

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

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

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

- 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/7/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.

Lösungsidee:

Die Klasse `InvestitionsrechnerA` sieht folgendermaßen aus.

```
1 import javax.swing.JOptionPane ;
2
3 public class InvestitionsrechnerA {
4
5     public static final double HERSTELLUNGSKOSTEN = 22500;
6     public static final double VERKAUFSPREIS = 25000;
7     public static final double VERKAUFSSTEIGERUNG = 5;
8
9     public static void main(String[] args) {
10        // Einzulesen
11        String einlesenInvestitionsbetrag = JOptionPane
12            .showInputDialog("Investitionsbetrag eingeben : ");
13        double investitionsbetrag = Double
14            .parseDouble(einlesenInvestitionsbetrag);
15
16        String einlesenAnzahlVerkaufterAutos = JOptionPane
17            .showInputDialog("Anzahl verkaufter Autos eingeben : ");
18        int anzahlVerkaufterAutos = Integer
19            .parseInt(einlesenAnzahlVerkaufterAutos);
20
21        // Zu berechnen
22        double gesamtGewinn = 0.0;
23        int anzahlJahre = 1;
24
25        while (gesamtGewinn < investitionsbetrag) {
26            double kostenProJahr = HERSTELLUNGSKOSTEN
27                * anzahlVerkaufterAutos;
28            double umsatzProJahr = VERKAUFSPREIS
29                * anzahlVerkaufterAutos;
```

```

30
31     double gewinnProJahr = umsatzProJahr - kostenProJahr;
32
33     gesamtGewinn = gesamtGewinn + gewinnProJahr;
34
35     System.out.println("Jahr " + anzahlJahre + ":");
36     System.out.println("    Anzahl verkaufter Autos: "
37         + anzahlVerkaufteAutos);
38     System.out.println("    Kosten: " + kostenProJahr);
39     System.out.println("    Umsatz: " + umsatzProJahr);
40     System.out.println("    Gewinn: " + gewinnProJahr);
41     System.out.println("    Gesamtgewinn: " + gesamtGewinn);
42
43     anzahlJahre++;
44     anzahlVerkaufteAutos = (int) (anzahlVerkaufteAutos
45         + VERKAUFSSTEIGERUNG / 100.0 * anzahlVerkaufteAutos);
46 }
47 }
48 }

```

- 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.

Lösungsidee:

Die verbesserte Version in der Klasse `InvestitionsrechnerB` sieht folgendermaßen aus:

```

1 import javax.swing.JOptionPane;
2
3 public class InvestitionsrechnerB {
4
5     public static final double HERSTELLUNGSKOSTEN = 22500;
6     public static final double VERKAUFSPREIS = 25000;
7     public static final double VERKAUFSSTEIGERUNG = 5;
8
9     public static void main(String[] args) {
10         // Einzulesen
11         String einlesenInvestitionsbetrag = JOptionPane
12             .showInputDialog("Investitionsbetrag eingeben : ");
13         double investitionsbetrag = Double
14             .parseDouble(einlesenInvestitionsbetrag);
15
16         String einlesenAnzahlVerkaufteAutos = JOptionPane
17             .showInputDialog("Anzahl verkaufter Autos eingeben : ");
18         int anzahlVerkaufteAutos = Integer
19             .parseInt(einlesenAnzahlVerkaufteAutos);
20
21         // Berechnung
22         amortisierung(investitionsbetrag, anzahlVerkaufteAutos);
23     }
24
25     public static void amortisierung(double investition, int autos) {
26         double investitionsbetrag = investition;
27         int anzahlVerkaufteAutos = autos;
28
29         double gesamtGewinn = 0.0;
30         int anzahlJahre = 1;
31
32         while (gesamtGewinn < investitionsbetrag) {
33             double kostenProJahr = HERSTELLUNGSKOSTEN
34                 * anzahlVerkaufteAutos;

```

```

35         double umsatzProJahr = VERKAUFSPREIS
36             * anzahlVerkaufterAutos;
37
38         double gewinnProJahr = umsatzProJahr - kostenProJahr;
39
40         gesamtGewinn = gesamtGewinn + gewinnProJahr;
41
42         System.out.println("Jahr " + anzahlJahre + ":");
43         System.out.println("    Anzahl verkaufter Autos: "
44             + anzahlVerkaufterAutos);
45         System.out.println("    Kosten: " + kostenProJahr);
46         System.out.println("    Umsatz: " + umsatzProJahr);
47         System.out.println("    Gewinn: " + gewinnProJahr);
48         System.out.println("    Gesamtgewinn: " + gesamtGewinn);
49
50         anzahlJahre++;
51         anzahlVerkaufterAutos = (int) (anzahlVerkaufterAutos
52             + VERKAUFSSTEIGERUNG / 100.0 * anzahlVerkaufterAutos);
53     }
54 }
55 }

```

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.

