

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

Aufgabe 7-1 Statische Methoden, Konstanten, Benutzereingaben *Präsenz*

Nehmen Sie für diese Aufgabe die im folgenden abgedruckte Klasse `Investitionsrechner` von Übungsblatt 5, Aufgabe 5-1, zur Grundlage und modifizieren Sie diese wie folgt.

```
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 + verkaufssteigerung
29                / 100.0 * anzahlVerkaufterAutos);
30        }
31    }
32 }
```

- a) Definieren Sie in einer Klasse `InvestitionsrechnerA` zunächst die Herstellungskosten von 22500 Euro für ein Auto, den Verkaufspreis von 25000 Euro sowie die zu erwartende prozentuale Verkaufssteigerung von 5% als geeignete Konstanten. Informieren Sie sich zudem in der Java-Dokumentation unter <http://docs.oracle.com/javase/8/docs/api/>, wie Sie mit Hilfe der Klasse `JOptionPane` der Bibliothek `javax.swing` eine Benutzereingabe realisieren können. Die Werte der Variablen `investitionsbetrag` sowie `anzahlVerkaufterAutos` sollen in der Klasse `InvestitionsrechnerA` über eine solche Benutzereingabe eingelesen werden können. Alle weiteren benötigten Variablen sind wie bisher als lokale Variablen zu realisieren. Die Ausgabe der Berechnungen soll unverändert auf der Konsole stattfinden. Testen Sie Ihr Programm, indem Sie beim Programmlauf folgende Werte eingeben: 1) einen Investitionsbetrag von einer Million Euro und eine Anzahl von 50 im ersten Jahr verkauften Autos, 2) einen Investitionsbetrag von 25 Millionen Euro und eine Anzahl von 1000 im ersten Jahr verkauften Autos.

- b) Schreiben Sie eine weitere Version der Klasse und nennen Sie diese `InvestitionsrechnerB`. In dieser Klasse soll die Berechnung des Gesamtgewinns sowie der verstrichenen Jahre (und deren Ausgabe) in eine eigene Methode `amortisierung` mit geeigneten formalen Parametern ausgelagert werden.

Aufgabe 7-2 Statische Methoden, Konstanten, Benutzereingaben *Hausaufgabe*

Nehmen Sie sich für diese Aufgabe die Klasse `Tilgungsrechner` von Übungsblatt 5, Aufgabe 5-2, zur Grundlage und modifizieren Sie diese wie folgt.

- a) Definieren Sie in einer Klasse `TilgungsrechnersrechnerA` zunächst einen Zinssatz von 2.0% sowie den Tilgungssatz von 10% als geeignete Konstanten. Der Wert der Variablen `darlehensbetrag` soll in der Klasse `TilgungsrechnersrechnerA` über eine Benutzereingabe mit Hilfe der Klasse `JOptionPane` im Paket `javax.swing` eingelesen werden können. Alle weiteren benötigten Variablen sind wie bisher als lokale Variablen zu realisieren. Die Ausgabe der Berechnungen soll unverändert auf der Konsole stattfinden. Testen Sie Ihr Programm, indem Sie beim Programmlauf einmal einen Darlehensbetrag von 10 000 Euro und einmal einen Darlehensbetrag von 200 000 Euro eingeben.
- b) Schreiben Sie eine weitere Version der Klasse und nennen Sie diese `TilgungsrechnerB`. In dieser Klasse sollen die mathematischen Berechnungen und Ausgaben in eine eigene Methode `restschuld` ausgelagert werden, die einen formalen Parameter für die Übergabe des eingegebenen Darlehensbetrags hat.

Aufgabe 7-3 Klassen, Vererbung, Benutzereingaben *Präsenz*

Nehmen Sie für diese Aufgabe die unten abgedruckte Klasse `Figur` von Übungsblatt 6, Aufgabe 6-3, zur Grundlage. Eine geometrische Figur soll wie bisher die Eigenschaften Mittelpunkt, Farbe und Füllung haben. Mögliche Spezialisierungen einer solchen Figur sind Kreise und Rechtecke, wobei sich Kreise durch einen Radius und Rechtecke durch Höhe und Breite genauer beschreiben lassen. Definieren Sie zusätzlich zur bestehenden Klasse `Figur` zwei Unterklassen `Kreis` und `Rechteck` zur Darstellung von Kreisen und Rechtecken. Beachten Sie, dass die Klasse `Figur` wieder die vorgegebene Klasse `Point` verwendet, welche Sie von der Webseite herunterladen können. Schreiben Sie eine Hauptklasse `MainFigur`, welche die drei definierten Klassen verwendet und über eine grafische Benutzerabfrage zuerst fragt, welches Objekt (Figur, Kreis oder Rechteck) erzeugt werden soll. Der Benutzer wird solange gefragt, bis er syntaktisch korrekt eines der Worte „Figur“, „Kreis“ oder „Rechteck“ eingibt. Verwenden Sie für diese Benutzereingabe die Klasse `JOptionPane` der Bibliothek `javax.swing`. Im Hauptprogramm sollen ebenfalls alle für die Erzeugung und Initialisierung des Objektes benötigten Eigenschaften über eine solche Benutzereingabe abgefragt und eingegeben werden. Das Hauptprogramm soll auf der Konsole ausgeben, welches Objekt gewählt und somit erzeugt wurde, und welche Koordinaten dessen Mittelpunkt hat. Anschließend soll das Objekt um bestimmte einzulesende Werte bewegt werden, woraufhin die Koordinaten des neuen Mittelpunkts ausgegeben werden sollen.

