Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. R. Hennicker, A. Klarl

WS 2016/17
Übungsblatt 5

Übungen zu Einführung in die Informatik: Programmierung und Software-Entwicklung: Lösungsvorschlag

Aufgabe 5-1 Wiederholungsanweisungen in Java (while-Schleife) *Präsenz*

Ein Autokonzern investiert für die Entwicklung eines neuen Fahrzeug-Modells einen bestimmten Investitionsbetrag. Der Konzern möchte kalkulieren, nach wievielen Jahren sich der Investitionsbetrag amortisiert hat, d.h. nach wievielen Jahren der Gewinn aus den Einnahmen durch den Autoverkauf diesen Investitionsbetrag deckt. Dabei wird davon ausgegangen, dass ein Auto 22500 Euro in der Herstellung kostet und für 25000 Euro verkauft wird. Der Konzern schätzt dazu, wieviele Autos im ersten Jahr verkauft werden. Er geht davon aus, dass jedes Jahr 5% mehr Autos als im Vorjahr verkauft werden.

Schreiben Sie in einer Klasse `Investitionsrechner` ein Java-Programm, welches pro Jahr ausgibt, das wievielte Jahr gerade berechnet wird, wieviele Autos in diesem Jahr schätzungsweise verkauft werden, wie hoch die Herstellungskosten für alle verkauften Autos in diesem Jahr sind, mit welchem Umsatz in diesem Jahr gerechnet werden kann, wie hoch der Reingewinn (also Umsatz - Herstellungskosten) in diesem Jahr ist und wie hoch der aktuelle Gesamtgewinn über alle Jahre gerechnet ist. Das Java-Programm soll diese Berechnung stoppen, sobald der Gesamtgewinn über alle Jahre den Investitionsbetrag erreicht oder überschreitet.

Bestimmen Sie zunächst, welche lokalen Variablen Sie benötigen (mit Typ). Testen Sie Ihr Programm für 1) einen Investitionsbetrag von einer Million Euro und einer geschätzten Anzahl von 50 im ersten Jahr verkauften Autos und 2) für einen Investitionsbetrag von 25 Millionen Euro und einer geschätzten Anzahl von 1000 im ersten Jahr verkauften Autos.

Lösungsidee:
Wir benötigen folgende Variablen:

- Eine Variable `investitionsbetrag` vom Typ `double` für den zu tätigenden Investitionsbetrag.
- Eine Variable `herstellungskosten` vom Typ `double` für die Herstellungskosten eines Autos, initialisiert mit 22500.
- Eine Variable `verkaufspreis` vom Typ `double` für den Verkaufspreis eines Autos, initialisiert mit 25000.
- Eine Variable `anzahlVerkaufterAutos` vom Typ `int` für die Schätzung der Anzahl der verkauften Autos pro Jahr.
- Eine Variable `verkaufssteigerung` vom Typ `double` für die zu erwartende prozentuale Verkaufssteigerung, initialisiert mit 5.
- Eine Variable `gesamtGewinn` vom Typ `double` um den Gesamtgewinn über alle Jahre zu speichern.
- Eine Variable `anzahlJahre` vom Typ `int` um die Anzahl der verstrichenen Jahre zu speichern.

Den Gesamtgewinn initialisieren wir mit 0.0, da bisher noch kein Gewinn gemacht wurde. Die Anzahl der verstrichenen Jahre initialisieren wir mit 1, da wir in diesem Jahr beginnen wollen.

Solange der Gesamtgewinn noch kleiner als der Investitionsbetrag ist (`gesamtGewinn < investitionsbetrag`), gehen wir wie folgt vor:

1. Berechne die Kosten pro Jahr als `herstellungskosten * anzahlVerkaufterAutos`.
2. Berechne den Umsatz pro Jahr als `verkaufspreis * anzahlVerkaufterAutos`.
3. Berechne den Gewinn pro Jahr als `umsatzProJahr - kostenProJahr`.
4. Berechne den aktuellen Gesamtgewinn über alle Jahre, indem der Gewinn dieses Jahres zum bisherigen Gesamtgewinn dazugezählt wird.
5. Gebe die berechneten Daten auf dem Bildschirm aus.
6. Zähle zuletzt die Anzahl der Jahre um eins hoch, da dieses Jahr nun verstrichen ist.
7. Berechne die Schätzung der Anzahl der verkauften Autos für das nächste Jahr unter Beachtung der Verkaufssteigerung von 5%.

