzeptes)
- 15) Funktionsparameter (Funktionen mit Parametern, Überladen von Funktionen, Parameterliste variabler Länge)
- 16) Rekursionen (nicht zu schaffen?! nicht ausgeführt)

Zu I (Kapitel 1-4)

Wichtig sind die Struktogramme und Sytaxdiagramme, der Umgang mit dem Struktogramm kann auch mit hase an späterer Stelle noch einmal geübt werden.
Die Sytaxdiagramme spielen bei jedem neuen Kapitel wieder eine Rolle. Die Hamstersprache wird soweit in Sytaxdiagramme gefasst, so dass am Ende die gesamte Sprache in Diagrammen darstellbar ist.

Boolesche Aussagen nur Negation, Konjunktion (und &&) und Disjunktion (oder | |) Merksatz:
Aussagen P und Q sind nur wahr, wenn sowohl P als auch Q wahr sind.
Aussagen P oder Q sind nur falsch, wenn sowohl P als auch Q falsch sind.

Zu II

Nach der Theorie ist die Motivation groß nun endlich zu hamstern.

- 5) Grundlagen
Umgang Kommentare
Programmierstil (Einrücken usw.)

Komponenten der Hamstersprache (Attribute, Methoden, Territorium)
Reservierte Wörter

6) Anweisungen und Programme

Demonstration erster Hamster Beispielprogramme

Syntaxdiagramm, Semantik der vier Grundbefehle `vor()`; `linksUm()`; `gib()`; `nimm()`; ,
Laufzeitfehler

Erstes Aufgabenblatt zum Programmieren

Die Aufgabenblätter sind so konzipiert, dass in den ersten Aufgaben die Beispielprogramme umgebaut werden müssen. Die weiteren Aufgaben bauen den Schwierigkeitsgrad auf. In den Aufgaben werden unter anderem auch Lösungen erarbeitet, die an anderer Stelle (späteren Aufgabenblättern) modifiziert wieder verwendet werden. In den Aufgaben werden auch immer wiederkehrende Territorien benutzt und mit dem jeweils neu eingeführten Programmierelement bearbeitet.

7) Prozeduren

Erweitern der Grundbefehle.

Schüler forderten selber die Einführung von Prozeduren, insofern war es leicht die Vorteile zu benennen. Prozedur ist ein erst einmal benutztes Wort aus der imperativen Programmierung und meint Funktionen ohne Rückgabewert.

Prozedurdefinition

Syntax, Bezeichner (Konventionen in Java), Semantik (keine)

Prozeduraufruf

Syntax, Semantik (Verzweigung in den Rumpf der Prozedur)

Programmierbeispiele mit syntaktischen Fehlern werden gemeinsam besprochen.

Zu Aufgabenstellungen werden Beispielprogramme gezeigt und besprochen, anschließend gehen die Schüler wieder in eine Programmierphase mit dem zweiten Aufgabenblatt.

Die von den Schülern entwickelten Programme werden gemeinsam oder in Partnerarbeit besprochen. Die Schüler erkennen, wenn ihr Hamster ohne Laufzeitfehler die geforderte Aufgabe erledigt, ist die Aufgabe gelöst. Von jedem Aufgabenblatt werden die schwierigeren Aufgaben gemeinsam an der Wand vorgestellt.

8) Auswahlanweisungen (if else)

Nutzen der Testbefehle `vornFrei()`, `maulLeer`, `kornDa()` im Programm, vermeiden von Laufzeitfehlern.

Testbefehle

Syntax, Semantik (Rückgabewerte der Methoden/ booleschen Funktionen `true` oder `false`), Exkurs zu booleschen Ausdrücken (Prioritäten, linksassoziativ, Auswertungsreihenfolge)

if (if else)

Syntax, Semantik (falls der Wert in der runden Klammer „true“ liefert wird die Anweisung ausgeführt, sonst die Alternativanweisung)

Vorstellen einiger Beispiele, geschachtelte if-Anweisungen, sichere Grundbefehle

Programmierphase Auswahlanweisungen

9) Wiederholungsanweisungen while/ do while

Syntax (mit geschweiften Klammern), Semantik (Überprüfung der Schleifenbedingung (boolescher Ausdruck), falls Wert `false` wird die Schleife sofort beendet. Falls wert `true` wird die Iterationsanweisung einmal ausgeführt, anschließend Bedingung neu überprüfen usw., `do while` wird mindestens 1x die Anweisung ausgeführt))

Keine geschachtelten Schleifen, keine Endlosschleifen

In den folgenden Programmierphasen werden keine Hamsterlandschaften mehr angegeben, sondern nur noch charakteristische Merkmale beschrieben (min Zeilen, min Spalten, quadratisch, innere Mauern, Anzahl Körner/Maul, regelmäßiger Berg, Blickrichtung und Standort Hamster). Das Programm arbeitet korrekt, wenn für alle Landschaften keine Laufzeitfehler auftreten, syntaktisch fehlerfrei, nach endlicher Zeit beendet ist (falls nichts anderes gefordert wurde).