Hinweis: Alle deklarierten öffentlichen (public) Klassen müssen in eigenen Dateien abgespeichert werden.

```
1 /**
2  * Diese Klasse repraesentiert eine geometrische Figur.
3  *
4  * @author Annabelle Klarl
5  *
6  */
7 public class Figur {
8     private Point mittelpunkt;
9     private String farbe;
10    private boolean ausgefuellt;
11
12    /**
13     * Konstruktor einer geometrischen Figur, wobei deren Mittelpunkt und die
```

```

14  * Farbe gegeben sein muessen sowie ob die Figur ausgefuellt ist oder nicht.
15  *
16  * @param p
17  *         der Mittelpunkt der Figur
18  * @param farbe
19  *         die Farbe der Figur
20  * @param ausgefuellt
21  *         ob die Figur ausgefuellt ist oder nicht
22  */
23  public Figur(Point p, String farbe, boolean ausgefuellt) {
24      this.mittelpunkt = p;
25      this.farbe = farbe;
26      this.ausgefuellt = ausgefuellt;
27  }
28
29  /**
30  * Konstruktor einer geometrischen Figur, wobei deren Mittelpunkt und die
31  * Farbe gegeben sein muessen sowie ob die Figur ausgefuellt ist oder nicht.
32  *
33  * @param x
34  *         x-Koordinate des Mittelpunkts der Figur
35  * @param y
36  *         y-Koordinate des Mittelpunkts der Figur
37  * @param farbe
38  *         die Farbe der Figur
39  * @param ausgefuellt
40  *         ob die Figur ausgefuellt ist oder nicht
41  */
42  public Figur(int x, int y, String farbe, boolean ausgefuellt) {
43      this.mittelpunkt = new Point(x, y);
44      this.farbe = farbe;
45      this.ausgefuellt = ausgefuellt;
46  }
47
48  /**
49  * Diese Methode versetzt den Mittelpunkt der Figur
50  *
51  * @param dx
52  *         Versatz in x-Richtung
53  * @param dy
54  *         Versatz in y-Richtung
55  */
56  public void bewegen(int dx, int dy) {
57      this.mittelpunkt.move(dx, dy);
58  }
59
60  /**
61  * Diese Methode gibt den Mittelpunkt der Figur zurueck
62  *
63  * @return der Mittelpunkt
64  */
65  public Point getMittelpunkt() {
66      return this.mittelpunkt;
67  }
68 }

```

Aufgabe 7-4

Klassen, Vererbung, Benutzereingaben

Hausaufgabe

Nehmen Sie für diese Aufgabe die Klasse `Fahrzeug` von Übungsblatt 6, Aufgabe 6-4, zur Grundlage. Ein Fahrzeug soll wie bisher die Eigenschaften `Position`, `Anzahl der Räder`, `Leergewicht` und `aktuelle Geschwindigkeit` haben. Mögliche Spezialisierungen eines Fahrzeugs sind `Fahrräder` und `Personenkraftwagen`, wobei sich `Fahrräder` dadurch auszeichnen, dass sie nur zwei Räder haben, und `Personenkraftwagen` dadurch, dass sie vier Räder sowie eine `Motorleistung` vom Typ `int` haben. Definieren Sie zusätzlich zur bestehenden Klasse `Fahrzeug` zwei Unterklassen `Fahrrad` und `Personenkraftwagen` zur Beschreibung dieser konkreteren Fahrzeuge. Beachten Sie, dass die

Klasse `Fahrzeug` wieder die vorgegebene Klasse `Point` verwendet, welche Sie von der Webseite herunterladen können. Schreiben Sie eine Hauptklasse `MainFahrzeug`, welche die drei definierten Klassen verwendet und über eine grafische Benutzerabfrage zuerst fragt, welches Objekt (Fahrzeug, Fahrrad oder Personenkraftwagen) erzeugt werden soll. Der Benutzer wird solange gefragt, bis er syntaktisch korrekt eines der Worte „Fahrzeug“, „Fahrrad“ oder „PKW“ eingibt. Verwenden Sie für diese Benutzereingabe die Klasse `JOptionPane` der Bibliothek `javax.swing`. Im Hauptprogramm sollen ebenfalls alle für die Erzeugung und Initialisierung des Objekts benötigten Eigenschaften über eine solche Benutzereingabe abgefragt und eingelesen werden. Beachten Sie, dass die Anzahl der Räder jedoch nur bei Objekten des Typs `Fahrzeug` einzulesen ist, da Fahrräder und Personenkraftwagen immer zwei bzw. vier Räder haben. Ebenso muss auch die Motorleistung nur bei Objekten des Typs `Personenkraftwagen` eingelesen werden. Das Hauptprogramm soll auf der Konsole ausgeben, welches Objekt gewählt und somit erzeugt wurde, und welche Geschwindigkeit es hat. Anschließend soll das Objekt um einen bestimmten einzulesenden Wert beschleunigt werden, woraufhin die neue Geschwindigkeit ausgegeben werden soll.

Hinweis: Alle deklarierten öffentlichen (`public`) Klassen müssen in eigenen Dateien abgespeichert werden.

Besprechung der Präsenzaufgaben in den Übungen ab 08.12.2017. Abgabe der Hausaufgaben bis Mittwoch, 20.12.2017, 14:00 Uhr über UniworX (siehe Folien der ersten Zentralübung).

- *Erstellen Sie zu jeder Aufgabe Klassen mit den entsprechenden Namen, die in der Aufgabe gefordert sind.*
- *Geben Sie nur die entsprechenden `.java`-Dateien ab. Wir benötigen **nicht** Ihre `.class`-Dateien.*
- *Geben Sie Java-Code nur in `.java`-Dateien ab. Java-Code in Bildern, PDF-Dokumenten und Text-Dateien wird nicht korrigiert.*