

Praktikum zu
**Einführung in die Informatik für
LogWings und WiMas**
Wintersemester 2014/15

Übungsblatt 11
Bearbeitungszeit:
19.01.15 -
23.01.15

Aufgabe 11.1 – Klassenvariablen und -methoden vs. Objektvariablen und -methoden

Bearbeitungszeit: 15 Minuten

Das folgende Programm enthält sowohl Klassenvariablen und -methoden als auch Objektvariablen und -methoden. Manche Zuweisungen und Methodenaufrufe im Programm sind 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 11.2 – Objektorientierung: Vertrauen ist gut, Kontrolle ist Besser!

Bearbeitungszeit: 15 Minuten

Im letztem Übungsblatt haben Sie eine Klasse **Box** geschrieben, welche ein geometrisches Objekt modelliert. Der Konstruktor und die Setter dieser Klasse nehmen alle gemäß des Datentyps möglichen Werte entgegen und setzen diese entsprechend, ohne auf den semantischen Zusammenhang der Werte zu achten. So ist es z.B. möglich, den Maßen der Kiste einen negativen Wert zuzuweisen.

Ändern Sie den Konstruktor, sowie die Setter der Klasse **Box** so um, dass sie überprüfen, ob die übergebenen Parameter größer als 0 sind. Im Konstruktor soll im entsprechenden Attribut dann der Wert nicht zugewiesen und in den Settern keine Änderung vorgenommen werden.

Aufgabe 11.3 – Vererbung

Bearbeitungszeit: 15 Minuten

Wie können Sie von einer bereits bestehenden Klasse erben? Gehen Sie z.B. von einer Klasse **BachelorStudent** aus, welche von der allgemeineren Klasse **Student** erben möchte?

Von wie vielen Klassen können Sie in Java eine Klasse erben lassen?

Wenn Sie von keiner Klasse erben, existiert dennoch eine Oberklasse?

Durch welche Sichtbarkeitsmodifikatoren werden Attribute und Methoden der Oberklasse **Student** in der Unterklasse **BachelorStudent** sichtbar?

Welche in der erbenden Klasse sichtbaren Attribute und Methoden der Oberklasse müssen Sie deklarieren, um sie benutzen zu können?

Welche Programmteile werden **nicht** an die Unterklasse vererbt?

Können Sie einer Variablen vom Typ **Student** ein Objekt vom Typ **BachelorStudent** zuweisen?

Wie können Sie explizit auf sichtbare Attribute und Methoden der Oberklasse verweisen?

Aufgabe 11.4 – Eigene Objekte - Vererbung

Erstellen Sie eine neue Klasse **OpenBox** im Paket **blatt11**, welche von der Klasse **Box** aus Übungsblatt 10 erbt und die eine geöffnete Kiste beschreiben soll. Durch die Öffnung wurde die obere Seite (auf der Ebene: Breite \times Länge) der Kiste entfernt.

Wie können Sie auf die privaten Attribute der Kiste innerhalb der Klasse **OpenBox** zugreifen?

-
- Visualisieren Sie sich die Kiste mit einer Skizze! Bitte reichen Sie diese Skizze bei Ihrem Praktikumsleiter ein.
 - Schreiben Sie einen Konstruktor für die Klasse **OpenBox**, welcher die Maße entgegennimmt und an den Konstruktor der Oberklasse weiter reicht.
 - Überschreiben Sie die Methode **getSurfaceArea()** der Oberklasse, indem Sie die nun sichtbare Oberfläche der gesamten geöffneten Kiste berechnen und zurückgeben.

Schreiben Sie anschließend eine neue Klasse **OpenBoxTest** mit einer **main**-Methode und testen Sie dort die Berechnung der Kiste. Gehen sie dabei wie folgt vor:

- Deklarieren Sie eine Variable des Typs **Box** und eine Variable des Typs **OpenBox**.
 - Weisen Sie den Variablen jeweils eine neue Kiste entsprechenden Typs mit gleichen Maßen zu.
 - Geben Sie die Oberfläche der beiden Kisten aus.
 - Ändern Sie den Typ Ihrer **OpenBox**-Variablen zu **Box**.
 - Ändert sich das Verhalten Ihres Programmes?
-

Aufgabe 11.5 – Vererbung

Gegeben sei folgende Klasse:

```
1 package blatt11;
2
3 public class Person {
4
5     private final String firstName;
6     private final String lastName;
7
8     public Person(String firstName, String lastName){
9         this.firstName = firstName;
10        this.lastName = lastName;
11    }
12
13    public String getFirstName(){
14        return this.firstName;
15    }
16
17    public String getLastName(){
18        return this.lastName;
19    }
20 }
```

Erstellen sie eine neue Klasse **Contact**:

- Diese Klasse soll von der Klasse **Person** erben und zusätzlich einen String besitzen, welcher die Adresse der Person repräsentieren soll.
- Schreiben Sie einen Konstruktor für die Klasse **Contact**, welcher zusätzlich zu Vor- und Nachname die Adresse entgegennimmt und setzt.

- Implementieren Sie einen Getter und einen Setter für die Adresse.

Erstellen Sie anschließend zwei weitere Klassen **Employee** und **Customer**, die Mitarbeiter und Kunden repräsentieren sollen:

- Diese sollen jeweils von der Klasse **Contact** erben.
- Die Mitarbeiter sollen zusätzlich zu den bisherigen Informationen auch einen Stundenlohn erhalten, welcher **nicht** unter einem bestimmten Mindestgehalt (9€ / Stunde) sein darf.
- Für Kunden soll zusätzlich zu den bisherigen Informationen auch vermerkt werden, wie viel Geld sie bisher ausgegeben haben.
- Geben Sie der Klasse **Employee** eine Klassenkonstante, die den Mindestlohn repräsentiert. Klassenkonstanten deklarieren Sie über **public static final Typ NAME = wert;**, mit dem Namen für gewöhnlich in vollständig großen Lettern.
- Implementieren Sie den Konstruktor der Klasse **Employee** so, dass ein Angestellter grundsätzlich den Mindestlohn erhält. Sollte ein Angestellter mehr verdienen, so muss dies über den entsprechenden Setter verändert werden.
- Dieser Setter des Stundenlohns sollte überprüfen, ob das Gehalt oberhalb des Mindestlohns liegt und im Falle, dass man den Stundenlohn zu gering einstellt nichts machen.
- Für den Kunden soll eine Methode existieren, die den Kauf einer Ware repräsentiert und den bezahlten Preis entgegennimmt. Dieser soll auf das bisher ausgegebene Geld addiert werden.
- Ein neuer Kunde hat offensichtlich auch noch nichts ausgegeben.

Ergänzende Aufgaben

Aufgabe 11.6 – Testen lassen

Kopieren Sie die Klasse **Firm**, welche Ihnen auf der Veranstaltungswebseite bereitgestellt wird, in das Paket **blatt11**.

Führen Sie das Programm dieser Klasse aus, um zu testen, ob Ihre Implementierung korrekt ist.