

Praktikum zu

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

Wintersemester 2013/14

Übungsblatt 10

Bearbeitungszeit:

20.–24.01.2013

Auf diesem Übungsblatt finden Sie ein Punktesystem.

Die Punkteverteilung entspricht in etwa einer Klausur mit insgesamt 100 Punkten.

Es soll Ihnen helfen, die Schwierigkeit und den Zeitaufwand der jeweiligen Aufgabe einzuschätzen.

Aufgabe 10.1 – Klassenvariablen und -methoden vs. Objektvariablen und -methoden (4 Punkte)

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 10.2 – Überladen von Methoden

(2 Punkte)

Einige Banken und Vereine bewerben Ihre Kreditkarten, indem Sie gesperrte Karten versenden, die entsperrt werden, wenn der Kunde erklärt, die Karte benutzen zu wollen. Sie sollen Ihr Programm entsprechend erweitern. Hierzu benötigen Sie das *Überladen von Methoden*. Informieren Sie sich auf den Folien 43–44 in Kapitel 6 über das Überladen von Methoden.

Öffnen Sie die Klasse `Kreditkarte` aus Aufgabe 9.2. Fügen Sie einen weiteren Konstruktor ein, dem als zusätzliches Argument ein boolescher Wert übergeben wird, der `true` ist, wenn die neu erzeugte Karte von Beginn an gesperrt sein soll. Testen Sie den neuen Konstruktor.

Aufgabe 10.3 – Bücher, Bücher, Bücher ...

(15 Punkte)

Entwerfen Sie eine Klasse `Buch`, mit der Bücher einer Bibliothek dargestellt werden können. Ein Buch hat dabei – für die Aufgabenstellung vereinfacht – folgende Attribute:

- einen Titel, bestehend aus einer Zeichenkette,
- eine Inventarnummer,
- eine Markierung, ob das Buch momentan ausgeliehen ist, und
- die Ausweisnummer der ausleihenden Person.

Die Klasse `Buch` soll folgende Operationen unterstützen:

- Erzeugen eines neuen Buches (Name und Inventarnummer sollen bei der Erzeugung angegeben werden),
- Ausgabe des Titels des Buches,
- Ausgabe der Inventarnummer,
- Ausleihen des Buches mit einer Ausweisnummer,
- Abfrage, ob das Buch ausgeliehen ist,
- Abfrage, wer das Buch ausgeliehen hat,
- Zurückgeben eines Buches an die Bibliothek.

Programmieren Sie nun die Klasse `Buch` und eine Klasse `BuchTest`. Gehen Sie anhand der folgenden Schritte vor und führen Sie dabei regelmäßig Tests mittels der `main`-Methode der Klasse `BuchTest` durch:

- Implementieren Sie die leere Klasse `Buch`.
- Geben Sie der Klasse `Buch` entsprechend der obigen Aufzählung passende Attribute. Alle Attribute sollen `private` deklariert sein (siehe Folie 21 in Kapitel 6). (4 Punkte)
- Implementieren Sie für die Klasse `Buch` entsprechend der obigen Aufzählung einen passenden Konstruktor. Der Konstruktor soll `public` deklariert sein (siehe Folie 21 in Kapitel 6). (1 Punkt)
- Deklariieren Sie eine neue Klasse `BuchTest`. Erstellen Sie in ihrer `main`-Methode zwei Bücher. (1 Punkt)
- Fügen Sie der Klasse `Buch` die passenden Methoden hinzu und implementieren Sie diese. Alle Methoden sollen `public` deklariert sein. (7 Punkte)
- Mit Hilfe dieser Methoden sollen die beiden Bücher in der Klasse `BuchTest` mehrfach ausgeliehen und zurückgegeben werden. (2 Punkte)

Für die folgenden Aufgaben brauchen Sie die Klasse `Buch`, die Sie in Aufgabe 10.3 programmiert haben. Wir gehen davon aus, dass die Methode, die den Titel des Buches ausgibt, den Namen `gibAus` hat. Ebenso gehen wir davon aus, dass die Klasse `Buch` die Attribute `inventarnummer`, `ausgeliehen` und `ausweisnummer` hat. Eventuell müssen sie ihre Klasse `Buch` entsprechend anpassen.

