



Übung zur Vorlesung *Einführung in die Informatik 2 für Ingenieure (MSE)*

Alexander van Renen (renen@in.tum.de)

<http://db.in.tum.de/teaching/ss17/ei2/>

Blatt Nr. 3

Dieses Blatt wird am Montag, den 22. Mai 2016 besprochen.

Aufgabe 1: Overloading

Welche der folgenden Methoden-Überladungen sind erlaubt und welche nicht? Überprüfen Sie Ihre Antworten indem Sie die Beispiele in Java programmieren. Eine unterstrichener Methodenname bedeutet dass es sich um eine Klassenmethode handelt.

a)

Car
-speed : double
+accelerate(mph : double)
+accelerate(kmh : double)

b)

MetricCar
-speed : double
+accelerate(kmh : double)

ImperialCar
-speed : double
+accelerate(mph : double)

c)

Car
-speed : double
+accelerate(kmh : double)
+accelerate(seconds : int)

d)

Car
-speed : double
+accelerate(seconds : int)
+accelerate(seconds : int) : double

e)

Car
-speed : double
+toString() : String
+toString() : String

f)

Car
-speed : double
+accelerate(kmh : double)
+accelerate(mph : double, slope : double)

g)

Car
-speed : double
+load(passengers : Set<Passenger>)
+load(luggage : Set<Suitcase>)

Aufgabe 2: Dynamisches Binden

Überlegen Sie sich welche Methoden aufgerufen werden, wenn man die Klasse `DynamicDispatch` ausführt. Überprüfen Sie anschließend ihre Vermutung, indem Sie das Programm tatsächlich ausführen.

```
1 class DynamicDispatch {
2     public static void main(String [] args) {
3         A a = new A();
4         B b = new B();
5         C c = new C();
6         D d = new D();
7
8         A[] array = {a, b, c, d};
9         for (A element : array) {
10            System.out.println("x():");
11            element.x();
12            System.out.println("\ny():");
13            element.y();
14            System.out.println("\nz():");
15            element.z();
16            System.out.println("\n=====\n");
17        }
18    }
19 }
20
21 class A {
22     public void x() {
23         System.out.println("->A_x()");
24         z();
25     }
26
27     public void y() {
28         System.out.println("->A_y()");
29         this.z();
30     }
31
32     public void z() {
33         System.out.println("->A_z()");
34     }
35 }
36
37 class B extends A {
38     public void y() {
39         System.out.println("->B_y()");
40         x();
41     }
42
43     public void z() {
```

```

44     System.out.println ("->B_z() ");
45 }
46 }
47
48 class C extends B {
49     public void x() {
50         System.out.println ("->C_x() ");
51         z();
52     }
53 }
54
55 class D extends A {
56     public void x() {
57         System.out.println ("->D_x() ");
58         super.x();
59     }
60
61     public void z() {
62         System.out.println ("->D_z() ");
63     }
64 }

```

Aufgabe 3: Don't repeat yourself (DRY)

Auf der Webseite¹ finden Sie die Klasse `Warenhaus.java`.

- Nennen Sie mindestens zwei Gründe, warum die Modellierung nicht optimal ist – beispielsweise, wenn sich der Mehrwertsteuersatz mal wieder ändern sollte.
- Implementieren Sie den Aufzählungstyp `Mehrwertsteuersatz`, der zwei Werte für den normalen und den vergünstigten Mehrwertsteuersatz hat.
- Ziehen Sie die Gemeinsamkeiten der Produkte in eine gemeinsame Oberklasse `Produkt`. Führen Sie dafür eine Methode ein, die für ein Produkt angibt, ob die vergünstigte Mehrwertsteuer anwendbar ist. Verwenden Sie diese Methode bei der Preisberechnung.
- Führen Sie die Klasse `Einkaufskorb` ein, die eine Menge von Produkten verwaltet und deren Gesamtpreis bestimmen kann. Verwenden Sie diese Klasse in der `main`-Methode des `Warenhauses`.

Aufgabe 4: Java Garbage Collection

Der folgende Java-Programmcode generiert mehrere Objekte. Zwei der erzeugten Objekte werden am Ende des Programms nicht mehr referenziert und können daher von der *Garbage Collection* bereinigt werden. Welche sind dies? Geben Sie auch jeweils die Zeile an, ab der das Objekt nicht mehr referenziert wird.

¹Hier: <http://db.in.tum.de/teaching/ss17/ei2>

```
1 Assistent wittgenstein = new Assistent(3004, "Wittgenstein",
2                                     "Sprachtheorie", null);
3 wittgenstein.boss = new Professor(2137, "Kant", Professor.Rang.C4);
4 Vorlesung ethik = new Vorlesung(5043, "Ethik", 3, wittgenstein.boss);
5 ethik.dozent = new Professor(2126, "Russel", Professor.Rang.C4);
6 wittgenstein.boss = new Professor(2133, "Popper", Professor.Rang.C3);
7 Student jonas = new Student(25403, "Jonas", 12);
8 Pruefung pruefung = new Pruefung(jonas, ethik,
9                                 wittgenstein.boss, termin);
10 pruefung.student = new Student(28106, "Carnap", 3);
11 pruefung.student = jonas;
12 wittgenstein.boss = new Professor(2136, "Curie", Professor.Rang.C4);
```