Prof. Dr. A. Weber
Dr. Hassan Errami

Universität Bonn
Institut für Informatik II
01.05.2019

Sommersemester 2019

Übungen zur Vorlesung
Objektorientierte Softwareentwicklung (BA-INF-024)
Aufgabenblatt 4 (20 Punkte)

Zu bearbeiten bis: 10.05.2019

Aufgabe 1 *(Ausgaben erklären - 3*2=6 Punkte)*

Gegeben seien die Klassen *Auto* und *Lastwagen*:

```
public class Auto {
    int reifen = 4;
    int getReifen() {
        return this.reifen;
    }
}

public class Lastwagen extends Auto {
    int reifen = 8;
    int getReifen() {
        return this.reifen;
    }
}
```

a) Legen Sie beide Klasse in einem neuen Projekt an. Legen Sie eine weitere Klasse an und fügen Sie das nachfolgende Codefragment in die main-Methode ein:

```
Auto auto1 = new Lastwagen();
System.out.println(auto1.reifen); // Ausgabe 1)
System.out.println(((Lastwagen) auto1).reifen);  // Ausgabe 2)
System.out.println(auto1.getReifen()); // Ausgabe 3)
System.out.println(((Lastwagen) auto1).getReifen()); // Ausgabe 4)
```

Bestimmen Sie die jeweiligen Ausgaben und begründen Sie, warum diese zustande kommen.

b) Entfernen Sie nun die *getReifen()*-Methode aus der Klasse *Lastwagen*. Führen Sie Ihre Testklasse erneut aus. Haben sich einige Ausgaben verändert? Falls ja, weshalb?

c) Fügen Sie jetzt die *getReifen()*-Methode wieder in die Klasse *Lastwagen* ein. Entfernen Sie sie diesmal aus der *Auto*-Klasse. Betrachten Sie erneut das Fragment in ihrer Testklasse. Was hat sich verändert und wieso?

Aufgabe 2 *(Überladen von Methoden - 3*2 = 6 Punkte)*

Betrachten folgende Interfaces und Klassen:

```
interface Food {
    public String getMeal();
}

class Cauliflower implements Food {
    public String getMeal() {
        return "Blumenkohl an Gorgonzolasoße";
    }
}

class Spaghetti implements Food {
    public String getMeal() {
```

```
        return "Spaghetti Bologneser Art";
    }
}
```

Die zweite Klasse *Student*:

```
class Student {
    public void eat(Food food) {
        System.out.println("I like " + food.getMeal());
    }
    public void eat(Cauliflower Cauliflower) {
        System.out.println("I don't like " + Cauliflower.getMeal());
    }
}
```

a) Welche Ausgabe erzeugt der folgende Aufruf? Warum?

```
Cauliflower essen1 = new Cauliflower();
student.eat(essen1);
```

b) Welche Ausgabe erzeugt der folgende Aufruf? Warum?

```
Spaghetti essen2 = new Spaghetti();
student.eat(essen2);
```

c) Welche Ausgabe erzeugt der folgende Aufruf? Warum?

```
Food essen = new Cauliflower();
student.eat(essen);
```

Aufgabe 3 *(Comparable - 4+4=8 Punkte)*

Das Interface Comparable (java.lang.Comparable) dient dazu, zwei Objekte miteinander zu vergleichen. Das Interface besteht aus nur einer Methode, int compareTo(Object o). Diese vergleicht das jeweilige Objekt mit einem Weiteren. Die Methode liefert einen negativen Wert, wenn das Objekt kleiner ist, als das Übergebene. Ist es größer, liefert die Methode einen positiven Wert. Sind die beiden Objekte gleich, wird 0 zurückgegeben. Schreiben Sie eine Klasse Person mit folgenden Attributen:

- Name
- Vorname
- Postleitzahl
- Straße
- Hausnummer

Diese Klasse soll das Interface Comparable implementieren. Dazu vergleicht sie die oben genannten Attribute in der angegebenen Reihenfolge. Das heißt, als erstes werden die Namen verglichen. Sind die Namen gleich, werden die Vornamen verglichen usw.
a) Erstellen Sie nun eine weitere Klasse, die ein Klassenmethode public static Comparable findMin(Comparable[] x) zu Verfügung stellt, welche das kleinste Element im Array zurückgibt. Schreiben Sie ein Testprogramm, dass die Methode findMin benutzt, um das Minimum in einem Array von Personen zu finden. Geben Sie das Element aus. Rufen Sie die Methode ein zweites mal auf, aber diesmal mit einem Array von Integer Werten.

Hinweise: Zur Ausgabe des Ergebnisses der Methode findMin, muss die Rückgabe zu dem jeweiligen Typ gecastet werden. Der Datentyp String implementiert auch das Interface Comparable. In der Dokumentation ist das Interface mit Comparable<T> angegeben. Da generische Datentypen noch nicht in der Vorlesung erklärt wurden, lassen Sie diesen einfach weg. Die Warnung in Eclipse, dass der Raw-Type genutzt wird, können Sie ignorieren.