
Programmierstarthilfe SS 2008
Fakultät für Ingenieurwissenschaften und Informatik

1. Blatt
Für den 21. und 22.4.2008

Organisatorisches

Die Monitore im Pool nicht ausschalten! Auch die Rechner bleiben an. Sie schalten sich nach ca. einer halben Stunde von allein ab.

Um auf die Mailingliste aufgenommen zu werden schicke einfach eine Mail an `guido.de-melo@uni-ulm.de`.

Die Webseiten zur Veranstaltung sind unter <http://www.uni-ulm.de/in/mi/lehre/ss2008/programmierstarthilfe.html> zu finden.

Mit diesem Blatt soll zuerst ein Grundverständnis für Algorithmen erarbeitet werden. Im Anschluss werden Datentypen eingeführt. Danach folgen einfache Verzweigungen. Im letzten Schritt sollen einfache Schleifen angewandt werden.

1. Aufgaben zu Algorithmen

Um bei der Erstellung eines Programms oder Algorithmus strukturiert vorzugehen verwendet man oft so genannten Pseudocode. Er ist eine Art Grundplan für den Algorithmus. Grundsätzlich besteht er aus Konstrukten, die auch in der Programmiersprache vorkommen, so z.B. Wiederholungen oder Bedingungen. Trotzdem können mehrere Anweisungen der Programmiersprache auch zu Blöcken zusammengefasst werden, deren Sinn durch Benennung in natürlicher Sprache ersichtlich ist. Normalerweise startet man mit einem groben Modell und verfeinert dann die Anweisungen schrittweise.

Bedingungen:

```
1 wenn Bedingung_gilt dann
2     tue_dies
3 ansonsten wenn andere_Bedingung_gilt dann
4     tue_das
5 ansonsten
6     tue_irgendwas
7 wenn_ende
```

Wiederholungen:

```

1 wiederhole solange Bedingung_gilt
2   tue dies
3 wiederhole_ende
    
```

Selbsterklärende Funktionen:

```

1 braue_frischen_Kaffee
2 zeige_zahl_an(42)
    
```

1.1. Mensabesuch

Schreibe einfachen Pseudocode für einen Besuch in der Mensa. Grundlegende Funktionen, z.B. `anstellen()`, `finde_essbares()`, `nimm()`, `bezahl()`, `iss()` können als gegeben angenommen werden.

1.2. Zähne putzen

Beschreibe den Algorithmus Zähne putzen, d.h. „bürstengestützte Zahnpflege in vernünftigem Umfang mit nicht-elektrischen Hilfsmitteln“. Beschreibe den Algorithmus zunächst grob in Pseudocode und benutze dann schrittweise Verfeinerung. Der Detaillierungsgrad soll so fein werden, dass ein Mensch ihn ohne zahnputztechnische Vorkenntnisse ausführen kann. Überlege dabei auch, ob welche grundlegenden Funktionen schon vordefiniert sein sollten.

1.3. Algorithmus zur Bestimmung des Maximums einer Menge

In einem Raum mit Studierenden soll das Alter der ältesten Person festgestellt werden. Welche Schritte sind hierfür nötig, um dies herauszufinden? Schreiben Sie den dafür verwendeten Algorithmus in Pseudocode nieder.

1.4. Ganzzahlen

- a) Ändere das "HelloWorld"-Programm so ab, dass es die Summe von 5 und 3 berechnet, und ausgibt. Ersetze dafür in der Zeile `System.out.println("HelloWorld!");` den String durch `5+3` (dazu musst du auch die Anführungszeichen entfernen).
- b) Ändere das Programm aus a) so ab, dass die Zeichen `5+3` ausgegeben werden, statt `8`.

1.5. CDs Sortieren

Du sortierst deine CD-Sammlung, zu Beginn hast du einen großen Stapel von CDs vor dir liegen. Nach und nach sollen die CDs, die du nicht mehr hörst einfach weggeworfen werden. Alles was übrig ist soll einen Aufkleber erhalten, der die Stilrichtung angibt. Die CDs gehören in die Kategorien House (gelb), Rock (blau), HipHop (schwarz) und Sampler (rot). Erstelle dazu einen Algorithmus in Pseudocode.

