



Praktikum zu  
**Einführung in die Informatik für  
LogWings, WiMas und MedPhys**  
Wintersemester 2021/22

**Übungsblatt 11**

Besprechung:  
24.–28.01.2022  
(KW 4)

**Präsenzaufgaben**

**Aufgabe 11.1:** Vererbung: Einstieg

Gegeben sind folgende Klassen:

```
1 public class Person {
2     private String firstname;
3     private String surname;
4
5     public Person(String firstname, String surname) {
6         this.firstname = firstname;
7         this.surname = surname;
8     }
9
10    public String toString() {
11        return this.firstname + " " + this.surname;
12    }
13 }
```

```
1 public class Student extends Person {
2     private int matrnr;
3
4     public Student(String firstname, String surname, int matrnr) {
5         super(firstname, surname);
6         this.matrnr = matrnr;
7     }
8
9     public String toString() {
10        String name = super.toString();
11        return this.matrnr + " " + name;
12    }
13 }
```

```
1 public class Employee extends Person {
2     private String chair;
3     private double salary;
4
5     public Employee(String firstname, String surname, String chair,
6         double salary) {
7         super(firstname, surname);
8         this.chair = chair;
```

```
9     this.salary = salary;
10 }
11
12 public String toString() {
13     String name = super.toString();
14     return "Name: " + name + ", Chair: " + this.chair
15         + ", Salary: " + this.salary + " Euro per hour";
16 }
17 }
```

Welche Ausgabe hat folgendes Programm? Testen Sie das Programm **nicht**, indem Sie es abtippen!

```
1 public class UniTest {
2     public static void main(String[] args) {
3         Person visitor = new Person("Max", "Mustermann");
4         System.out.println(visitor.toString());
5
6         Student junior = new Student("Karl", "Karlson", 123456);
7         System.out.println(junior.toString());
8
9         Employee scientist = new Employee("Markus", "Mueller",
10             "Software Engineering", 11.0);
11         System.out.println(scientist.toString());
12
13         Person senior = new Student("Mark", "Mustermann", 1248);
14         System.out.println(senior.toString());
15
16         Person admin = new Employee("Egon", "Schneider", "Databases", 13.5);
17         System.out.println(admin.toString());
18     }
19 }
```

---

---

---

---

---

---

### Aufgabe 11.2: Vererbung: Quizfragen

In dieser Aufgabe sollen Sie sich mit dem Konzept der Vererbung beschäftigen.

- a) Mit welchem Schlüsselwort kann man auf das aktuelle Objekt zugreifen? Z. B. um auf dessen Attribute zuzugreifen, wenn sie von einer lokalen Variable überdeckt werden.

---

- b) Welches Schlüsselwort wird verwendet um in der Klassendeklaration das Erben von einer anderen Klasse zu kennzeichnen?

---

- c) Welche Methoden und Attribute sind innerhalb einer Unterklasse von der Oberklasse sichtbar?

---

- d) Mit welchem Schlüsselwort können Sie (unter Umständen überschriebene) Methoden der Oberklasse aufrufen?

---

- e) Eine Klasse **BachelorStudent** erbt von der Klasse **Student**. Ist die Zuweisung `Student max = new BachelorStudent("Max", "Mustermann");` gültig?

(Unter der Annahme, dass der Konstruktor korrekt aufgerufen wird)

---

- f) Ist entsprechend eine Zuweisung `BachelorStudent maria = new Student("Maria", "Musterfrau");` gültig?

(Unter der Annahme, dass der Konstruktor korrekt aufgerufen wird)

---

### Aufgabe 11.3: Vererbung: Erste Anwendung

In dieser Aufgabe wollen wir eine Unterklasse schreiben und verwenden. Sie benötigen hierzu die Klassen **Vehicle** und **Car** von Blatt 9 oder 10. Diese besitzen eine offensichtliche „*ist ein*“-Eigenschaft zueinander. Kopieren Sie die Quellcode-Dateien der beiden Klassen und ändern Sie die Klasse **Car** so ab, dass sie von **Vehicle** erbt. Der Konstruktor von **Car** ist nun unvollständig. Erweitern Sie den Konstruktor mithilfe des Schlüsselwortes **super** so, dass neu erzeugte Autos Vehikel mit **vier** Reifen sind, die **Benzin** als Treibstoff verwenden.

#### Aufgabe 11.4: Vererbung: Methoden überschreiben

In dieser Aufgabe wollen wir Methoden überschreiben, um so ihre Funktionalität zu erweitern.

- a) Überschreiben Sie die **toString**-Methode der **Vehicle**-Klasse, sodass der Text, den Sie bisher in der **print**-Methode ausgeben, nun als Zeichenkette zurückgegeben wird.
- b) Überschreiben Sie die **toString**-Methode der **Car**-Klasse, sodass der Text der **toString**-Methode der Oberklasse, **Vehicle**, um eine neue Zeile (**\n**) und dem Text, den Sie bisher in der **print**-Methode gegeben haben, zurückgegeben wird.
- c) Schreiben Sie eine Testklasse mit einer **main**-Methode, in der Sie verschiedene Objekte vom Typ **Vehicle** und **Car** erstellen und direkt an die Funktion **System.out.println()** übergeben.

#### Aufgabe 11.5: Vererbung: abstrakte Klassen

In dieser Aufgabe wollen wir uns mit dem Konzept abstrakter Klassen beschäftigen:

- a) Wenn Sie die Klasse **Vehicle** als Abstrakt deklarieren wollen würden, wie müsste dann die Deklaration der Klasse aussehen?

---

- b) Vehikel sollen nun grundsätzlich eine **getPower()**-Methode, wie die **Car**-Klasse, haben. Die Unterklassen sollen gezwungen werden eine Implementierung anzugeben. Wie sieht die Deklaration dieser Methode aus?

---

- c) Können Sie nach diesen Änderungen noch Objekte vom Typ **Vehicle** instanziiieren?

---

---

- d) Worin läge der Vorteil dieser Änderungen?

---

---

## Ergänzende Aufgaben

### **Aufgabe 11.6:** Umsetzung

Setzen Sie die Änderungen der Aufgabe 11.5 in die Praxis um. Ändern Sie entsprechend auch Ihre Testfälle in der Testklasse.