Die Klasse `Buch` ist so ausgestattet, dass sie in einem Programm, welches eine Bibliothek verwaltet, verwendet werden kann. In einer Bibliothek können aber auch speziellere Arten von Büchern vorkommen, zum Beispiel Wörterbücher und Schul- oder Lehrbücher. Diese besitzen im Vergleich zu normalen Büchern noch weitere Attribute. Bei Wörterbüchern kommt z. B. die Sprache hinzu.

In Java lässt sich dies durch *Vererbung* ausdrücken. Sie werden im Folgenden zwei Klassen `Schulbuch` und `Woerterbuch` programmieren, die beide von der Klasse `Buch` *erben*. Man sagt auch: Sie sind von der Klasse `Buch` *abgeleitet*. Oder: `Buch` ist eine Oberklasse von `Woerterbuch` und `Schulbuch`.

Aufgabe 10.4 – Wörterbuch

Programmieren sie eine Klasse namens `Woerterbuch`, die von `Buch` erbt. In Java sieht das so aus:

```
class Woerterbuch extends Buch {  
    ...  
}
```

Die Klasse `Woerterbuch` enthält jetzt automatisch alle Methoden und Attribute der Klasse `Buch`, abgesehen vom Konstruktor, den sie noch der Klasse hinzufügen müssen:

```
public Woerterbuch(String titel, int inventarnummer) {  
    super(titel, inventarnummer);  
}
```

Die `super`-Zeile sorgt dafür, dass der Konstruktor der Oberklasse aufgerufen wird. Fügen Sie Ihrer `BuchTest`-Klasse einen Testaufruf hinzu:

```
Woerterbuch wb = new Woerterbuch("Titel ...", 123);  
wb.gibAus();
```

`gibAus` ist eine Methode aus Ihrer selbstprogrammierten `Buch`-Klasse, eventuell hat diese Methode daher bei Ihnen einen anderen Namen. Jetzt und im Folgenden gilt: **Testen Sie Ihr Programm nach jeder Änderung!**

- Fügen Sie der Klasse `Woerterbuch` ein Attribut hinzu, das die Sprache des Buches speichert.
- Passen Sie den Konstruktor an, sodass ihm als zusätzlichen Parameter die Sprache übergeben wird und das entsprechende Attribut nach dem `super`-Aufruf auf den Wert dieses Parameters gesetzt wird.
- Fügen Sie der Klasse eine Methode `gibSpracheAus` hinzu, die nur die Sprache des Wörterbuchs ausgibt.
- *Überschreiben* Sie in der Klasse `Woerterbuch` die Methode `gibAus`, die den Titel des Buches ausgibt, sodass sowohl der Titel als auch die Sprache ausgegeben werden. Dies bedeutet: Schreiben Sie in der Klasse `Woerterbuch` eine Methode, die auch `gibAus` heißt, die aber etwas anderes tut als die entsprechende Methode in der `Buch`-Klasse. Ein Tipp: Nutzen Sie `super.gibAus()`, um die `gibAus`-Methode der Oberklasse aufzurufen.

Aufgabe 10.5 – Schulbuch

(4 Punkte)

Gehen Sie nach dem Muster der obigen Aufgabe vor, aber implementieren Sie nun eine Klasse `Schulbuch`, die die Klasse `Buch` um das Attribut `klasse` erweitert. Dieses Attribut speichert die Klassenstufe, für die das Schulbuch gedacht ist. Geben Sie auch hier dem Konstruktor einen dritten Parameter. Als einzige neue Methode soll `Schulbuch` die Methode `setzeKlasse` bekommen, die das Attribut `klasse` setzt. Wenn der übergebene Wert für `klasse` nicht zwischen 1 und 12 liegt, soll der Wert unverändert bleiben. Überschreiben Sie ebenfalls die Methode `gibAus`, sodass sowohl Titel als auch die Klassenstufe ausgegeben werden.

Stellen Sie sicher, dass folgender Aufruf funktioniert:

```
Schulbuch sb = new Schulbuch("Biologie fuer Kids", 88, 2);
```

Aufgabe 10.6 – Was geht?

Betrachten Sie folgenden Java-Code:

```
1 Buch b = new Woerterbuch("Oxford Dictionary", 890, "Englisch");
2 b.gibAus();
3 b.gibSpracheAus();
```

Warum ist es möglich, einer Variablen vom Typ `Buch` ein Objekt vom Typ `Woerterbuch` zuzuweisen (Zeile 1)?

Warum funktioniert Zeile 2?

Warum funktioniert Zeile 3 nicht?