Dieser Algorithmus wird folgendermaßen in Java umgesetzt.

```
1  public class Investitionsrechner {
2    public static void main(String[] args) {
3      double investitionsbetrag = 1000000;
4      double herstellungskosten = 22500;
5      double verkaufspreis = 25000;
6      int anzahlVerkaufterAutos = 50;
7      double verkaufssteigerung = 5;
8      double gesamtGewinn = 0.0;
9      int anzahlJahre = 1;
10
11     while (gesamtGewinn < investitionsbetrag) {
12       double kostenProJahr = herstellungskosten * anzahlVerkaufterAutos;
13       double umsatzProJahr = verkaufspreis * anzahlVerkaufterAutos;
14
15       double gewinnProJahr = umsatzProJahr - kostenProJahr;
16
17       gesamtGewinn = gesamtGewinn + gewinnProJahr;
18
19       System.out.println("Jahr " + anzahlJahre + ":");
20       System.out.println("   Anzahl verkaufter Autos: "
21         + anzahlVerkaufterAutos);
22       System.out.println("   Kosten: " + kostenProJahr);
23       System.out.println("   Umsatz: " + umsatzProJahr);
24       System.out.println("   Gewinn: " + gewinnProJahr);
25       System.out.println("   Gesamtgewinn: " + gesamtGewinn);
26
27       anzahlJahre++;
28       anzahlVerkaufterAutos = (int) (anzahlVerkaufterAutos +
29         verkaufssteigerung/100.0 * anzahlVerkaufterAutos);
30     }
31   }
32 }
```

Aufgabe 5-2 Wiederholungsanweisungen in Java (while-Schleife) *Hausaufgabe*

Sie möchten ein Immobilien-Darlehen über eine unbegrenzte Laufzeit aufnehmen und berechnen, nach wievielen Jahren Sie das Darlehen komplett zurückgezahlt haben. Mit der Bank haben Sie einen jährlichen Zinssatz sowie einen Tilgungssatz vereinbart. Pro Jahr wird eine gleichbleibende Jahresrate an die Bank gezahlt in Höhe von `Darlehensbetrag * (Zinssatz + Tilgungssatz)/100.0`.

Am Ende jedes Jahres sind Jahreszinsen entsprechend des vereinbarten Zinssatzes angewandt auf die Restschuld am Ende des vorherigen Jahres zu zahlen. Die Jahrestilgung am Ende jedes Jahres ergibt sich dann durch die Differenz aus Jahresrate und Jahreszinsen. Die Restschuld am Ende jedes Jahres verringert sich entsprechend der Jahrestilgung.

Schreiben Sie in einer Klasse `Tilgungsrechner` ein Java-Programm, das für einen gegebenen Darlehensbetrag, Zinssatz und Tilgungssatz eine Ausgabeliste in der unten angegebenen Form ausgibt, in der, bis das Darlehen vollständig zurückgezahlt ist, pro Jahr die Jahreszinsen, die Jahrestilgung und die Restschuld am Ende des Jahres ausgewiesen sind.

Ausgabeformat:

```
Jahr 1:
   Jahreszinsen: 200.0
   Jahrestilgung: 1000.0
   Restschuld: 9000.0
Jahr 2:
   Jahreszinsen: 180.0
   Jahrestilgung: 1020.0
   Restschuld: 7980.0
Jahr 3:
   Jahreszinsen: 159.6
   Jahrestilgung: 1040.4
   Restschuld: 6939.6
...
Jahr 10:
Jahreszinsen: 4.907431377689086
Jahrestilgung: 245.3715688844543
Restschuld: 0.0
```

Testen Sie Ihr Programm für 1) einen Darlehensbetrag von 10000 Euro, einen Zinssatz von 2.0% sowie einen Tilgungssatz von 10.0% und 2) einen Darlehensbetrag von 300000 Euro, einen Zinssatz von 3.5% sowie einen Tilgungssatz von 2.0%. Im Fall 1) soll die oben angegebene Ausgabe erfolgen.

Lösungsidee:
Wir benötigen folgende Variablen:

- Eine Variable `darlehensbetrag` vom Typ `double` für das Darlehen.
- Eine Variable `zinssatz` vom Typ `double`, die den Zinssatz in Prozent angibt.
- Eine Variable `tilgungssatz` vom Typ `double`, die den Tilgungssatz in Prozent angibt.
- Eine Variable `jahreszinsen` vom Typ `double` für die zu zahlenden Jahreszinsen.
- Eine Variable `jahrestilgung` vom Typ `double` für die Jahrestilgung.
- Eine Variable `jahresrate` vom Typ `double` für die zu zahlende Jahresrate.
- Eine Variable `restschuld` vom Typ `double` für die verbleibende Restschuld.