1.6. PI-Übung

Ein Mitstudent hat das Verfahren, in dem die PI-Übung gehalten wird, mit dem ganzen Votieren und Erklären und Fehler der anderen korrigieren (verständlicherweise) nicht verstanden. Erkläre ihm möglichst einfach, wie er sich in welchen Phasen seines Tutoriums verhalten sollte, und was er in welcher Reihenfolge dort machen sollte, damit zumindest nicht auffällt, dass er das System nicht verstanden hat. Versuche möglichst vom Beginn des Tutoriums an bis zum Ende (chronologisch) zu beschreiben was passieren kann und dabei auch einige Eventualitäten abzudecken. Versuche auch zu erklären, wo er Punkte sammeln kann.

Diese Aufgabe lässt sich umgangssprachlich lösen, mach dir einige systematische Notizen, aus denen der Tutoriums-Ablauf ersichtlich wird und trage dann alles kurz und sortiert vor.

2. Aufgaben zu Datentypen

Variable: Ein Name für einen Speicherbereich in einem Computerprogramm, der verschiedene Werte annehmen kann (siehe auch Wikipedia). Deklaration: Dem Programm den Typ und Namen einer Variable bekannt geben, z.B.

```
1 int meinErsterInteger;
```

der Typ ist hier int. Zuweisung: Einer Variablen einen Wert geben, z.B.

```
1 meinErsterInteger = 5;
```

In Java heißen Funktionen Methoden und fassen Anweisungen zusammen. Methoden haben einen Rückgabotyp, der vor ihrem Namen steht. Mit dem Schlüsselwort return wird dann das Ergebnis bezeichnet. Es können ihnen auch Parameter übergeben werden. Als Beispiel hier eine Methode zum Ausrechnen des Quadrats:

```
1 static int quadrat(int x){
2     return x * x;
3 }
```

2.1. Variable 1

- a) Schreibe ein Programm, in dem eine Variable `a` vom Typ `int` deklariert wird und dann den Wert 5 zugewiesen bekommen. Gib mit `System.out.println(a)`; den Wert der Variablen aus.
- b) Deklariere ein zweite Variable `b` und weise ihr in der gleichen Zeile den Wert zu. Gib mit `System.out.println(a+b)`; die Summe der Variablen aus.

2.2. Variable 2

Ändere das Programm von Aufgabe Variable 1 so ab, dass eine dritte Variable `c` vom Typ `int` deklariert und ihr der Wert der Summe von `a` und `b` zugewiesen wird. Gib anschließend den Wert von `c` aus.

2.3. Boolean 1

Es gibt in Java einen besonderen Datentyp, der nur zwei Werte annehmen kann: `boolean`. Dieser `boolean` kann die Werte `true` oder `false` annehmen.

- a) Schreibe ein Programm, dass eine Variable vom Typ `boolean` deklariert und dann der Wert der Variablen ausgibt.
- b) Weise der Variablen den Wert `true` zu.

2.4. Boolean 2

In Java gibt es die Möglichkeit Vergleiche zwischen z.B. `int` zu machen. Es gibt `<<<=>==` `!=`. Beachte: Das Gleichheitszeichen ist schon für die Zuweisung vergeben. Das Ergebnis eines Vergleiches ist vom Typ `boolean`.

- a) Ändere das Programm der Aufgabe Boolean so ab, dass dem `boolean` das Ergebnis des Vergleichs `5<=10` zugewiesen wird.
- b) Deklariere weitere Variablen, denen du jeweils den Wert eines beliebigen weiteren Vergleichs zuweist. Gib jede Variable aus.

