technische universität
dortmund

Fakultät für Informatik
Lehrstuhl 14
Lars Hildebrand

Praktikum zu

Einführung in die Informatik für LogWings, WiMas und MedPhys

Wintersemester 2024/25

Übungsblatt 11

Besprechung:
20.01.–24.01.2025
(KW 4)

Präsenzaufgaben

Aufgabe 11.1: Vererbung: Einstieg

Gegeben sind folgende Klassen:

```java
public class Person {
    private String firstname;
    private String surname;

    public Person(String firstname, String surname) {
        this.firstname = firstname;
        this.surname = surname;
    }

    public String toString() {
        return this.firstname + " " + this.surname;
    }
}
```

```java
public class Student extends Person {
    private int matrnr;

    public Student(String firstname, String surname, int matrnr) {
        super(firstname, surname);
        this.matrnr = matrnr;
    }

    public String toString() {
        String name = super.toString();
        return this.matrnr + " " + name;
    }
}
```

```java
public class Employee extends Person {
    private String chair;
    private double salary;

    public Employee(String firstname, String surname, String chair,
            double salary) {
        super(firstname, surname);
        this.chair = chair;
```

```
        this.salary = salary;
    }

    public String toString() {
        String name = super.toString();
        return "Name: " + name + ", Chair: " + this.chair
            + ", Salary: " + this.salary + " Euro per hour";
    }
}
```

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

```
public class UniTest {
    public static void main(String[] args) {
        Person visitor = new Person("Max", "Mustermann");
        System.out.println(visitor.toString());

        Student junior = new Student("Karl", "Karlson", 123456);
        System.out.println(junior.toString());

        Employee scientist = new Employee("Markus", "Mueller",
                "Software Engineering", 11.0);
        System.out.println(scientist.toString());

        Person senior = new Student("Mark", "Mustermann", 1248);
        System.out.println(senior.toString());

        Person admin = new Employee("Egon", "Schneider", "Databases", 13.5);
        System.out.println(admin.toString());
    }
}
```

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**, und der 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** instanziieren?

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.