technische universität
dortmund

Fakultät für Informatik
Lehrstuhl 14
Lars Hildebrand, Marcel Preuß,
Iman Kamehkhosh, Alexander Lochmann

Praktikum zu

Einführung in die Informatik für LogWIIngs und WiMas

Wintersemester 2016/17

Übungsblatt 6

Besprechung:
5.12.–9.12.2016
(KW 49)

Vorbereitende Aufgaben

Aufgabe 6.1: Codeformatierung

Für Programmierer ist die richtige Formatierung von Programmcode sehr wichtig. Lesbarer Programmcode erleichtert die Fehlersuche und macht Code leichter erweiterbar.

Formatieren Sie den folgenden Code so, dass er leichter lesbar ist.

a)
```
if(a==7){System.out.println("a ist sieben");}
```

b)
```
int b=12;for(int i=2;i<12;i++
){b=b*i;}System.out.println(b);
```

c)
```
int a=3;if(a<5){int p=7;while(p>0){System.out.println(p);p--;}}
```

d)
```
for(int i=0;i<23;i++){if(i%2==0){for
(int j=1;j<i;j++){System.out.println(
i+";"+j);}}else{System.out.println(i);}}
```

Präsenzaufgaben

Aufgabe 6.2: Umwandlung von Schleifen

In dieser Aufgabe sollen Sie Schleifen von einem Typ in einen anderen umwandeln.

a) Wandeln Sie die folgende **for**-Schleife in eine **while**-Schleife um.

```
int a = 0;
for(int b = 0; b < 17; b++) {
    a = a + 2;
}
System.out.println("a: " + a);
```

b) Wandeln Sie die folgende **do-while**-Schleife in eine **while**-Schleife um.

```
int o = 7;
int p = 256;
do {
    p = p - 20;
    o--;
} while(o > 0);
System.out.println("p :" + p);
```

c) Wandeln Sie die folgende **while**-Schleife in eine **for**-Schleife um.

```
int b = 0;
int n = 7;
int e = 1;
while(b < 15 && n < 9) {
    e = e + b * n - 2;
    b = b + 2;
}
System.out.println("e: " + e);
```

Aufgabe 6.3: Unterfunktionen und Subroutinen

In dieser Aufgabe sollen Sie sich mit der Deklaration und dem Aufruf statischer Methoden vertraut machen.

Bei der Konstruktion eines Geschwindigkeitsmessers für Automobile soll ein Programm die derzeitige Geschwindigkeit in Kilometern pro Stunde (km/h) ausgeben. Die Sensoren geben allerdings die zurückgelegte Strecke in Metern und die dabei vergangene Zeit in Sekunden an.

a) Da das Umrechnen von Geschwindigkeiten eine sehr allgemeine Aufgabe ist, die an vielen Stellen nützlich sein kann, wollen wir eine Funktion schreiben, die diese Umrechnung für uns übernimmt.

- Überlegen Sie sich, wie man eine Geschwindigkeit von m/s in km/h umrechnet.

- Legen Sie eine neue Klasse **SpeedSensor** im Paket **blatt06** an.
- Fügen Sie die **main**-Methode ein, in der Sie zwei Variablen vom Typ **double** anlegen: **currentMeters** und **currentSeconds**. Diese sollen die Sensorwerte repräsentieren. Denken Sie sich beliebige Werte dafür aus.
- Legen Sie innerhalb der SpeedSensor-Klasse eine statische Methode an:
  `public static double velocity`
  Diese soll zwei Parameter vom Typ **double** haben: **meters** und **seconds**.
- Bestimmen Sie anschließend die **Geschwindigkeit** in km/h und geben Sie diese mit Hilfe des **return**-Statements zurück.
- Geben Sie in der **main**-Methode mit folgendem Code die derzeitige Geschwindigkeit aus:

```
System.out.println("Current speed is: " +
            velocity(currentMeters, currentSeconds) + " km/h");
```

b) Jetzt wollen wir dem Fahrer auch noch den Bremsweg anzeigen. Sei v die aktuelle Geschwindigkeit in km/h, dann ist eine Näherung für den Bremsweg s in Metern:

$$s = \frac{1}{2}\left(\frac{v}{10}\right)^2$$

- Schreiben Sie eine Funktion **brakingDistance** vom Typ **double**, die die Sensorwerte **meters** und **seconds** als Parameter nimmt und den Bremsweg in Metern zurückgibt.
- Geben Sie den Bremsweg in der main-Funktion aus mit

```
System.out.println("Current braking distance is: " +
            brakingDistance(currentMeters, currentSeconds) + " m");
```

Aufgabe 6.4: Aufrufverhalten

In dieser Aufgabe sollen Sie sich mit dem Aufruf von statischen Methoden mit primitiven Datentypen als Parameter vertraut machen.

Betrachten Sie folgendes Programm. Was erwarten Sie als Ausgabe?

```
1  package blatt06;
2
3  public class Program {
4      public static void add5ToInt(int x) {
5          x = x + 5;
6          System.out.println("x in function add5ToInt: " + x);
7      }
8
9      public static void main(String[] args) {
10         int x = 8;
11
12         System.out.println("x before function call: " + x);
13         add5ToInt(x);
14         System.out.println("x after function call: " + x);
15     }
16 }
```

- Was ist die Ausgabe des Programms nach Ausführung?

- Was fällt Ihnen auf? Was ist der Grund für dieses Programmverhalten?

Ergänzende Aufgaben

Aufgabe 6.5: Programmstrukturierung

In dieser Aufgabe sollen Sie sich mit der Möglichkeit befassen, ein Programm durch Umstrukturierung in Subroutinen verständlicher zu machen.

Wir wollen ein Programm schreiben, das entscheidet, welcher von 3 Quadern das größte Volumen besitzt.

- Markieren Sie zuerst, in welchem Teil des Programmes sich komplexe, sich wiederholende Strukturen befinden.
- Lagern Sie diese Strukturen in Subroutinen aus, indem Sie das Programm umschreiben. Legen Sie dazu eine neue Klasse mit dem Namen **BiggestVolume** im Paket **blatt06** an.

```
1  package blatt06;
2
3  public class BiggestVolume {
4      public static void main(String[] args) {
5          double firstHeight = 16.5, firstWidth = 27.5, firstDepth = 38.0;
6          double secondHeight = 20.0, secondWidth = 20.0, secondDepth = 20.0;
7          double thirdHeight = 40.2, thirdWidth = 22.5, thirdDepth = 18.5;
8
9          if((firstHeight * firstWidth * firstDepth >=
10                 secondHeight * secondWidth * secondDepth) &&
11                 (firstHeight * firstWidth * firstDepth >=
12                 thirdHeight * thirdWidth * thirdDepth)) {
13             System.out.println("Volume number 1 is the biggest");
14         }
15         else if((secondHeight * secondWidth * secondDepth >=
16                 firstHeight * firstWidth * firstDepth) &&
17                 (secondHeight * secondWidth * secondDepth >=
18                 thirdHeight * thirdWidth * thirdDepth)) {
19             System.out.println("Volume number 2 is the biggest");
20         }
21         else {
22             System.out.println("Volume number 3 is the biggest");
23         }
24     }
25 }
```