2.5. PrintIntials

Modifiziere das HelloWorld-Programm in ein Programm `PrintIntials`, dessen `main`-Methode deine Initialen in der folgenden Form ausgibt, hier im Beispiel FS:

```

1 ****  ****
2 *      *
3 **     ****
4 *       *
5 *       ****
    
```

In deinem Programm dürfen deine Initialen fest einprogrammiert sein. Es ist nicht notwendig, diese vom Benutzer zu erfragen. Wer sich durch diese Aufgabenstellung unterfordert fühlt, dem ist es natürlich freigestellt, dies noch zusätzlich zu programmieren.

3. Aufgaben zu Verzweigungen

Um Entscheidungen treffen zu können braucht man in Java Verzweigungen:

```

1 if ( Bedingung ) {
2     Code
3 } else {
4     Code
5 }
    
```

Wenn die Bedingung zutrifft, dann wird der Teil zwischen den geschweiften Klammern ausgeführt, sonst nicht. Die Bedingung muss vom Typ Boolean sein, z.B. das Ergebnis eines Vergleichs a;b. Mit else kann man angeben was passieren soll, falls die Bedingung nicht erfüllt ist.

Switch

Für Mehrfachverzweigungen ist oft einfacher statt vielen if-else ein switch zu schreiben, mit dem mehrere Fälle unterschieden werden können:

```

1 switch ( Ausdruck ) {
2     case konstante1 : anweisung1 ; break ;
3     case konstante2 : anweisung2 ; break ;
4     case konstante3 : anweisung3 ; break ;
5     default : anweisung ;
6 }
    
```

Mit Hilfe von break endet ein Fall, ansonsten werden werden die Anweisungen der nächsten Fälle ebenfalls ausgewertet. default sammelt alle übrigen Fälle, die nicht durch Konstanten abgedeckt werden.

3.1. Verzweigung

Schreibe ein Programm, in dem eine Variable `a` vom Typ `Boolean` deklariert und mit `true` initialisiert wird. Anschließend soll mit

```

1  if (a){
2      System.out.println("Hallo, ");
3  }
4  System.out.println("<insert_your_name_here>");
    
```

'Hallo, <insert_your_name_here>' ausgegeben werden. Was passiert wenn `a` mit `false` initialisiert wird?

3.2. HelloWorld mit Tageszeit

Eine HelloWorld-Ausgabe soll so erweitert werden, dass der Benutzer abhängig von der Tageszeit begrüßt wird. Um die aktuelle Stunde als Integer-Wert zu bekommen, kann folgender Befehl abgetippt und benutzt werden:

```

1  int stunde = new GregorianCalendar().get(GregorianCalendar.HOUR_OF_DAY);
    
```

Damit dieser Befehl funktioniert, muss außerdem folgende Zeile am Anfang des Quellcodes eingefügt werden:

```

1  import java.util.GregorianCalendar;
    
```

Ändere nun ein normales HelloWorld so ab, dass der Benutzer morgens und abends mit einem besonderen Text begrüßt wird.

3.3. Switch Ziffern

Implementiere eine Methode, welche als Parameter eine Ziffer als Zahl `int` übergeben bekommt und diese dann als Wort (z.B.: "eins" auf dem Bildschirm ausgibt. Verwende dazu das `switch`-Konstrukt.

Wird eine Zahl übergeben, die aus mehr als einer Ziffer besteht, soll eine Fehlermeldung ausgegeben werden.

4. Aufgaben zu Schleifen

Schleifen dienen dazu Anweisungen mehrfach auszuführen. For-Schleifen zählen dabei wie oft etwas gemacht wird, while-Schleifen machen etwas so lange eine Bedingung gilt. `i++` ist die Kurzform von `i=i+1`;

```

1 for (int i=1; i<10; i++) {
2     Code
3 }
4 while (Bedingung) {
5     Code
6 }
    
```

4.1. Zählen

Schreibe ein Programm, das die Zahlen von 1 bis 100 ausgibt. Ändere es folgendermaßen ab: Zunächst alle Zahlen in einer Zeile, dann mit Leerzeichen dazwischen und dann jede Zahl in einer eigenen Zeile. Du kannst mit `System.out.print Text` ausgeben, mit `System.out.println` wird automatisch ein Zeilenumbruch angehängt.

