



Praktikum zu
**Einführung in die Informatik für
LogWings, WiMas und MedPhys**
Wintersemester 2024/25

Übungsblatt 6
Besprechung:
02.12–06.12.2024
(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. Dafür gibt es in Java Programmier-Konventionen¹ Formatieren Sie den folgenden Code, sodass er leichter zu lesen 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);`

¹<https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>.

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 wollen wir 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: Funktionsköpfe

Nun wollen wir den Umgang mit Funktionen üben. Schreiben Sie Funktionsköpfe für die folgenden Funktionen. Die Funktionsrumpfe oder eine konkrete Implementierung der Funktionen werden dabei nicht benötigt.

a) die Funktion `addFive` soll eine `int`-Variable `x` mit 5 addieren und die Summe zurückgeben

b) die Funktion `mult` soll zwei Zahlen miteinander multiplizieren und das Produkt zurückgeben

c) eine Funktion, die eine Potenz berechnet und das Ergebnis zurückgibt

d) eine Funktion, die die Summe dreier Zahlen ausgibt

e) eine Funktion, die das erste Zeichen eines Wortes zurückgibt

f) eine Funktion, die die Gleichheit zweier Zahlen überprüft und das Ergebnis zurückgibt

g) eine Funktion, die das Maximum zweier Zahlen zurückgibt

h) eine Funktion, die das Maximum zweier Zahlen ausgibt

i) eine Funktion, die eine Dezimalzahl als Binärzahl ausgibt

Aufgabe 6.4: Funktionen

a) Erstellen Sie eine Klasse mit dem Namen **Cuboid** und schreiben Sie eine **main**-Methode in ihr. In der main-Methode sollen die Daten eines Quaders gespeichert werden, wofür wir drei **int**-Variablen für Länge, Breite und Höhe benötigen. Legen Sie daher drei Variablen an und weisen Sie ihnen Werte zu.

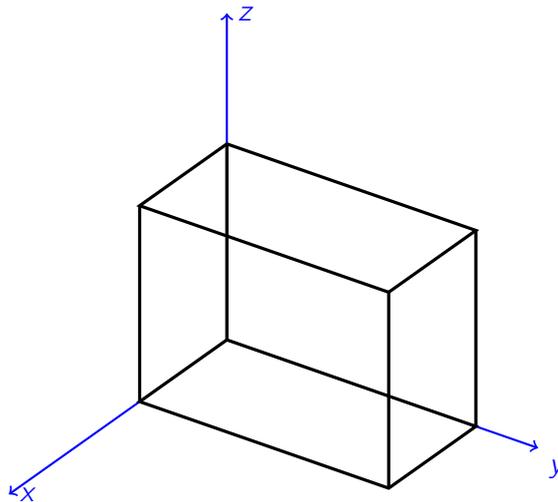


Abbildung 1: Darstellung eines Quaders

Ergänzende Aufgaben

Aufgabe 6.6: Programmstrukturierung

In dieser Aufgabe wollen wir uns mit der Möglichkeit befassen, ein Programm durch Umstrukturierung in Funktionen verständlicher zu machen. Daher wollen wir ein Programm schreiben, das entscheidet, welcher von drei unterschiedlich großen Quadern das größte Volumen besitzt.

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

```
1 public class BiggestVolume {
2     public static void main(String[] args) {
3         double height1 = 16.5, width1 = 27.5, depth1 = 38.0;
4         double height2 = 20.0, width2 = 20.0, depth2 = 20.0;
5         double height3 = 40.2, width3 = 22.5, depth3 = 18.5;
6
7         if ((height1 * width1 * depth1 >= height2 * width2 * depth2)
8             && (height1 * width1 * depth1 >= height3 * width3 * depth3)) {
9             System.out.println("Volume number 1 is the biggest");
10        } else if ((height2 * width2 * depth2 >= height1 * width1 * depth1)
11                && (height2 * width2 * depth2 >= height3 * width3 * depth3)) {
12            System.out.println("Volume number 2 is the biggest");
13        } else {
14            System.out.println("Volume number 3 is the biggest");
15        }
16    }
17 }
```