

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 im folgenden abgedruckte Klasse `Investitionsrechner` von Übungsblatt 5, Aufgabe 5-1, zur Grundlage und modifizieren Sie diese schrittweise wie in den Teilaufgaben angegeben.

```
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.

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                 + anzahlVerkaufterAutos);
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             anzahlVerkaufterAutos = (int) (anzahlVerkaufterAutos
45                 + VERKAUFSSTEIGERUNG / 100.0 * anzahlVerkaufterAutos);
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 einlesenAnzahlVerkaufterAutos = JOptionPane
17             .showInputDialog("Anzahl verkaufter Autos eingeben : ");
18         int anzahlVerkaufterAutos = Integer
19             .parseInt(einlesenAnzahlVerkaufterAutos);
20
21         // Berechnung
22         amortisierung(investitionsbetrag, anzahlVerkaufterAutos);
23     }
24
25     public static void amortisierung(double investition, int autos) {
26         double investitionsbetrag = investition;
27         int anzahlVerkaufterAutos = autos;
28
29         double gesamtGewinn = 0.0;
30         int anzahlJahre = 1;
31
32         while (gesamtGewinn < investitionsbetrag) {
33             double kostenProJahr = HERSTELLUNGSKOSTEN
34                 * anzahlVerkaufterAutos;
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 schrittweise wie in den Teilaufgaben angegeben. Die Klasse ist auch im ZIP-Archiv zur Angabe vorhanden.

- 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.

Lösungsidee:

Die Klasse `TilgungsrechnerA` sieht folgendermaßen aus.

```
1 import javax.swing.JOptionPane;
2
3 public class TilgungsrechnerA {
4
5     public static final double ZINSSATZ = 2.0;
6     public static final double TILGUNGSSATZ = 10.0;
7
8     public static void main(String[] args) {
9         // Einzulesen
10        String einlesenDarlehensbetrag = JOptionPane
11            .showInputDialog("Darlehen eingeben : ");
12        double darlehensbetrag = Double
13            .parseDouble(einlesenDarlehensbetrag);
14
15        // Zu berechnen
16        double jahreszinsen;
17        double jahrestilgung;
18
19        double jahresrate = darlehensbetrag
20            * (ZINSSATZ + TILGUNGSSATZ) / 100.0;
21        double restschuld = darlehensbetrag;
22        int jahr = 1;
23
24        while (restschuld > 0.0) {
25            jahreszinsen = ZINSSATZ / 100.0 * restschuld;
26            jahrestilgung = jahresrate - jahreszinsen;
27            if (jahrestilgung >= restschuld) {
28                jahrestilgung = restschuld;
29            }
30            restschuld = restschuld - jahrestilgung;
31
32            System.out.println("Jahr " + jahr + ":");
33            System.out.println("    Jahreszinsen: " + jahreszinsen);
34            System.out.println("    Jahrestilgung: " + jahrestilgung);
35            System.out.println("    Restschuld: " + restschuld);
36
37            jahr++;
38        }
39    }
40 }
```

- 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 Me-

thode `restschuld` ausgelagert werden, die einen formalen Parameter für die Übergabe des eingegebenen Darlehensbetrags hat.

Lösungsidee:

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