4.2. * Zahlenraten

Schreibe ein einfaches Ratespiel. Der Spieler soll eine Zufallszahl erraten die zwischen 0 und einer vom Spieler gewählten Obergrenze liegt. Der Spieler darf einen Tipp abgeben. Das Programm antwortet jeweils ob die getippte Zahl kleiner oder größer als die Zufallszahl ist. Das Spiel läuft solange bis der Spieler die Zahl erraten hat. Zusätzlich soll die Anzahl der Versuche gezählt und am Ende ausgegeben werden. Verwende die Hilfsklasse `Util`.

4.3. GgtAlgorithmus

Der größte gemeinsame Teiler (ggT) zweier Ganzzahlen n und m ($n, m \geq 0$) ist die größte natürliche Zahl, durch die beide Zahlen ohne Rest teilbar sind. Für die Bestimmung des ggT gibt es ein sehr altes und sehr berühmtes Verfahren von Euklid. Es funktioniert, indem man so lange iterativ die jeweils kleinere der beiden Zahlen von der größeren abzieht, bis beide Zahlen gleich groß sind, d.h. dem ggT der Ausgangszahlen entsprechen.

Formuliere das Verfahren in Pseudocode, falls du schon weißt wie, als Flussdiagramm.

Verfeinere nun dein Programm, indem du dabei die Mächtigkeit von Java beachtest, z.B. kann man in Java nicht in einem Schritt zwei Werte vertauschen.

4.4. 99 Bottles

Ein berühmtes US-Amerikanisches Lied um Kinder auf längeren Autofahrten zu beschäftigen beginnt etwa so:

99 bottles of beer on the wall 99 bottles of beer! If one of those bottles should
happen to fall 98 bottles of beer!
98 bottles of beer on the wall 98 bottles of beer...

Da wir heute noch besseres zu tun haben, als das ganze Lied zu rezitieren, lassen wir das doch einfach von unseren Rechnern machen, wenn auch nur als Text: Schreibe ein Programm, das den gesamten Liedtext aller 100 Strophen ausgibt. Die letzte Strophe soll mit "Nobottlesofbeer . . ." beginnen.

4.5. * Zahlenspiel Zack

Beim Spiel Zack wird im Kreis reihum weiter gezählt. Der Spieler sagt entweder die nächste Zahl oder zack falls die Zahl durch sieben teilbar ist. Schreibe ein Programm, das die Zahlen von 1 bis 100 ausgibt, statt Zahlen, die durch 7 teilbar sind soll der Text `zack` erscheinen. Varianten für Fortgeschrittene:

- a) Es soll auch `ßack`erscheinen wenn die Ziffer sieben in der Zahl vorkommt.
- b) Es soll auch `ßack`erscheinen wenn die Quersumme der Zahl sieben ist.

4.6. Teilbarkeit

Schreibe ein Programm, das den Benutzer auffordert eine Ganzzahl einzugeben und dann die Teilbarkeit durch 3 prüft und entsprechende etwas ausgibt (ist teilbar/nicht teilbar). Ändere das Programm anschließend ab, so dass eine `for`-Schleife verwendet wird, um die Teilbarkeit durch 3, 5, 7 und 9 zu prüfen.

4.7. Kurstabelle für Euro und Dollar

Es soll ein einfaches Java-Programm erstellt werden, das eine Kurstabelle ausgibt. Dabei sollst du in 10-Euro Schritten die Euro- und zugehörigen Dollar-Beträge von 10 bis 100 Euro erzeugen und anzeigen. Nimm als Umrechnungskurs den Faktor 1.6.

4.8. Primzahltest

- a) Schreibe ein Programm welches testet ob es sich bei einer gegebenen Zahl um eine Primzahl handelt oder nicht. Die Zahl soll von der Eingabe gelesen werden.
- b) Schreibe dein Programm so um, dass es alle Primzahlen zwischen 0 und 100 ausgibt.