

Praktikum zu

**Einführung in die Informatik für
LogWilngs und WiMas**

Wintersemester 2016/17

Übungsblatt 10

Besprechung:

23.01.–

27.01.2017

(KW 4)

Vorbereitende Aufgaben

Aufgabe 10.1: Wiederholung: Klammern

Geben Sie an, wofür folgende Klammern in Java verwendet werden.

- [...]

- (...)

- { ... }

Aufgabe 10.2: Wiederholung: Funktionsparameter

Geben Sie einen geeigneten Methodenkopf für die folgenden öffentlichen, statischen Funktionen an.

- Eine Funktion **average**, die den Durchschnitt eines **double**-Arrays berechnet.
Beispielantwort: `public static double average(double[] array)`
- Eine Funktion **plus**, die zwei rationale Zahlen miteinander addiert und die Summe zurückgibt.

- Eine Funktion **countWords**, die die Wörter in einem **String** zählt und zurückgibt.

- Eine Funktion **printMaximum**, die das Maximum eines **int**-Arrays mit `System.out.println` auf dem Bildschirm ausgibt.

- Eine Funktion **times**, die einen Integer **n** und einen Integer **x** entgegen nimmt und ein **n** Elemente langes Array, gefüllt mit dem Wert **x** zurückgibt.

Präsenzaufgaben

Aufgabe 10.3: Objektvariablen und -methoden vs. Klassenvariablen und -methoden

In dieser Aufgabe sollen Sie sich mit der unterschiedlichen Verwendung von Objekt- und Klassen-
elementen vertraut machen.

Manche Zuweisungen und Methodenaufrufe sind im unteren Programm nicht erlaubt (vgl. dazu
Folien 39–42 in Kapitel 6). Notieren Sie auf den Linien neben dem Programmtext, ob die jeweilige
Zuweisung oder der jeweilige Methodenaufruf erlaubt ist oder nicht.

```
1  class Tester {  
2      int var1;  
3      static int var2;  
4  
5      void test1() {  
6          var1++;  
7          var2--;  
8      }  
9  
10     static void test2() {  
11         var1++;  
12         var2--;  
13     }  
14  
15     public static void main(String[] args) {  
16         var1 = 1;  
17         var2 = 1;  
18         test1();  
19         test2();  
20  
21         Tester testObjekt = new Tester();  
22         testObjekt.var1 = 2;  
23         testObjekt.var2 = 2;  
24         testObjekt.test1();  
25         testObjekt.test2();  
26  
27         Tester.var1 = 3;  
28         Tester.var2 = 3;  
29         Tester.test1();  
30         Tester.test2();  
31     }  
32 }
```

Aufgabe 10.4: Vererbung: Einstieg

Gegeben sind folgende Klassen:

```
1 package blatt10;
2
3 public class Person {
4     private String firstname;
5     private String surname;
6
7     public Person(String firstname, String surname) {
8         this.firstname = firstname;
9         this.surname = surname;
10    }
11
12    public String toString() {
13        return this.firstname + " " + this.surname;
14    }
15 }
```

```
1 package blatt10;
2
3 public class Student extends Person {
4     private int matrnr;
5
6     public Student(String firstname, String surname, int matrnr) {
7         super(firstname, surname);
8         this.matrnr = matrnr;
9     }
10
11    public String toString() {
12        String name = super.toString();
13        return this.matrnr + " " + name;
14    }
15 }
```

```
1 package blatt10;
2
3 public class Employee extends Person {
4     private String chair;
5     private double salary;
6
7     public Employee(String firstname, String surname, String chair,
8                     double salary) {
9         super(firstname, surname);
10        this.chair = chair;
11        this.salary = salary;
12    }
13
14    public String toString() {
15        String name = super.toString();
16        return "Name: " + name + ", Chair: " + this.chair +
17            ", Salary: " + this.salary + " Euro per hour";
18    }
19 }
```

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

```
1 package blatt10;
2
3 public class UniTest {
4     public static void main(String[] args) {
5         Person visitor = new Person("Max", "Mustermann");
6         System.out.println(visitor.toString());
7
8         Student junior = new Student("Karl", "Karlson", 123456);
9         System.out.println(junior.toString());
10
11        Employee scientist = new Employee("Markus", "Mueller",
12            "Software Engineering", 11.0);
13        System.out.println(scientist.toString());
14
15        Person senior = new Student("Mark", "Mustermann", 1248);
16        System.out.println(senior.toString());
17
18        Person admin = new Employee("Egon", "Schneider", "Databases", 13.5);
19        System.out.println(admin.toString());
20    }
21 }
```

Aufgabe 10.5: Vererbung: Quizfragen

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

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

2. Welches Schlüsselwort wird verwendet um in der Klassendeklaration das Erben von einer anderen Klasse zu kennzeichnen?

3. Welche Methoden und Attribute sind innerhalb einer Unterklasse von der Oberklasse sichtbar?

4. Mit welchem Schlüsselwort können Sie (unter Umständen überschriebene) Methoden der Oberklasse aufrufen?

5. 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)

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

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

Aufgabe 10.6: Vererbung: Erste Anwendung

In dieser Aufgabe sollen Sie eine Unterklasse schreiben und verwenden. Sie benötigen hierzu eine funktionierende Lösung der Aufgaben aus Blatt 9.

- Auf dem letzten Blatt haben Sie Klassen für Vehikel und Autos geschrieben. Diese besitzen eine offensichtliche „*ist ein*“-Eigenschaft zueinander. Kopieren Sie die Quellcode-Dateien der beiden Klassen in das Paket **blatt10** und ändern Sie die Klasse **Car** so ab, dass sie nun von **Vehicle** erbt!
- Der Konstruktor von **Car** ist nun unvollständig. Erweitern Sie den Konstruktor mit Hilfe des Schlüsselwortes **super** so, dass neu erzeugte Autos Vehikel mit **vier** Reifen sind, die **Benzin** als Treibstoff verwenden.

Aufgabe 10.7: Vererbung: Methoden Überschreiben

In dieser Aufgabe sollen Sie Methoden überschreiben um so ihre Funktionalität zu erweitern.

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

Aufgabe 10.8: Vererbung: abstrakte Klassen

In dieser Aufgabe sollen Sie sich mit dem Konzept abstrakter Klassen auseinandersetzen:

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

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

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

- Worin läge der Vorteil dieser Änderungen?

Ergänzende Aufgaben

Aufgabe 10.9: Umsetzung

Setzen Sie die Änderungen der Aufgabe 10.8 in die Praxis um. Ändern Sie entsprechend auch Ihre Testfälle in der Testklasse.

Aufgabe 10.10: Fehlersuche

Betrachten Sie folgendes Programm, das auf die Klassen aus Aufgabe 10.4 zurückgreift. Welche Fehler haben sich eingeschlichen?

```
1 package blatt10;
2
3 public class WrongCode {
4     public static void main(String[] args) {
5         Person people[] = new Person[3];
6
7         people[0] = new Employee("Robert", "Schneider");
8         System.out.println(people[0].toString());
9
10        people[1] = Person("Sabine", "Meier");
11        System.out.println(people[1].toString());
12
13        Student admin = new Employee("Anja", "Mueller", "Computer Science", 17.0);
14        people[2] = admin;
15        System.out.println(people[2].toString());
16
17        Person theNewOne = new Student("Paul", "Thomas", 113862);
18        people[3] = admin;
19        System.out.println(people[3].toString());
20    }
21 }
```