```
1 import javax.swing.JOptionPane;
2
3 public class TilgungsrechnerB {
4
5     public static final double ZINSSATZ = 2.0;
6     public static final double TILGUNGSSATZ = 10.0;
7
8     public static void main(String[] args) {
9         // Einzulesen
10        String einlesenDarlehensbetrag = JOptionPane
11            .showInputDialog("Darlehen eingeben : ");
12        double darlehensbetrag = Double
13            .parseDouble(einlesenDarlehensbetrag);
14
15        // Berechnung
16        restschuld(darlehensbetrag);
17    }
18
19    public static void restschuld(double darlehensbetrag) {
20        double jahreszinsen;
21        double jahrestilgung;
22
23        double jahresrate = darlehensbetrag
24            * (ZINSSATZ + TILGUNGSSATZ) / 100.0;
25        double restschuld = darlehensbetrag;
26        int jahr = 1;
27
28        while (restschuld > 0.0) {
29            jahreszinsen = ZINSSATZ / 100.0 * restschuld;
30            jahrestilgung = jahresrate - jahreszinsen;
31            if (jahrestilgung >= restschuld) {
32                jahrestilgung = restschuld;
33            }
34            restschuld = restschuld - jahrestilgung;
35
36            System.out.println("Jahr " + jahr + ":");
37            System.out.println("    Jahreszinsen: " + jahreszinsen);
38            System.out.println("    Jahrestilgung: " + jahrestilgung);
39            System.out.println("    Restschuld: " + restschuld);
40
41            jahr++;
42        }
43    }
44 }
```

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. 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 „Recht-

eck“ 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.

Beachten Sie, dass die Klasse `Figur` wieder die vorgegebene Klasse `Point` verwendet. Beide Klassen sind im ZIP-Archiv zur Angabe vorhanden.

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

```
1  /**
2   * Diese Klasse repraesentiert eine geometrische Figur.
3   */
4  public class Figur {
5      private Point mittelpunkt;
6      private String farbe;
7      private boolean ausgefuellt;
8
9      /**
10     * Konstruktor einer geometrischen Figur, wobei deren Mittelpunkt und die
11     * Farbe gegeben sein muessen sowie ob die Figur ausgefuellt ist oder nicht.
12     *
13     * @param p
14     *         der Mittelpunkt der Figur
15     * @param farbe
16     *         die Farbe der Figur
17     * @param ausgefuellt
18     *         ob die Figur ausgefuellt ist oder nicht
19     */
20     public Figur(Point p, String farbe, boolean ausgefuellt) {
21         this.mittelpunkt = p;
22         this.farbe = farbe;
23         this.ausgefuellt = ausgefuellt;
24     }
25
26     /**
27     * Konstruktor einer geometrischen Figur, wobei deren Mittelpunkt und die
28     * Farbe gegeben sein muessen sowie ob die Figur ausgefuellt ist oder nicht.
29     *
30     * @param x
31     *         x-Koordinate des Mittelpunkts der Figur
32     * @param y
33     *         y-Koordinate des Mittelpunkts der Figur
34     * @param farbe
35     *         die Farbe der Figur
36     * @param ausgefuellt
37     *         ob die Figur ausgefuellt ist oder nicht
38     */
39     public Figur(int x, int y, String farbe, boolean ausgefuellt) {
40         this.mittelpunkt = new Point(x, y);
41         this.farbe = farbe;
42         this.ausgefuellt = ausgefuellt;
43     }
44
45     /**
46     * Diese Methode versetzt den Mittelpunkt der Figur
47     *
48     * @param dx
49     *         Versatz in x-Richtung
50     * @param dy
51     *         Versatz in y-Richtung
52     */
```

```

53     public void bewegen(int dx, int dy) {
54         this.mittelpunkt.move(dx, dy);
55     }
56
57     /**
58      * Diese Methode gibt den Mittelpunkt der Figur zurueck
59      *
60      * @return der Mittelpunkt
61      */
62     public Point getMittelpunkt() {
63         return this.mittelpunkt;
64     }
65 }

```

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 public class Kreis extends Figur {
2     private double radius;
3
4     public Kreis(int x, int y, String farbe, boolean ausgefuellt,
5                 double radius) {
6         super(x, y, farbe, ausgefuellt);
7         this.radius = radius;
8     }
9
10    public double getRadius() {
11        return this.radius;
12    }
13 }

```

```

1 public class Rechteck extends Figur {
2     private double hoehe;
3     private double breite;
4
5     public Rechteck(int x, int y, String farbe, boolean ausgefuellt,
6                    double hoehe, double breite) {
7         super(x, y, farbe, ausgefuellt);
8         this.hoehe = hoehe;
9         this.breite = breite;
10    }
11
12    public double getHoehe() {
13        return this.hoehe;
14    }
15
16    public double getBreite() {
17        return this.breite;
18    }
19 }

