

Praktikum zu
**Einführung in die Informatik für
LogWings und WiMas**
Wintersemester 2011/12

Übungsblatt 12
Bearbeitungszeit:
30.1.–3.2.2012

Für die folgenden Aufgaben brauchen Sie die Klasse `Buch`, die Sie in Aufgabe 11.3 programmiert haben. Eine mögliche Lösung finden Sie unter <http://pastie.org/3243689>. Wir gehen davon aus, dass die Methode, die den Titel des Buches ausgibt, den Namen `gibAus` hat.

Die Klasse `Buch` ist mit den Attributen `inventarnummer`, `ausgeliehen` und `ausweisnummer` so vorbereitet, 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 12.1 – 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 12.2 – Schulbuch

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 12.3 – 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?

Aufgabe 12.4 – Listen

Für die folgenden Aufgaben benötigen Sie die Klasse `Element`. Benutzen Sie nicht die Version aus der Vorlesung, sondern folgende vereinfachte Variante.

```
public class Element {
    public int wert;
    public Element weiter;
    public Element(int wert) {
        this.wert = wert;
        this.weiter = null;
    }
}
```

- Programmieren Sie eine Klasse `Liste`, die ein privates Attribut namens `kopf` vom Typ `Element` besitzt. Der Konstruktor soll keine Parameter haben und dieses Attribut auf `null` setzen. Fügen Sie eine `main`-Methode hinzu, in der sie die Klasse nach jeder der folgenden Änderungen testen.
- Fügen Sie der Klasse eine Methode `einfuegenVorn` hinzu, die einen übergebenen ganzzahligen Wert am Anfang der Liste einfügt.
- Fügen Sie der Klasse eine Methode `gibAlleAus` hinzu, die alle Elemente der Liste in korrekter Reihenfolge ausgibt. Die einzelnen Werte sollen durch Leerzeichen getrennt werden.
- Fügen Sie eine Methode `istLeer` hinzu, die zurückgibt, ob die Liste leer ist (`true` wenn leer, `false` wenn nicht).
- Fügen Sie eine Methode `anzahl` hinzu, die zurückgibt, wie viele Elemente die Liste besitzt.
- Fügen Sie eine Methode `letztesElement` hinzu, die das letzte Element der Liste zurückgibt. Es soll *nicht* der Wert, sondern eine Referenz auf das Element zurückgegeben werden. Ist die Liste leer, so wird `null` zurückgegeben.

Eine alte Klausuraufgabe

Die Aufgabenstellung wurde an Java angepasst.

Aufgabe 12.5

(15 Punkte)

a) Implementieren Sie die Klasse `Mensch`.

(10 Punkte)

i) Die Klasse `Mensch` hat folgende private Attribute:

- Geburtsjahr
- Größe in Metern
- Geschlecht

ii) Die Klasse `Mensch` hat einen öffentlichen Konstruktor, der alle Werte für Attribute als Parameter übergeben bekommt.

iii) Die Klasse `Mensch` hat folgende öffentliche Methoden:

- `setzeGroesse` mit `void` als Rückgabetyt und einem Fließkommamparameter namens „gross“
- `gibAlter` mit `int` als Rückgabetyt und einem ganzzahligen Parameter namens „aktuelles-Jahr“.

b) Schreiben Sie folgendes Programmfragment:

(4 Punkte)

i) Ein Mädchen wird im Jahr 2000 geboren. Es ist 0.52 m groß.

ii) Die Größe ändert sich auf 1.20 m.

iii) Das Alter des Mädchens im Jahre 2006 wird ausgegeben.

c) Nehmen wir an, dass die Klasse `Mensch` von der Klasse `Lebewesen` erben soll. Wie muss die Klasse `Mensch` geändert werden, damit sie von der Klasse `Lebewesen` erbt?

(1 Punkt)