10) Boolesche Funktionen (Kunstwort)

Analog zu Prozeduren, z.B. zum Erweitern der Testbefehle.

return Anweisung

Syntax, Semantik (Beendigung der Funktion, Rückgabewert des booleschen Ausdrucks)

Definition der booleschen Funktion

Syntax, Bezeichner (Konventionen in Java), Semantik (keine)

kurze fehlerhafte Programmsequenzen werden besprochen, Seiteneffekte

sollen im Kommentar erwähnt werden, falls diese nötig sind. Ansonsten gilt es sie zu vermeiden.

Aufruf boolescher Funktion

Syntax, Semantik (Verzweigung in den Rumpf der Prozedur)

Programmierphase zu booleschen Funktionen

11) Programmentwurf

Kapitel wird kopiert (Aufgabe => Analyse, Entwurf, Implementierung, Test, Dokumentation)

12) Boolesche Variablen

Der Hamster bekommt ein Gedächtnis.

Da in I, Kapitel 8) und 10) immer wieder mit booleschen Ausdrücken und Methoden gearbeitet wurde, wird nun der Begriff und der erste Umgang mit Variablen am Datentyp boolean erarbeitet. Dies hatte zur Folge, dass der Umgang mit Variablen vom Datentyp integer zu kurz kam.

Motivation der Variablen, Definition, Syntax, Semantik (keine Programmauswirkungen, Wert des bA wird als Initialwert gespeichert), Gestaltungskonventionen

Nutzung b Variablen

Boolesche Zuweisung (rechtsassoziativ)

Gültigkeitsbereich, lokale, globale Variable

Lebensdauer der Variablen

Gerade zu den letzten vier Punkten werden einige (auch fehlerhafte)

Programmsequenzen besprochen. Die Schüler werden dazu aufgefordert die

Beispielprogramme in Partnerarbeit zu erläutern und zu verstehen. Anschließend

gehen sie wieder in eine Programmierphase.

13) Zahlen, Variablen, Ausdrücke

Hamster soll Rechnen lernen.

Datentyp integer wird eingeführt

Arithmetische Ausdrücke, Ziffern, unäre (+!) und binäre (&& | | + - * / % == != < <= > >=) Operatoren

In selbstständiger Arbeit lesen die Schüler über andere Datentypen (short, long, char, byte float, double) und Typumwandlung

Programmsequenzen werden besprochen, Beispielprogramme erläutert. Anschließend sollten die Schüler in eine kurze Programmierphase gehen.

14) Prozeduren und Funktionen in 12_1

Analog zu den booleschen Funktionen werden int Funktionen eingeführt.

- return Anweisung
- Syntax, Semantik (Beendigung der Funktion, Rückgabewert des booleschen Ausdrucks)
- Definition der int Funktion
 - Syntax, Bezeichner (Konventionen in Java), Semantik (keine)
 - kurze fehlerhafte Programmsequenzen werden besprochen, Seiteneffekte
- Aufruf int Funktion
 - Syntax, Semantik (Verzweigung in den Rumpf der Prozedur)

Verallgemeinerung des Funktionskonzepts, Funktionstyp void kann eine return Anweisung ohne Ausdruck besitzen.

Beispielprogramme und eine Programmierphase runden dieses Kapitel ab.

15) Funktionsparameter in 12_1

Der Einsatz von Parametern wird motiviert, damit die Funktionen und Prozeduren mehr Flexibilität erhalten.

Definition von Funktionen mit Parametern

- Syntax, formale Parameterliste (Konvention der Parameter/Variablen)

Aufruf von Funktionen mit Parametern

Syntax, Parameterübergabe Der Wert des aktuellen Parameters wird beim Aufruf einer Funktion einem formalen Parameter der Funktion als Initialisierungswert übergeben (call-by-value-Parameterübergabe). Anzahl und Typ der formalen Parameter muss mit den aktuellen Parameter übereinstimmen! Fehlerhaften Code besprechen

Funktionsaufrufe als aktueller Parameter z.B. felderVor (summe(3));

Auswertungsreihenfolge der aktuellen Parameterliste von links nach rechts z.B.

```
int n=3;
```

```
int s= summe(n,n=n+1,n);
```

Überladen von Funktionen, zwei Funktionen dürfen den selben Namen besitzen, wenn sie sich durch die Anzahl der Typ der formalen Parameter unterscheiden. Fehlerhaften Code besprechen.

varargs-Konstrukt (Funktionen mit variabler Parameterlänge) ??

Beispielprogramme und Programmierübungen

16) Rekursionen

werden eingeführt, falls ich sie in 12_1 benötigen sollte, sonst nicht vgl. Lehrplan