```

```

1 import javax.swing.JOptionPane;
2
3 public class MainFigur {
4
5     public static void main(String[] args) {
6         Figur figur = null;
7
8         String einlesenObjekt = JOptionPane
9             .showInputDialog("Was wollen Sie erzeugen "
10                             + "[Figur, Kreis, Rechteck]: ");
11

```

```

12     while (!einlesenObjekt.equals("Figur")
13           && !einlesenObjekt.equals("Kreis")
14           && !einlesenObjekt.equals("Rechteck")) {
15         einlesenObjekt = JOptionPane
16             .showInputDialog("Was wollen Sie erzeugen "
17                             + "[Figur, Kreis, Rechteck]: ");
18     }
19
20     String einlesenPositionX = JOptionPane
21         .showInputDialog("Mittelpunkt x-Position: ");
22     int xpos = Integer.parseInt(einlesenPositionX);
23
24     String einlesenPositionY = JOptionPane
25         .showInputDialog("Mittelpunkt y-Position: ");
26     int ypos = Integer.parseInt(einlesenPositionY);
27
28     String farbe = JOptionPane.showInputDialog("Farbe: ");
29
30     String einlesenFuellung = JOptionPane
31         .showInputDialog("Fuellung [true/false]: ");
32     boolean fuellung = Boolean.parseBoolean(einlesenFuellung);
33
34     if (einlesenObjekt.equals("Figur")) {
35         figur = new Figur(xpos, ypos, farbe, fuellung);
36     }
37     else if (einlesenObjekt.equals("Kreis")) {
38         String einlesenRadius = JOptionPane
39             .showInputDialog("Radius: ");
40         double radius = Double.parseDouble(einlesenRadius);
41         figur = new Kreis(xpos, ypos, farbe, fuellung, radius);
42     }
43     else if (einlesenObjekt.equals("Rechteck")) {
44         String einlesenHoehe = JOptionPane
45             .showInputDialog("Hoehe: ");
46         double hoehe = Double.parseDouble(einlesenHoehe);
47         String einlesenBreite = JOptionPane
48             .showInputDialog("Breite: ");
49         double breite = Double.parseDouble(einlesenBreite);
50         figur = new Rechteck(xpos, ypos, farbe, fuellung, hoehe,
51                             breite);
52     }
53
54     System.out.println("Wir haben folgendes Objekt erzeugt: "
55                       + figur + " mit dem Mittelpunkt: "
56                       + figur.getMittelpunkt().getX() + ", "
57                       + figur.getMittelpunkt().getY());
58
59     String einlesenDx = JOptionPane
60         .showInputDialog("Objekt bewegen um x-Wert: ");
61     int dx = Integer.parseInt(einlesenDx);
62
63     String einlesenDy = JOptionPane
64         .showInputDialog("Objekt bewegen um y-Wert: ");
65     int dy = Integer.parseInt(einlesenDy);
66
67     figur.bewegen(dx, dy);
68
69     System.out.println("Das Objekt " + figur
70                       + " hat nun den Mittelpunkt: "
71                       + figur.getMittelpunkt().getX() + ", "
72                       + figur.getMittelpunkt().getY());
73 }
74 }

```


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. 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.

Beachten Sie, dass die Klasse **Fahrzeug** wieder die vorgegebene Klasse **Point** verwendet. Beide Klassen sind im ZIP-Archiv zur Angabe vorhanden.