- Eine Variable `jahr` vom Typ `int` für das aktuelle Jahr.

Es gilt:

- `Jahresrate = Darlehensbetrag * (Zinssatz + Tilgungssatz)/100.0`.
- `Jahreszinsen im Jahr j = (Restschuld im Jahr j-1) * Zinssatz/100.0`.
- `Jahrestilgung im Jahr j`* `= Jahresrate - (Jahreszinsen im Jahr j)`.
- `Restschuld im Jahr j=(Restschuld im Jahr j-1 )-(Jahrestilgung im Jahr j)`.

**Gilt im letzten Jahr im Allgemeinen nicht.*
Vorgehensweise:

1. Deklaration der lokalen Variablen.
2. Berechnung der Jahresrate.
3. Setze `Restschuld = Darlehensbetrag`.
4. Setze `Jahr = 1`.
5. Solange `Restschuld > 0`:
   (a) Berechnung der Jahreszinsen
   (b) Berechnung der Jahrestilgung
   (c) Falls `Jahrestilgung > Restschuld`, dann setze `Jahrestilgung = Restschuld`
   (d) Setze `Restschuld = Restschuld - Jahrestilgung`
   (e) Ausgabe von Jahr, Jahreszinsen, Jahrestilgung und Restschuld
   (f) Erhöhe Jahr um 1.

Der Algorithmus wird folgendermaßen in Java umgesetzt.

```
1  public class Tilgungsrechner {
2    public static void main(String[] args) {
3
4      //Gegeben
5      double darlehensbetrag = 10000;
6      double zinssatz = 2.0;
7      double tilgungssatz = 10.0;
8
9      //Zu berechnen
10     double jahreszinsen;
11     double jahrestilgung;
12
13     double jahresrate = darlehensbetrag * (zinssatz + tilgungssatz)/100.0;
14     double restschuld = darlehensbetrag;
15     int jahr = 1;
16
17     while (restschuld > 0.0)
18     {
19       jahreszinsen = zinssatz/100.0 * restschuld;
20       jahrestilgung = jahresrate - jahreszinsen;
21       if (jahrestilgung >= restschuld) jahrestilgung = restschuld;
22       restschuld = restschuld - jahrestilgung;
23
24       System.out.println("Jahr " + jahr + ":");
25       System.out.println("   Jahreszinsen: " + jahreszinsen);
```

```
26        System.out.println("   Jahrestilgung: " + jahrestilgung);
27        System.out.println("   Restschuld: " + restschuld);
28
29        jahr++;
30      }
31    }
32 }
```

Aufgabe 5-3 **Wiederholungsanweisungen in Java (for-Schleife)** *Präsenz*

Sie möchten einen Bausparvertrag aufnehmen und berechnen, wieviel Bausparvermögen Sie nach einer vorgegebenen Zeit angespart haben. Die Verzinsung ist fix auf jährlich 4.5% festgelegt und wird jeweils am Ende des Jahres aufgeschlagen. Verträge sind nur mit einer jährlichen Laufzeit möglich. Die Einzahlung in den Bausparvertrag erfolgt monatlich mit einem festen Betrag.

Schreiben Sie in einer Klasse `Bausparkonto` ein Java-Programm, das für eine Laufzeit von 6 Jahren (72 Monaten) das am Ende des Zeitraums angesparte Vermögen berechnet, wenn monatlich 200 Euro eingezahlt werden. Verwenden Sie dazu als Iterationsanweisung eine `for`-Schleife. Bestimmen Sie zunächst, welche lokalen Variablen Sie benötigen (mit Typ).

Lösungsidee:
Wir benötigen folgende Variablen:

- Eine Variable `einzahlung` vom Typ `double` für die monatlichen Einzahlungen.
- Eine Variable `zinssatz` vom Typ `double`, die den Zinssatz in Prozent angibt.
- Eine Variable `ersparnisse` vom Typ `double`, die das angesparte Bausparvermögen angibt.
- Eine Variable `monate` vom Typ `int`, welche die Mindestlaufzeit in Monaten angibt.

Es gilt für die Berechnung des angesparten Vermögens:

- `In den ersten 11 Monaten: Ersparnisse = Ersparnisse + Einzahlung`.
- `Im zwölften Monat: Ersparnisse = (Ersparnisse + Einzahlung) + (Ersparnisse + Einzahlung) * Zinssatz`.