- 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.
- 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 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.

Lösungsidee:

Die beteiligten Klassen `Kreis` und `Rechteck` sowie das Hauptprogramm `MainFigur` sehen wie folgt aus. Die Klassen `Figur` und `Point` können Sie unverändert von Übungsblatt 6 übernehmen.

```
1 // Klasse Kreis
2 public class Kreis extends Figur {
3     private double radius;
4
5     public Kreis(int x, int y, String farbe, boolean ausgefuellt,
6                 double radius) {
7         super(x, y, farbe, ausgefuellt);
8         this.radius = radius;
9     }
10
11    public double getRadius() {
12        return this.radius;
13    }
14 }
15
16 // Klasse Rechteck
17 public class Rechteck extends Figur {
18     private double hoehe;
19     private double breite;
20
21    public Rechteck(int x, int y, String farbe, boolean ausgefuellt,
22                  double hoehe, double breite) {
23        super(x, y, farbe, ausgefuellt);
24        this.hoehe = hoehe;
25        this.breite = breite;
26    }
27
28    public double getHoehe() {
29        return this.hoehe;
30    }
31
32    public double getBreite() {
33        return this.breite;
34    }
35 }
36
37 // Klasse MainFigur
38 import javax.swing.JOptionPane;
39
40 public class MainFigur {
41
42    public static void main(String[] args) {
43        Figur figur = null;
44
45        String einlesenObjekt = JOptionPane
46            .showInputDialog("Was wollen Sie erzeugen "
47                + "[Figur, Kreis, Rechteck]: ");
48
49        while (!einlesenObjekt.equals("Figur")
50            && !einlesenObjekt.equals("Kreis")
51            && !einlesenObjekt.equals("Rechteck")) {
52            einlesenObjekt = JOptionPane
53                .showInputDialog("Was wollen Sie erzeugen "
```

```

54         + "[Figur, Kreis, Rechteck]: ");
55     }
56
57     String einlesenPositionX = JOptionPane
58         .showInputDialog("Mittelpunkt x-Position: ");
59     int xpos = Integer.parseInt(einlesenPositionX);
60
61     String einlesenPositionY = JOptionPane
62         .showInputDialog("Mittelpunkt y-Position: ");
63     int ypos = Integer.parseInt(einlesenPositionY);
64
65     String farbe = JOptionPane.showInputDialog("Farbe: ");
66
67     String einlesenFuellung = JOptionPane
68         .showInputDialog("Füllung [true/false]: ");
69     boolean fuellung = Boolean.parseBoolean(einlesenFuellung);
70
71     if (einlesenObjekt.equals("Figur")) {
72         figur = new Figur(xpos, ypos, farbe, fuellung);
73     }
74     else if (einlesenObjekt.equals("Kreis")) {
75         String einlesenRadius = JOptionPane
76             .showInputDialog("Radius: ");
77         double radius = Double.parseDouble(einlesenRadius);
78         figur = new Kreis(xpos, ypos, farbe, fuellung, radius);
79     }
80     else if (einlesenObjekt.equals("Rechteck")) {
81         String einlesenHoehe = JOptionPane
82             .showInputDialog("Höhe: ");
83         double hoehe = Double.parseDouble(einlesenHoehe);
84         String einlesenBreite = JOptionPane
85             .showInputDialog("Breite: ");
86         double breite = Double.parseDouble(einlesenBreite);
87         figur = new Rechteck(xpos, ypos, farbe, fuellung, hoehe,
88             breite);
89     }
90
91     System.out.println("Wir haben folgendes Objekt erzeugt: "
92         + figur + " mit dem Mittelpunkt: "
93         + figur.getMittelpunkt().getX() + ", "
94         + figur.getMittelpunkt().getY());
95
96     String einlesenDx = JOptionPane
97         .showInputDialog("Objekt bewegen um x-Wert: ");
98     int dx = Integer.parseInt(einlesenDx);
99
100    String einlesenDy = JOptionPane
101        .showInputDialog("Objekt bewegen um y-Wert: ");
102    int dy = Integer.parseInt(einlesenDy);
103
104    figur.bewegen(dx, dy);
105
106    System.out.println("Das Objekt " + figur
107        + " hat nun den Mittelpunkt: "
108        + figur.getMittelpunkt().getX() + ", "
109        + figur.getMittelpunkt().getY());
110 }
111 }

```

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 30.11.2016. Abgabe der Hausaufgaben bis Mittwoch, 14.12.2016, 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 Ihren Code **NICHT** in einer `.pdf-Datei` oder `.class-Datei` ab, sondern in einer `.java-Datei`