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

Lösungsidee:

Die beteiligten Klassen **Fahrrad** und **Personenkraftwagen** sowie das Hauptprogramm **MainFahrzeug** sehen wie folgt aus. Die Klasse **Point** und **Fahrzeug** können Sie unverändert von Übungsblatt 6 übernehmen.

```
1 public class Fahrrad extends Fahrzeug {
2     public Fahrrad(int x, int y, double gewicht,
3         double geschwindigkeit) {
4         super(x, y, 2, gewicht, geschwindigkeit);
5     }
6 }
```

```
1 public class Personenkraftwagen extends Fahrzeug {
2     private int motorleistung;
3
4     public Personenkraftwagen(int x, int y, double gewicht,
5         double geschwindigkeit, int motorleistung) {
6         super(x, y, 4, gewicht, geschwindigkeit);
7         this.motorleistung = motorleistung;
8     }
9 }
```

```
1 import javax.swing.JOptionPane;
2
3 public class MainFahrzeug {
4
5     public static void main(String[] args) {
6
7         Fahrzeug fahrzeug = null;
8
9         String einlesenObjekt = JOptionPane
```

```

10         .showInputDialog("Was wollen Sie erzeugen "
11             + "[Fahrzeug, Fahrrad, PKW]: ");
12
13     while (!einlesenObjekt.equals("Fahrzeug")
14         && !einlesenObjekt.equals("Fahrrad")
15         && !einlesenObjekt.equals("PKW")) {
16         einlesenObjekt = JOptionPane
17             .showInputDialog("Was wollen Sie erzeugen "
18                 + "[Fahrzeug, Fahrrad, PKW]: ");
19     }
20
21     String einlesenPositionX = JOptionPane
22         .showInputDialog("Aktuelle x-Position: ");
23     int xpos = Integer.parseInt(einlesenPositionX);
24
25     String einlesenPositionY = JOptionPane
26         .showInputDialog("Aktuelle y-Position: ");
27     int ypos = Integer.parseInt(einlesenPositionY);
28
29     String einlesenGewicht =
30         JOptionPane.showInputDialog("Gewicht: ");
31     double gewicht = Double.parseDouble(einlesenGewicht);
32
33     String einlesenGeschwindigkeit = JOptionPane
34         .showInputDialog("Geschwindigkeit: ");
35     double geschwindigkeit =
36         Double.parseDouble(einlesenGeschwindigkeit);
37
38     if (einlesenObjekt.equals("Fahrzeug")) {
39         String einlesenRaeder =
40             JOptionPane.showInputDialog("Raeder: ");
41         int raeder = Integer.parseInt(einlesenRaeder);
42
43         fahrzeug = new Fahrzeug(xpos, ypos, raeder, gewicht,
44             geschwindigkeit);
45     } else if (einlesenObjekt.equals("Fahrrad")) {
46         fahrzeug = new Fahrrad(xpos, ypos, gewicht, geschwindigkeit);
47     } else if (einlesenObjekt.equals("PKW")) {
48         String einlesenMotorleistung = JOptionPane
49             .showInputDialog("Motorleistung: ");
50         int motorleistung = Integer.parseInt(einlesenMotorleistung);
51
52         fahrzeug = new Personenkraftwagen(xpos, ypos, gewicht,
53             geschwindigkeit, motorleistung);
54     }
55
56     System.out.println("Wir haben folgendes Objekt erzeugt: "
57         + fahrzeug + " mit der aktuellen Geschwindigkeit: "
58         + fahrzeug.getAktuelleGeschwindigkeit());
59
60     String einlesenBeschleunigung = JOptionPane
61         .showInputDialog("Um wieviel moechten Sie beschleunigen: ");
62     double beschleunigung = Double.parseDouble(einlesenBeschleunigung);
63
64     fahrzeug.beschleunigen(beschleunigung);
65
66     System.out.println("Das Objekt: " + fahrzeug
67         + " hat nun die aktuelle Geschwindigkeit: "
68         + fahrzeug.getAktuelleGeschwindigkeit());
69 }
70 }

```

Besprechung der Präsenzaufgaben in den Übungen ab 30.11.2018. Abgabe der Hausaufgaben bis Mittwoch, 12.12.2018, 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.*