



# Programmierung für Mathematik (HS13)

---

## Übung 3

### 1 Aufgabe: Begriffe der Objektorientierung

#### 1.1 Lernziele

1. Die wichtigsten Begriffe der objektorientierten Programmierung definieren und zuordnen können.

#### 1.2 Aufgabenstellung

##### a) Begriffe zuordnen

Ordnen Sie die Begriffe den jeweiligen Nummern aus dem untenstehenden Java-Code in Listing 1 hinzu.

- (a) Parameter
- (b) Leerer Rückgabewert
- (c) Klassenname
- (d) Condition (Bedingung)
- (e) Deklaration des Rückgabewertes
- (f) Methodenkommentar (javadoc)
- (g) Klassenkommentar (javadoc)
- (h) Instanzvariable
- (i) Initialisierung
- (j) Rückgabewert der Methode

```

1  /**
2  * Defines the speed of the car.
3  *
4  */
5  public class Car {
6
7      private int speed;
8
9      /**
10     * Sets the speed of the car.
11     * @param speed
12     */
13     public void setSpeed(int speed) {
14         if ( speed < 0 ) {
15             System.err.println("Invalid speed.");
16         } else {
17             this.speed = speed;
18         }
19     }
20
21     /**
22     * Gets the speed of the car.
23     * @return
24     */
25     public int getSpeed() {
26         return this.speed;
27     }
28 }

```

Listing 1: Nummern der Objektorientierung hier zuordnen

**b) Begriffe definieren**

Erklären Sie die folgenden Begriffe in ein bis zwei Sätzen:

- Methodenkommentar
- Klassenname
- Methodename
- Parameter
- Deklaration des Rückgabewertes

## 2 Aufgabe: Code verstehen

### 2.1 Lernziele

1. Präzedenz verstehen

### 2.2 Aufgabenstellung

#### a) Präzedenz

Im Folgenden finden Sie verschiedene Codeausschnitte. Finden Sie heraus, welchen Wert die Variablen am Ende haben. Kontrollieren Sie das Ergebnis anschliessend mit Java.

```
1 int a=3, b=12;  
2 a+++b;
```

Listing 1: Snippet 1

```
1 int a=1, b=4;  
2 b+a+++b;
```

Listing 2: Snippet 2

```
1 int a=1, b=3;  
2 b+++a+b*b;
```

Listing 3: Snippet 3

```
1 int a=1, b=3;  
2 b+++*--a;
```

Listing 4: Snippet 4

```
1 int a=4, b=8;  
2 a+++b*a/--b+b*a-b--;
```

Listing 5: Snippet 5

## 3 Aufgabe: Klassen und Methoden schreiben

### 3.1 Lernziele

1. Klassen definieren und schreiben können
2. Methoden auf Objekten dieser Klassen ausführen können

### 3.2 Aufgabenstellung

Schreiben Sie eine Klasse `Vector` mit drei Dimensionen. Dazu müssen drei *floats* gespeichert werden. Schreiben Sie ausserdem einen Testdriver um die Funktionalität zu testen. Erweitern Sie die `Vector` Klasse anschliessend mit den folgenden Methoden:

- Einen Konstruktoren, um die drei Werte direkt einem Vektoren zuzuordnen.
- Getter-Methoden, um auf die Werte ausserhalb der Klasse zuzugreifen.
- Eine `add`-Methode, welche einen `Vector` als Input nimmt (Methoden-Parameter) und diesen mit dem intern gespeicherten Vektor addiert. Der neue Vektor (die Summe) soll anschliessend als `Vector` zurückgegeben werden.
- Eine `subtract`-Methode, welche einen `Vector` als Input nimmt und diesen vom intern gespeicherten Vektor subtrahiert. Das Resultat wird wiederum als `Vector` zurückgegeben.
- Eine Methode `scalarProduct`, die das Skalar-Produkt berechnet. Diese nimmt ebenfalls einen `Vector` als Input, berechnet das Skalarprodukt und returniert das Resultat als `float`.
- Eine `Norm`-Methode, die die Euklidische Norm des Vektors returnieren soll. Um diese Teilaufgabe zu lösen, brauchen Sie die Klasse `Math`. Lesen Sie in der API<sup>1</sup> nach, welche Methoden diese Klasse anbietet.
- Eine `angle`-Methode, die den Winkel zwischen dem gespeicherten Vektoren und einem als Input angenommenen Vektoren berechnet und als `float` ausgibt. Falls Sie mathematische Methoden benötigen, finden Sie diese wiederum in der `Math`-Klasse.
- Schreiben Sie zum Schluss eine `print`-Methode, die den aktuellen Vektoren ausgibt.

---

<sup>1</sup><http://docs.oracle.com/javase/6/docs/api/java/lang/Math.html>