

Übung 2: Objektorientierte Programmierung

Ausgabe: 24.10.2012

Abgabe: 01.11.2012

Aufgabe 3: Klassen-Deklarationen als Datenbank-Entwurf (6 Punkte)

Angenommen, Sie wollen Anmeldungen von Studenten für Vorlesungen (Vorlesungs-Belegungen) mit einem Java-Programm verwalten. Deklarieren Sie dazu drei Klassen:

- Eine Klasse für Studenten (selbstverständlich auch Studentinnen) mit den Attributen Name, Vorname und Matrikelnummer.
- Eine Klasse für Vorlesungen mit den Attributen Titel und Leistungspunkte.
- Eine Klasse für Belegungen von Studenten für Vorlesungen. Jedes Objekt der Klasse soll einer Anmeldung eines Studenten für eine Vorlesung entsprechen. Diese Klasse muss also ein Attribut für den Studenten enthalten (Klassen sind ja auch als Datentypen für Variablen/Attribute verwendbar), und eins für die Vorlesung. Außerdem soll in diesem Objekt die Note gespeichert werden: Zunächst der Wert 0, solange der Student noch keine Prüfung gemacht hat, später eine der Noten 1.0, 1.3, 1.7, 2.0, 2.3, 2.7, 3.0, 3.3, 3.7, 4.0, 5.0. Natürlich können Sie mit den bisher in der Vorlesung vorgestellten Mitteln noch nicht sicherstellen, dass nur diese Werte für die Note gespeichert werden, aber zumindest sollten diese Werte in dem Attribut gespeichert werden können.

Deklarieren Sie die drei Klassen in drei Java-Dateien. Wählen Sie dabei sinnvolle Datentypen. Matrikelnummern sind 9-stellige Zahlen. Schreiben Sie dann eine vierte Klasse "StudentTest" mit einem Hauptprogramm "main", das

- zwei Studenten-Objekte anlegt, und zwar für "Frank Schmidt", Matrikelnummer 207000000, und für "Lisa Weiss", Matrikelnummer 123456789.
- ein Vorlesungs-Objekt anlegt, und zwar für die Vorlesung "Objektorientierte Programmierung" mit 5 Leistungspunkten.
- zwei Belegungs-Objekte erzeugt, und zwar für die Belegung von beiden Studierenden für die Vorlesung "Objektorientierte Programmierung". Als Note tragen Sie bitte für Lisa Weiss eine 1.0 ein, und für Frank Schmidt eine 2.3.
- Schließlich drucken Sie bitte alle drei Attribute von beiden Studenten-Objekten aus, und zwar in der Form

Frank Schmidt, Matrikel-Nr. 207000000
Lisa Weiss, Matrikel-Nr. 123456789

Geben Sie bitte die vier Java-Dateien mit den Klassendeklarationen gepackt in einem jar-Archiv ab.

Sie können ein jar-Archiv auf folgende Weise erzeugen: Angenommen Ihre vier Dateien tragen die Namen Student.java, Vorlesung.java, Belegung.java und Aufgabe3.java und befinden sich alle im gleichem Verzeichnis z.B. C:\NetBeansProjects\JavaApplication1\src, Wechseln Sie in dieses Verzeichnis und rufen Sie dort das folgende Kommando auf:

```
jar -cvf aufgabe3.jar Student.java Vorlesung.java Belegung.java Aufgabe3.java
oder
jar -cvf aufgabe3.jar *.java
```

Die einzelnen Optionen haben folgende Bedeutung:

Option c steht für create und zeigt an, dass ein neues jar-Archiv angelegt wird.

Option v steht für verbose. Es werden Ausgaben über den aktuellen Stand des Vorgangs produziert.

Option f steht für file und gibt den Namen des jar-Archives (hier: aufgabe3.jar) an.

Aufgabe 4: Zinsen (4 Punkte)

Angenommen, Sie wollen gerne wissen, nach wie viel Jahren aus einem Euro mit 3% Zinsen 100 Euro geworden sind. Schreiben Sie dafür eine Klasse `Zins` mit Prozedur `main`, die die Jahreszahl ausgibt, bei der Ihr Kontostand erstmals mindestens 100 Euro ist, wenn Sie 1 Euro in 2012 anlegen. Sie repräsentieren den Kontostand am besten in Cent in einer `int`-Variable (die Bank rundet die 3% jedes Jahr auf volle Cent ab). Die wesentliche Formel für die Neuberechnung des Kontostandes in jedem Jahr ist

$$\text{kontostand} = \text{kontostand} + (\text{kontostand} * 3 / 100);$$

Sie müssen also eine Schleife schreiben, die den Kontostand entsprechend erhöht und die Jahreszahl (beginnend von 2012) hochzählt, solange der Kontostand noch kleiner ist als 100 Euro, d.h. 10000 Cent. Wenn Sie wollen, können Sie zur Kontrolle die Jahreszahl und den Kontostand in der Schleife ausgeben, z.B. würde man so die Ausgabe "2013: 103" bekommen. Am Ende sollen Sie den Text "100 Euro erreicht in: " mit der Jahreszahl ausgeben.

Geben Sie bitte die Java-Datei mit dem Programm ab.