

## Einführung in die Programmierung für Kunst und Multimedia

### Übungsblatt 6

Ende der Abgabefrist: 04.12.2013 12:00 Uhr

#### Hinweise zur Abgabe:

Übungsblätter dürfen NICHT in Teams abgegeben werden, da Sie sich durch eine erfolgreiche Bearbeitung einen Bonus für die Klausur verdienen können. Es ist zwar sinnvoll in kleinen Teams die Aufgaben zu diskutieren, die Lösungen müssen aber von jedem Studenten EINZELN bearbeitet werden. Bitte beachten Sie, dass abgeschriebene Lösungen mit 0 Punkten bewertet werden!

Sammeln Sie die Lösungen zu diesem Übungsblatt in einem zip-Archiv loesung05.zip. Dieses zip-Archiv können Sie schließlich in UniWorX abgeben.

**Wichtig:** Achten Sie bitte darauf, dass Ihre Lösungsdateien den korrekten Namen und das korrekte Format haben! Beides wird in der Angabe explizit angegeben. Dateien im falschen Format oder mit falschem Namen werden im Allgemeinen nicht korrigiert.

#### Aufgabe 6-1     Objekte und Klassen

**9 Punkte**

In der Übung haben Sie die grundlegende Idee von objektorientierter Programmierung kennengelernt. Programmieren Sie nun eine Klasse `Rechteck.java`, mithilfe derer Rechtecke als Objekte erzeugt werden können.

Rechtecke sollen in diesem Programm mit den folgenden vier Daten gespeichert werden:

- X-Koordinate
- Y-Koordinate
- Erste Seitenlänge
- Zweite Seitenlänge

Dabei soll es zwei Möglichkeiten (Konstruktoren!) geben, um ein Rechteck-Objekt zu erzeugen:

- Erzeugung eines normalen Rechtecks mit allen vier Werten (x, y, s1, s2)  
`Rechteck rechteck = new Rechteck(1, 2, 5.7, 7.4);`
- Erzeugung eines Quadrats mit Angabe von nur einer Seitenlänge (x, y, s)  
`Rechteck quadrat = new Rechteck(3, -5, 3.3);`

Um mit einem Rechteck arbeiten zu können, soll es folgende Methoden geben:

- `resize(double faktor)`  
verändert die Seitenlängen des Rechtecks um den übergebenen Faktor
- `moveTo(int x, int y)`  
verschiebt das Rechteck auf die neuen Koordinaten
- `changeSize(double seite)`  
verändert beide Seiten des Rechtecks auf eine übergebene Länge
- `changeSize(double seite1, double seite2)`  
verändert beide Seiten des Rechtecks auf die beiden übergebenen Längen
- `String toString()`  
gibt die Daten des Rechtecks in Klammern kommagetrennt aus

Testen Sie Ihre Klasse mit der Klasse `RechteckTest.java`, die Sie auf der Vorlesungshomepage herunterladen können. Die Testklasse enthält auch die `main`-Methode, um das Programm auszuführen. Kopieren Sie die Testklasse in dasselbe Package und führen Sie das Programm normal aus.

Sie sollten die folgende Ausgabe auf der Konsole bekommen:

```
Zwei Rechtecke wurden erstellt  
Rechteck 1: (1,2,5.7,7.4)  
Rechteck 2: (3,-5,3.3,3.3)
```

```
Rechtecke wurden verändert  
Rechteck 1: (2,3,64.8,40.5)  
Rechteck 2: (4,5,5.5,5.5)
```

Bitte geben Sie Ihre Lösung als kompilierbare Java-Datei `Rechteck.java` ab. Bitte achten Sie unbedingt darauf, dass Ihre Lösung fehlerfrei und kompilierbar ist. **Es werden ausschließlich lauffähige Lösungen im korrekten Format bewertet.**

## **Aufgabe 6-2**    Objekte und Instanzen

**11 Punkte**

In einer objektorientierten Umgebung soll eine stark vereinfachte Simulation des Bankwesens programmiert werden. Erstellen Sie dazu in JAVA eine Klasse `Banksimulation`, mit der das Programm aufgerufen wird. Erstellen Sie außerdem eine Klasse `Bank`, eine Klasse `Kunde` und eine Klasse `Konto`.

Das Programm wird in der Klasse `Banksimulation` (`main`-Methode!) ausgeführt. In dieser Klasse benötigen Sie keine weiteren Methoden, Sie können also alle Aufrufe in die `main`-Methode schreiben. In der `main`-Methode sollten die Klassen `Kunde` und `Konto` nicht verwendet werden.

Überlegen Sie sich sinnvolle Variablen und Methoden in den einzelnen Klassen, die in einer Banksimulation benötigt werden, implementieren Sie Ihre Lösung jedoch so, dass mindestens die folgenden Eigenschaften erfüllt sind:

- In der Klasse Banksimulation werden alle Banken erzeugt und gespeichert
- Jede Bank hat einen Namen und eine Bankleitzahl
- In einer Bank können neue Kunden und neue Konten angelegt werden
- In einer Bank können Konten an Kunden zugewiesen werden
- Jeder Kunde hat einen Namen
- Ein Kunde kann mehrere Konten besitzen
- Jedes Konto hat eine Kontonummer
- Jedes Konto hat einen Besitzer und eine Bank
- In einem Konto können Einzahlungen und Auszahlungen vorgenommen werden sowie der aktuelle Kontostand abgefragt werden
- Konten dürfen nicht überzogen werden

Achten Sie auf eine sinnvolle Benennung Ihrer Variablen und ausreichende Kommentierung Ihres Codes, insbesondere Javadoc. Für schlecht benannte Variablen und fehlende Kommentare gibt es Punktabzug.

Bitte geben Sie Ihre Lösungsklassen als *Banksimulation.java*, *Bank.java*, *Konto.java* und *Kunde.java* ab. Bitte achten Sie unbedingt darauf, dass Ihre Lösung fehlerfrei und kompilierbar ist. **Es werden ausschließlich lauffähige Lösungen im korrekten Format bewertet.**

### **Abgabe**

Zulässiges Dateiformat für die Lösungen dieses Übungsblattes ist JAVA. Bitte geben Sie Ihre Lösung als ZIP-Datei bis zum 04.12.13 12:00 Uhr in UniWorX (<https://uniworx.ifi.lmu.de>) ab.

**Hinweis:** Verspätete Abgaben, Abgaben im falschen Dateiformat und nicht lauffähige Java-Dateien werden nicht bewertet.