

# Klassen, Objektorientierung, Information Hiding

Beispielklasse Artikel:

```
public class Artikel {

    private String name;
    private String kennnummer;
    private int stueckzahl = 0;
    private float preis;

    public Artikel(String name, String kennnummer, float preis) {
        this.name = name;
        this.kennnummer = kennnummer;
        this.preis = preis;
    }

    public String getKennnummer() {
        return kennnummer;
    }

    public void setPreis(float preis) {
        this.preis = preis;
    }

    public void zukaufen(int stueckzahl) {
        this.stueckzahl += stueckzahl;
    }

}
```

Verwendung der Klasse Artikel:

```
public class MyProgram {

    public static void main(String[] args) {
        Artikel mixer = new Artikel("Mixer", "A4711", 74.99f);
        mixer.zukaufen(100);
        mixer.setPreis(72.99f);
        System.out.println("Kennnummer: "+mixer.getKennnummer());
    }

}
```

### **Bsp 1: Klasse Point**

Schreiben Sie eine Klasse `Point`, die folgende Methoden beinhaltet:

```
public Point();
public Point(double x, double y);
public void setLocation(double x, double y);
public void move(double dx, double dy);
public double getX();
public double getY();
```

### **Bsp 2: Public, Private, Protected**

Was bedeuten die Schlüsselworte `public`, `private` und `protected` vor Methoden und Variablen?

### **Bsp 3: Konstruktoren**

- a) Wie kann eine Klasse, in der kein Konstruktor definiert wurde, instanziiert werden?
- b) Kann eine Klasse, in der ausschließlich der Konstruktor `public Punkt(double x, double y)` definiert wurde mit `new Punkt()` instanziiert werden?
- c) Kann ein Konstruktor auch als `protected` oder `private` deklariert werden?

### **Bsp 4: Klassenelement vs. Instanzelement**

Schreiben Sie eine beliebige Klasse, in der folgende Elemente vorhanden sind: Klassenmethode, Klassenvariable, Instanzmethode, Instanzvariable. Verwenden Sie die Elemente aus einem Testprogramm heraus.

- a) Was ist der Unterschied zwischen den Klassenelementen und den Instanzelementen?
- b) Kann ein Instanzelement auf ein Klassenelement zugreifen?
- c) Kann ein Klassenelement auf ein Instanzelement zugreifen?

### **Bsp 5: Methoden aufrufen**

Gegeben ist folgende Klasse:

```
public class KlasseA {

    public static void m1Static() {...};
    private static void m2Static() {...};

    public void m1() {...};
    private void m2() {...};

}
```

Welche der folgenden Methodenaufrufe sind erlaubt:

```
public class KlasseB {

    public static void main(String[] args) {
        KlasseA.m1Static();
        KlasseA.m2Static();
        KlasseA.m1();
        KlasseA.m2();

        KlasseA a = new KlasseA();
        a.m1Static();
        a.m2Static();
        a.m1();
        a.m2();
    }

}
```

### **Bsp 6: Wörterbuch**

Schreiben Sie ein Wörterbuch, das zumindest folgende Methoden beinhaltet:

```
public class Dictionary {
    public boolean insert(String wortSprache1, String wortSprache2) {...}
    public void remove(String wortSprache1) {...}
    public String lookup(String wortSprache1) {...}
}
```

Die Methode `insert` soll ein Wortepaar in das Wörterbuch einfügen. Wurde das Wort korrekt aufgenommen, so soll `true` zurückgegeben werden, ist kein Platz mehr im Wörterbuch oder wurde das Wort bereits aufgenommen, so soll `false` geliefert werden. Die Methode `remove` soll das zum Parameter gehörige Wortepaar entfernen. Die Methode `lookup` soll nach einem Wort suchen und die Übersetzung des Wortes zurückliefern. Wird das Wort nicht gefunden, so soll `null` geliefert werden. Schreiben Sie eine weitere Klasse `Translation`, um die Wortepaare im Wörterbuch darzustellen.

### **Bsp 7: Rechtecke schneiden**

Definieren Sie eine Klasse `Rectangle` mit sinnvollen Methoden und Konstruktor(en). Schreiben Sie auch eine Methode `public Rectangle intersect(Rectangle rect)`, welche das übergebene Rechteck mit dem aufgerufenen Rechteck schneidet und als Ergebnis das Schnittrechteck zurückgibt. Gibt es keine Schnittfläche, so soll null zurückgegeben werden.

### **Bsp 8: Math.min(...)**

In der Mathematik-Klasse `Math` des Javasystems gibt es mehrere Implementierungen für die min-Methode. Warum ist dies so? Wie wird dies genannt (Fachausdruck)?

### **Bsp 9: Artikelverwaltung**

Schreiben Sie ein Programm, das eine Verkaufszahlendatei einliest:

102700	999	1	3	1	2	4	0
102701	3250	2	13	4	1	1	0
102702	1190	2	1				0

Die erste Nummer ist die Artikelnummer, dann folgt der Stückpreis des Artikels und eine beliebige Anzahl von verkauften Stück (Ganzzahlen größer Null). Ein Datensatz wird durch eine Null abgeschlossen.

Entwickeln Sie ein Programm, das eine tabellarische Auflistung von Artikelnummer und Umsatz (= Artikelpreis \* Summe der verkauften Mengen) ausgibt. Definieren Sie dazu eine Klasse `Artikel`, die alle notwendigen Daten eines Artikeldatensatzes hält. Schreiben Sie weiters eine Methode `Artikel readArtikel()` und `void writeArtikel(Artikel article)`, um einen Datensatz aus der Datei zu lesen bzw. eine Zeile der Ausgabetabelle darzustellen.