Der Algorithmus wird folgendermaßen in Java umgesetzt.

```
1 public class Bausparkonto {
2   public static void main(String[] args) {
3
4     //Gegeben
5     double einzahlung = 200;
6     double zinssatz = 4.5;
7     double ersparnisse = 0.0;
8     int monate = 72; //6 Jahre
9
10    for (int i = 1; i <= monate; i++) {
11
12      if (i%12 != 0) {
13        ersparnisse =  ersparnisse + einzahlung;
14      }
15      else {
16        double jahresvermoegen = (ersparnisse + einzahlung);
17        ersparnisse = jahresvermoegen + jahresvermoegen * zinssatz/100.0;
18      }
```

```
19
20        String ausgabe = "Monat: " + i + "   Angespartes Vermoegen: "
21           + ersparnisse;
22
23        if (i%12 == 0) ausgabe = ausgabe + "\n";
24
25        System.out.println(ausgabe);
26      }
27    }
28 }
```

Aufgabe 5-4 **Wiederholungsanweisungen in Java (for-Schleife)** *Hausaufgabe*

Bei Ihrem Supermarkt um die Ecke gibt es ein neues Bonuspunkte-System. Für jeden Tag, an dem Sie einkaufen gehen, erhalten Sie einen Bonuspunkt (unabhängig vom Betrag des Einkaufs). Gehen Sie aber an mehreren Tagen in Folge einkaufen, bekommen Sie pro Tag mehr Bonuspunkte, wie folgt: Am ersten Tag erhalten Sie einen Bonuspunkt. Am 2. und 3. Tag erhalten Sie je zwei Bonuspunkte; am 4. bis 6. Tag erhalten Sie je drei Bonuspunkt; am 7. bis 10. Tag erhalten Sie je vier Bonuspunkte usw. Das heißt, Sie erhalten einen Tag lang einen Bonuspunkt, zwei Tage lang zwei Bonuspunkte, drei Tage lang drei Bonuspunkte usw.

Schreiben Sie in einer Klasse `Bonuspunkte` ein Java-Programm, das berechnet, wie viele Tage hintereinander Sie einkaufen gehen müssen, um 100 Punkte **pro Tag** zu erhalten. Das Programm soll für jede Anzahl an Bonuspunkten ≤ 100 ausgeben, ab welchem Tag Sie diese Anzahl erhalten. Folgende Ausgabe sollte also produziert werden:

```
Ab Tag 1 erhält man pro Einkauf: 1 Bonuspunkte
Ab Tag 2 erhält man pro Einkauf: 2 Bonuspunkte
Ab Tag 4 erhält man pro Einkauf: 3 Bonuspunkte
Ab Tag 7 erhält man pro Einkauf: 4 Bonuspunkte
Ab Tag 11 erhält man pro Einkauf: 5 Bonuspunkte
Ab Tag 16 erhält man pro Einkauf: 6 Bonuspunkte
Ab Tag 22 erhält man pro Einkauf: 7 Bonuspunkte
...
Ab Tag 4852 erhält man pro Einkauf: 99 Bonuspunkte
Ab Tag 4951 erhält man pro Einkauf: 100 Bonuspunkte
```

Verwenden Sie dazu als Iterationsanweisung eine `for`-Schleife. Bestimmen Sie zunächst, welche lokalen Variablen Sie benötigen (mit Typ).

Wir benötigen folgende Variablen:

- Eine Variable `punkte` vom Typ `int` für die Anzahl der zu erreichenden Punkte.
- Eine Variable `tage` vom Type `int` für die Anzahl an Tagen, die in Folge eingekauft wurde.

```
1 public class Bonuspunkte {
2   public static void main(String[] args) {
3     int punkte = 100;
4     int tage = 1;
5
6     for (int i = 1; i <= punkte; i++) {
7       System.out.println("Ab Tag " + tage + " erhaelt man pro Einkauf: "
8           + i + " Bonuspunkte");
9       tage = tage + i;
10    }
11  }
12 }
```

Besprechung der Präsenzaufgaben in den Übungen ab 16.11.2016. Abgabe der Hausaufgaben bis Mittwoch, 30.11.2016, 14:00 Uhr über UniworX (siehe Folien der ersten Zentralübung).

- *Erstellen Sie zu jeder Aufgabe eine Klasse, die den Namen trägt, der in der Aufgabe gefordert ist.*
- *Geben Sie nur die entsprechenden .java-Dateien ab. Wir benötigen* ***nicht*** *Ihre .class-Dateien.*