

Einführung in die Programmierung für Kunst und Multimedia

Übungsblatt 8

Ende der Abgabefrist: 18.12.2013 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 loesung08.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 8-1 Erweiterung der Banksimulation

13 Punkte

Im Rahmen des 6. Übungsblattes sollten Sie den ersten Entwurf einer einfachen Banksimulation in Java erstellen. Eine Musterlösung dazu finden Sie zum Download auf der Homepage. Sie können sowohl auf diese als auch auf Ihre eigene Lösung aufbauen – beachten Sie jedoch, dass Ihre Lösung die Anforderungen von Übungsblatt 6 vollständig erfüllt.

1. In der Vorlesung haben Sie das Prinzip der Vererbung kennengelernt. Um nun zwischen Girokonten und Sparkonten unterscheiden zu können, erstellen Sie eine Klasse `Girokonto` und eine Klasse `Sparkonto`. Beide sollen von der Klasse `Konto` erben.

Die Eigenschaften der Elternklasse `Konto` werden dabei vollständig übernommen.

Girokonten bekommen als zusätzliche Eigenschaft einen Dispositionskredit; dieser wird bei Erzeugung des Girokontos definiert. Nur Sparkonten können nicht überzogen werden, Girokonten können bis zur Höhe des Dispositionskredits überzogen werden.

Die Klasse `Sparkonto` erhält als zusätzliche Eigenschaft einen Zinssatz, der ebenfalls bei der Erzeugung übergeben wird. Eine neue Methode soll zurückgeben, wie hoch die aktuellen Zinsen wären ($\text{Kontostand} \cdot \text{Zinssatz}$).

Passen Sie ebenfalls die Datenstrukturen in den Klassen `Bank` und `Kunde` an, in denen die Konten gespeichert werden sowie die Methoden, mit denen neue Konten erstellt werden, um künftig zwischen Sparkonten und Girokonten unterscheiden zu können

2. In der Übung haben Sie Listen als neue Form kennengelernt, um mehrere Werte in einem Objekt zu speichern. Eine Liste bietet dabei gegenüber Arrays den Vorteil, dass sie beliebig erweiterbar ist und somit nicht bekannt sein muss, wie viele Elemente gespeichert werden.

Die Musterlösung von Blatt 6 verwendet ebenfalls Listen, um Banken, Konten und Kunden in den jeweiligen Klassen zu speichern. Erweitern Sie die Klassen `Bank` und `Kunde` um je eine Funktion, die die aktuellen Inhalte der Listen der Klasse syntaktisch schön auf die Konsole ausgibt, also z.B. eine Funktion `printAlleBanken()` in der Klasse `Bank`, die alle Kunden einer Bank auf die Konsole schreibt, eine Funktion `printSparkonten()` in der Klasse `Kunde`, die alle Sparkonten eines Kunden auf die Konsole schreibt usw.

3. In der Vorlesung haben Sie Javadoc-Kommentare kennengelernt, in der Übung haben Sie gesehen, wie man eine Javadoc-Dokumentation erzeugt. Statten Sie Ihre Klassen und Methoden mit entsprechenden Kommentaren aus und erzeugen Sie eine Javadoc Ihres Programms.

Diese wird nach Konvention im Ordner `doc` auf derselben Ebene wie `src` in Ihrem Projekt gespeichert.

Bitte geben Sie Ihr vollständiges Eclipse-Projekt als zip-Archiv ab. Verwenden Sie dazu einfach den gesamten Ordner Ihres Projektes im workspace. Bitte achten Sie unbedingt darauf, dass Ihre Lösung fehlerfrei und kompilierbar ist. **Es werden ausschließlich lauffähige Lösungen im korrekten Format bewertet.**

Aufgabe 8-2 Vererbung

7 Punkte

In der Vorlesung haben Sie das Prinzip der Vererbung kennengelernt. Erstellen Sie dazu eine Klasse `Mensch` mit entsprechenden Klassenvariablen für Name, Vorname, Adresse, drei Integer-Variablen für das Geburtsdatum und eventuell weiteren Eigenschaften. Denken Sie dabei auch an entsprechende Getter-/Setter-Methoden und erstellen Sie eine Methode, die das Geburtsdatum aus den drei Integer-Variablen im deutschen Format als String zurückgibt.

Erstellen Sie außerdem die Klassen `Student`, `Professor` und `Mitarbeiter`. Alle drei Klassen erben von der Klasse `Mensch` und bekommen dazu noch weitere Eigenschaften. Überlegen Sie sich zu jeder Rolle mindestens zwei weitere Eigenschaften wie Matrikelnummer, Gehalt, Vorgesetzter o.ä. und legen Sie entsprechende Konstruktoren und Getter-/Setter-Methoden an.

Implementieren Sie Ihre main-Methode in einer Klasse `Vererbung`, in der Sie testweise Objekte aller Klassen erzeugen und jede derer Funktionen einmal aufrufen.

Bitte geben Sie Ihre Lösung als Java-Dateien `Vererbung.java`, `Mensch.java`, `Student.java`, `Professor.java` und `Mitarbeiter.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 18.12.13 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.