


| | | |
|-----------|--|--|
| Kurs: IT3 | AB 6 –Java-Übungen Einlesen, Ausgeben, Selektion, Schleifen |  |
| Datum: | | |

Situation

Im Rahmen eines Praktikums bei der Firma *BBS-IT* sollen Sie die Entwickler bei der Programmierung in Java unterstützen. Nachdem Sie sich schon ein wenig in die Programmierung mit Java eingearbeitet haben, testen (und festigen) Sie Ihr Wissen, indem Sie Programmierübungen zu den einzelnen Gebieten absolvieren.

Aufgabe 1

Schreiben Sie ein Programm *Subtraktion.java*, das folgende Ausgabe erzeugt, hierbei sind Nutzereingaben **fett** gedruckt):

Subtraktion zweier ganzer Zahlen

Zahl 1: **23**

Zahl 2: **4**

Das Ergebnis der Subtraktion 23 – 4 ist 19

Programmende Subtraktion.

Aufgabe 2

Schreiben Sie ein Programm *Punktabstand.java*, das den Abstand zweier Punkte $P_1(x_1; y_1)$ und $P_2(x_2; y_2)$ berechnet und ausgibt. Formel: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ (auf welchem Satz basiert diese Formel?)

Punktabstand

1. Punkt x: **8**

1. Punkt y: **3**

2. Punkt x: **4**

2. Punkt y: **-6**

Der Punktabstand ist: 9,85

Programmende Subtraktion

Aufgabe 3

Schreiben Sie ein Programm *Durchschnittsverbrauch.java*, das nach Eingabe der gefahrenen Kilometer und der Menge des verbrauchten Kraftstoffes den Durchschnittsverbrauch auf 100 km berechnet und ausgibt. Die Berechnung darf nur erfolgen, wenn die eingegebenen Werte größer als null sind, ansonsten soll eine sinnvolle Fehlermeldung ausgegeben werden. Außerdem soll bei unter 5 l je 100km ausgegeben werden, dass der Verbrauch gering ist, bei mehr als 8 l je 100 km, dass der Verbrauch hoch ist und dazwischen „normal“.

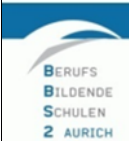
Berechnung Durchschnittsverbrauch

verbrauchter Kraftstoff in l: **41**

gefahrte Strecke in km: **750**

Der Verbrauch liegt bei 5,47 Litern je 100 km

Programmende Durchschnittsverbrauch

| | | |
|-----------|--|--|
| Kurs: IT3 | AB 6 –Java-Übungen Einlesen, Ausgeben, Selektion, Schleifen |  |
| Datum: | | |

Aufgabe 4

Schreiben Sie ein Programm `ZahlInfo.java`, das eine ganze Zahl einliest und danach prüft, ob sie größer, kleiner oder gleich null ist. Außerdem soll angegeben werden, ob es sich um eine gerade oder ungerade Zahl handelt.

`ZahlInfo`

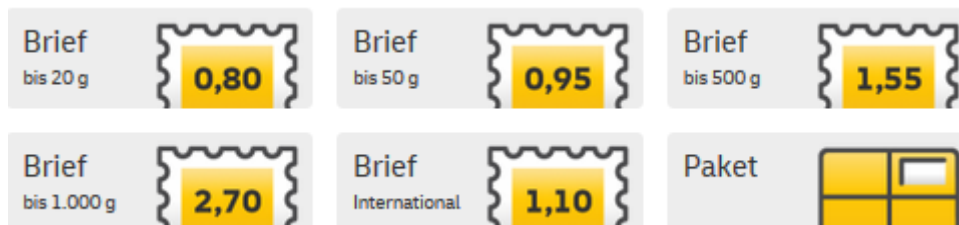
Zahl: 2

Die eingegebene Zahl ist grösser 0 und gerade.

Programmende `ZahlInfo`.

Aufgabe 5

Nationale Briefe werden nach ihrem Gewicht frankiert:



Schreiben Sie ein Programm `Briefporto.java`, das das Gewicht eines Briefes einliest und das entsprechende Porto ausgibt.

Aufgabe 6

(`Primzahl.java`) Lassen Sie prüfen, ob eine long-Zahl eine Primzahl ist. Dividieren Sie die zu prüfende Zahl durch $i = 2, 3$ usw. bis $i^2 > x$ ist und prüfen Sie, ob die Division $\frac{x}{i}$ einen Rest lässt.

`Primzahlpruefung`

Ganze Zahl eingeben: 66

66 ist keine Primzahl.

Programmende `Primzahl`.

Aufgabe 7

(`SummeUndDurchschnitt.java`) Lassen Sie eine beliebige Anzahl von Zahlen addieren, bis die Zahl 0 eingegeben wird. Danach wird die Gesamtsumme und der Durchschnitt ausgegeben.

`SummeUndDurchschnitt`

Zahl (Ende mit 0): 1.5

Zahl (Ende mit 0): 3

Zahl (Ende mit 0): 1.5

Zahl (Ende mit 0): 0

Summe: 6; Durchschnitt 2

Programmende `SummeUndDurchschnitt`.

Aufgabe 8

Lassen Sie den Rechner die Summe der folgenden Zahlen bis zu einem Grenzwert g berechnen und geben Sie die Anzahl der benötigten Glieder aus.

- a) (SumA.java) $1 + 2 + 3 + 4 + \dots + n + \dots$
- b) (SumB.java) $1 - 2 + 3 - 4 + 5 \dots + (2n + 1) - 2n + \dots$
- c) (SumC.java) $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} + \dots$
- d) (SumD.java) $1 + 4 + 9 + 16 + 25 + \dots + n^2 + \dots$

SumA

Berechnung der Summe 1+2+3+4+... bis zum Grenzwert

Grenzwert: 16

Nach 6 Gliedern ist 16 erreicht. Die Summe ist 21.

Programmende SumA.

Aufgabe 9

(Matrix.java) Es soll eine Multiplikationsmatrix auf dem Bildschirm (von 10 Zeilen/ 10 Spalten) angezeigt werden. Nach Eingabe einer Ziffer zwischen 2 und 9 werden alle Zahlen durch einen Stern gekennzeichnet, die diese Ziffer enthalten oder durch die eingegebene Zahl ohne Rest teilbar sind.

Matrix von 0 .. 99

Zahl (2..9): 4

```

0   1   2   3   *   5   6   7   *   9
10  11  *  13  *  15  *  17  18  19
*   21  *  23  *  25  26  27  *  29
30  31  *  33  *  35  *  37  38  39
*   *   *   *   *   *   *   *   *
50  .....
.....
90  91  *  93  *  95  *  97  98  99

```

Programmende Matrix.