

Übung 3: Objektorientierte Programmierung

Ausgabe: 01.11.2012

Abgabe: 08.11.2012

Aufgabe 5: Klassen-Deklarationen als Datenbank-Entwurf (7 Punkte)

Sie haben in der letzten Übungsserie (Aufgabe 3) eine Klasse für Studenten, Vorlesungen und Belegungen von Studenten für Vorlesungen geschrieben. Inzwischen wurde in der Vorlesung erläutert, dass es günstig ist, Attribute als `private` zu deklarieren, und den Zugriff nur über Methoden der Klasse zu erlauben. Ändern Sie die drei Klassen entsprechend. Im Einzelnen haben Sie die folgenden Aufgaben:

- Deklarieren Sie eine Klasse für Studenten mit privaten Attributen für Name, Vorname und Matrikelnummer. Des Weiteren benötigen Sie:
 - einen Konstruktor, welcher als Parameter den Namen, den Vornamen und die Matrikelnummer des entsprechenden Studenten bekommt und die Attribute entsprechend setzt,
 - für jedes Attribut eine Methode, welche den Wert des jeweiligen Attributes mittels `return` zurückgibt,
 - eine Methode `toString`, die Vorname, Nachname und in Klammern die Matrikelnummer als Zeichenkette zurückliefert (also eine kompakte Repräsentation der Daten des Objektes). Folgen Sie dem Beispiel in der Vorlesung und deklarieren Sie die Methode als `@Override public String toString() {...}`

Die Attribute können somit nicht mehr nachträglich geändert werden. Sie brauchen im Konstruktor keine Korrektheitsprüfung für die Attributwerte vorzusehen.

- Deklarieren Sie eine Klasse für Vorlesungen mit privaten Attributen für Titel und Leistungspunkte. Da auch bei einer Vorlesung der Titel und die erreichbaren Leistungspunkte nicht nachträglich verändert werden können, müssen Sie nur:
 - einen Konstruktor erstellen, welcher Werte für die entsprechenden Attribute als Parameter erhält und
 - für jedes Attribut eine Methode schreiben, welche den Wert des jeweiligen Attributes zurückgibt,
 - wieder eine Methode `toString()` deklarieren, die z.B. für diese Vorlesung folgende Zeichenkette liefert: "Objektorientierte Programmierung (5 LP)"

Testen Sie im Konstruktor bitte, dass die Leistungspunkte nicht negativ sind. Im Fehlerfall geben Sie eine Meldung auf die Konsole aus und beenden anschließend das Programm mittels `System.exit(1);` (wie in der Vorlesung gezeigt). Andere Möglichkeiten zur Fehlerbehandlung werden in der Vorlesung erst später besprochen.

- Deklarieren Sie eine Klasse für Belegungen von Studenten für Vorlesungen. Wie in der letzten Übungsserie beschrieben, besitzt diese Klasse drei Attribute: ein Attribut, welches auf den Studenten verweist, eins, welches auf die Vorlesung verweist, und eins für die Note. Da diesmal alle Attribute `private` sind, müssten Sie

- einen Konstruktor erstellen, welcher einen Parameter für den Studenten und einen Parameter für die Vorlesung hat. Die Note wird nicht als Parameter übergeben, sondern von dem Konstruktor auf 0 gesetzt.
- eine Methode für jedes Attribut, welche den Wert des entsprechenden Attributes zurückgibt.
- eine Methode mit dem Namen `bewerte`, welche als Parameter die Note übergeben bekommt. Da eine einmal gegebene Note nicht einfach verändert werden darf, muss überprüft werden, ob das Attribut `Note` den Wert 0 hat. Nur in diesem Fall, darf die Note geändert werden, andernfalls soll auf der Konsole mittels `System.out.println` ausgegeben werden, dass die Note bereits festgesetzt wurde. Damit der Benutzer weiß, auf welchen Studenten und welche Veranstaltung sich die Fehlermeldung bezieht, sollen diese ebenfalls mit ausgegeben werden. Das Programm soll aber nicht abgebrochen werden, sondern die Note soll einfach unverändert bleiben. Des Weiteren müssen Sie überprüfen, ob die Note einen gültigen Wert hat (es reicht aus zu prüfen, ob die Note zwischen 1.0 und 5.0 liegt!). Falls sie keinen gültigen Wert hat, soll dies ebenfalls ausgegeben werden. Wieder soll das Programm nicht abgebrochen werden.

