

Praktikum zu

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

Wintersemester 2015/16

Übungsblatt 11

Besprechung:

25.–29.01.16

(KW 3)

Vorbereitende Aufgaben

Aufgabe 11.1: 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)

Präsenzaufgaben

Aufgabe 11.2: Objektvariablen und -methoden vs. Klassenvariablen und -methoden

In dieser Aufgabe sollen Sie sich mit der unterschiedlichen Verwendung von Objekt- und Klassenelementen 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 11.3: Objektreferenzen

In dieser Aufgabe sollen Sie sich mit der Handhabung von Referenzen am Beispiel von Strings beschäftigen. Denken Sie daran, dass Strings in Java Objekte und keine primitiven Datentypen sind!

Implementieren Sie die **main**-Methode einer Klasse **ReferenceTest** im Paket **blatt11** wie folgt:

- a) • Deklarieren Sie zwei Variablen **a** und **b** vom Typ **String**.
- Initialisieren Sie beide Strings mit **null**.
- Geben Sie das Ergebnis des Ausdrucks **a == b** aus.

- b)
 - Weisen Sie **a** einen neuen String mit der Anweisung `new String("Hallo")` zu.
 - Geben Sie das Ergebnis des Ausdrucks `a == b` aus.
- c)
 - Weisen Sie **b** einen neuen String mit der Anweisung `new String("Hallo")` zu.
 - Geben Sie das Ergebnis des Ausdrucks `a == b` aus.
 - Geben Sie das Ergebnis des Ausdrucks `a.equals(b)` aus.
- d)
 - Weisen Sie **b** einen neuen String mit der Anweisung `new String("World")` zu.
 - Geben Sie das Ergebnis des Ausdrucks `b.equals(a)` aus.
 - Geben Sie das Ergebnis des Ausdrucks `a.equals(b)` aus. (Symmetrie)
- e)
 - Weisen Sie der Variablen **b** den Wert von **a** zu.
 - Geben Sie das Ergebnis des Ausdrucks `a == b` aus.
 - Geben Sie das Ergebnis des Ausdrucks `a.equals(b)` aus.
- f)
 - Weisen Sie **a** den Wert **null** zu.
 - Geben Sie das Ergebnis des Ausdrucks `b.equals(a)` aus.
 - Geben Sie das Ergebnis des Ausdrucks `b == a` aus.
 - Geben Sie das Ergebnis des Ausdrucks `a == b` aus.
 - Geben Sie das Ergebnis des Ausdrucks `a.equals(b)` aus.

Worin liegt der Unterschied zwischen `equals` und `==`?

Was geschieht bei `a.equals(b)` und wie erklären Sie sich dies?

Aufgabe 11.4: Vererbung: Einstieg

Gegeben sind folgende Klassen:

```
1 package blatt11;
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 blatt11;
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 blatt11;
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?

```
1 package blatt11;
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            "Algorithm 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 11.5: Überladene Methoden

In dieser Aufgabe sollen Sie eine überladene Methode schreiben.

- Übernehmen Sie die Klassen **Cuboid** und **Sphere** aus Blatt 9 oder verwenden Sie die Vorgabe von der EINI-Webseite.
- Schreiben Sie für die Klasse **Cuboid** eine Methode **scale**, die den Quader in allen 3 Achsen proportionserhaltend vergrößert oder verkleinert. Ein $(3 \times 2 \times 4)$ großer Quader, für den `scale(2.5)` aufgerufen wird, sollte z. B. danach $(7.5 \times 5 \times 10)$ groß sein.

Schreiben Sie darüber hinaus eine weitere **scale**-Methode, die drei **double**-Parameter annimmt, mit der Sie die Streckung bzw. Stauchung in 3 Achsen separat angeben können.

- Schreiben Sie analog eine Methode **scale** für die Klasse **Sphere**.

Aufgabe 11.6: Vererbung: Erste Anwendung

In dieser Aufgabe sollen Sie eine Unterklasse schreiben und verwenden.

- Schreiben Sie eine Klasse **Cube**, die einen geometrischen Würfel repräsentiert. Diese soll von **Cuboid** erben. Der Konstruktor von **Cube** hat dabei nur einen Parameter vom Typ **double**: **edgeLength**, die Kantenlänge.
- Legen Sie in einer Testklasse mehrere **Sphere**-, **Cuboid**- und **Cube**-Objekte an und skalieren Sie sie.
- Warum müssen Sie keine separate **scale**-Methode für die Klasse **Cube** schreiben?

-
- Was könnte problematisch an diesem Ansatz sein?
-
-

Aufgabe 11.7: Vererbung: abstrakte Klassen

In dieser Aufgabe sollen Sie die bisher geschriebenen Klassen für geometrische Objekte unter einer gemeinsamen Oberklasse zusammenfassen.

- Schreiben Sie eine abstrakte Klasse mit dem Namen **GeometricObject**, die zwei abstrakte Methoden **getVolume** und **getSurfaceArea** sowie eine nicht abstrakte Methode **compare** deklariert.
- Die Methoden **getVolume** und **getSurfaceArea** sollen von den Unterklassen **Cuboid** und **Sphere** implementiert werden.
- Schreiben Sie für die Klasse **GeometricObject** eine Methode **compare** wie in Aufgabe 9.6, die geometrische Objekte anhand ihres Volumens miteinander vergleicht.
- Lassen Sie anschließend die Klassen **Cuboid** und **Sphere** von der Klasse **GeometricObject** erben.
- Welche Methoden werden nicht mehr gebraucht bzw. müssen nicht mehr implementiert werden?

Ergänzende Aufgaben

Aufgabe 11.8: Fehlersuche

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

```
1 package blatt11;
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", 9.0);
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 }
```
