Institut für Informatik
der Universität München
Prof. Dr. M. Hofmann
Dr. J. Johannsen, Dr. R. Matthes

SS 2004

6.5.04

Übungen zur Vorlesung
Informatik II

Blatt 3

Abgabe der Hausaufgaben bis spätestens am 17.5.04, 14:00 Uhr über
`http://lehre.tcs.ifi.lmu.de/info2/Abgabe/abgabe.php`,
Bearbeitung in Gruppen zu max. 3 Personen ist zulässig.

Bitte halten Sie sich an folgende Grundsätze zur Bearbeitung der Aufgaben. Die Bewertung Ihrer Abgaben kann andernfalls deutlich herabgesetzt werden.

Verwenden Sie genau die Signatur der Aufgabenstellung, insbesondere die Namen der Klassen und Methoden in der vorgegebenen Klein- und Großschreibung.

Verwenden Sie keine „magischen“ Größen in Ihren Programmen, d. h., weisen Sie Konstanten (außer den Zahlen 0 und 1 und dem leeren String) immer einer mit `final` deklarierten und in Großbuchstaben geschriebenen „Variablen“ zu und verwenden Sie anschließend nur diese. Wer gute Gründe hat für eine Abweichung von dieser Regel, der kommentiere seinen Code entsprechend.

Zu jeder Klasse xxx, für die keine `main`-Methode verlangt ist, muß eine Testklasse xxx`Test` geschrieben werden, die sinnvolle Tests der Klasse xxx enthält (in einer `main`-Methode). In den Tests dürfen „magische“ Größen natürlich beliebig vorkommen!

Jede Klasse und `public`-Methode muß angemessen mit `javadoc`-Kommentaren dokumentiert werden.

Programmieraufgabe P-7 (`Wechsler2.java`): **6 Punkte**

Erstellen Sie eine Klasse `Wechsler2`, die das Gleiche leistet wie die Klasse `Wechsler` aus Aufgabe **P-1**. Benutzen Sie nun jedoch statische Methoden, um das vielfache Vorkommen von gleichartigem Code zu vermeiden.

Dabei sollen die Euro- und Centausgaben durch dieselben Methoden behandelt werden, ebenso die Blöcke von Münzen bzw. Scheinen, deren Werte von der gleichen Größenordnung sind (also etwa 1, 2 und 5 genauso wie 10, 20 und 50 Euro).

Programmieraufgabe P-8 (`Darlehen.java, DarlehenTest.java`): **8 Punkte**

Schreiben Sie eine Klasse `Darlehen`, die die Entwicklung eines Annuitätendarlehens simuliert. Beim Abschluss eines Darlehens wird ein gewisser Betrag B ausgeliehen, sowie ein effektiver jährlicher Zinssatz J und eine anfängliche Tilgungsrate T vereinbart. Daraus berechnen sich der monatliche Zinssatz M und die monatliche Rate R anhand der Formeln:

$$\left(1+\frac{M}{100}\right)^{12} = 1+\frac{J}{100}$$

$$R = B * \frac{1}{100}\left(M + \frac{T}{12}\right)$$

Diese Rate R wird jeden Monat zurückgezahlt, davon werden $M\%$ der Restschuld als Zinsen einbehalten, und die Differenz von der Restschuld abgezogen (im Bankenjargon: *getilgt*).

Die Klasse soll Methoden zur Abfrage der monatlichen Rate und der verbleibenden Restschuld anbieten, sowie zwei Methoden, die das Weiterschreiten der Zeit um einen Monat bzw. ein Jahr simulieren.

Beachten Sie, dass die Darlehenslaufzeit in dem Monat endet, in dem der Restschuldbetrag erstmals nicht mehr positiv ist.

Testen Sie Ihre Klasse, indem Sie in der Testklasse ein Darlehen über EUR 150 000 mit einem effektiven Jahreszins von 4,5% und einer anfänglichen Tilgungsrate von 1,5% beobachten. Zur Kontrolle: dieses Darlehen ist im 32. Jahr zurückgezahlt.

Programmieraufgabe P-9 (`DarlehenApplet.java`, `Darlehen.html`): **6 Punkte**

Schreiben Sie ein Applet, in dem die Entwicklung der Restschuld eines Darlehens wie in Aufgabe **P-8** grafisch veranschaulicht wird.

Es soll ein Balkendiagramm gezeichnet werden (mittels Objekten aus `java.awt.Rectangle` oder `java.awt.geom.Rectangle2D.Double`), in dem der $n$te Balken die Höhe der Restschuld nach n Jahren repräsentiert, wobei n von 0 bis zur Laufzeit des Darlehens in Jahren läuft. Ein Beispiel (als Screenshot) finden Sie auf der Vorlesungshomepage.

Zur Vereinfachung der Skalierung der grafischen Ausgabe dürfen Sie den Darlehensbetrag fest vorgeben. Zinssatz und Tilgungsrate sollen vom Benutzer des Applets eingegeben werden können. Dazu wird die statische Methode `showInputDialog` der Klasse `javax.swing.JOptionPane` (nach Importieren!) benutzt z. B. wie folgt.

```
String input = JOptionPane.showInputDialog("Zinssatz in % :");
zinssatz = Double.parseDouble(input);
```

Da ein Applet (mit `appletviewer` oder einem Browser) ausgeführt werden kann, brauchen Sie hier natürlich keine Testklasse abzugeben, dafür aber eine HTML-Datei, in der das Applet aufgerufen wird.