Beachten Sie bitte, dass der Test auf Gleichheit in Java als `==` geschrieben wird, weil `=` schon für die Zuweisung verbraucht ist. Wenn Sie prüfen wollen, ob das Attribut `note` den Wert 0 hat, geht dies also mit `note == 0`. Der umgekehrte Test, ob `note ≠ 0`, schreibt sich in Java `note != 0`. Falls Sie zwei Bedingungen gleichzeitig testen wollen, können Sie diese Bedingungen mittels `&&` logisch "und" verknüpfen. Entsprechend ist `||` das logische "oder".

- Schließlich benötigen Sie auch für diese Klasse eine Methode `toString()`. Die Ausgabe soll wie folgt aussehen: `Frank Schmidt (207000000), Objektorientierte Programmierung, Note: 2.3`

Nachdem Sie diese drei Klassen erstellt haben, schreiben Sie auch eine Klasse `StudentTest`, in welcher sie im Hauptprogramm (`main`):

- zwei Studenten-Objekte anlegen für: „Frank Schmidt“ mit der Matrikelnummer: 207000000 und „Lisa Weiss“ mit der Matrikelnummer 123456789
- ein Vorlesungs-Objekt für die Vorlesung „Objektorientierte Programmierung“ anlegen
- zwei Belegungs-Objekte erzeugen (jeder der genannten Studenten besucht auch die OOP-Vorlesung)

Anschließend tragen Sie für Lisa Weiss die Note 1.0 ein. Versuchen Sie daraufhin diese Note auf eine 2.0 zu ändern (in diesem Fall wird die Fehlermeldung als Ausgabe wie oben beschrieben erwartet). Versuchen Sie die Note für Frank Schmidt auf 5.3 zu setzen (auch hier sollte eine Fehlermeldung ausgegeben werden), dann auf `-1.0` (wieder muss es eine Fehlermeldung geben), und setzen Sie die Note für ihn schließlich auf 2.3.

Geben Sie zur Kontrolle alle fünf Objekte mit `System.out.println(...)` aus, dabei wird automatisch die von Ihnen definierte Methode `toString()` benutzt.

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

Aufgabe 6: Fakultätsberechnung mit Eingabe von der Konsole (3 Punkte)

Schreiben Sie ein Programm, welches die Fakultät einer gegebenen Zahl berechnet ($n! = 1 * 2 * 3 * \dots * (n - 1) * n$) und den berechneten Wert auf der Konsole ausgibt. Beachten Sie dabei die Spezialfälle, dass eine negative Zahl oder die Zahl Null übergeben wird. Verwenden Sie das folgende Programm und ergänzen Sie an der gekennzeichneten Stelle die Fakultätsberechnung und die Ausgabe.

Listing 1: Java-Code

```
1 import java.io.*;
2
3 public class Fakultaet
4 {
5     public static void main(String[] args) throws IOException
6     {
7         int zahl = 0;
8         BufferedReader buffer = new BufferedReader(
9             new InputStreamReader(System.in));
10
11         System.out.println("Bitte geben Sie eine Zahl ein:");
12         zahl = Integer.parseInt(buffer.readLine());
13
14
15         // hier erfolgt Ihre Berechnung der Fakultaet!
16         // ...
17
18         // Achten Sie darauf, dass der Fakultätswert auch ausgegeben wird!
19     }
20 }
```