

## 2 Rechnen in Java

*Ziel: Benutzung geeigneter Datentypen, korrekte Typkonvertierung, Verständnis für Rechenvorgänge mit den Datentypen*

- 2.1. Implementieren Sie das Struktogramm von Aufgabe 1.8 von Lsg\_Struktogramm.pdf. Formatieren Sie die Ausgabe so, dass die Ausgabe auf eine Nachkommastelle gerundet wird.

```
20 * A0201BMI.java
5  package strProgAufg;
6
7  import input.Eingabe;
8
9  /**
10   * @author stk
11   * Kurzbeschreibung: Programm berechnet den Body-Mass-Index
12   */
13  public class A0201BMI
14  {
15
16      * @param args
17      public static void main(String[] args)
18      {
19          int iGewicht; // Körpergewicht in kg
20          double dGroesse; // Körpergröße in m
21          double dBmi; // Body-Mass-Index
22
23          iGewicht = Eingabe.getInt("Geben Sie Ihr Körpergewicht in kg ein");
24          dGroesse = Eingabe.getDouble("Geben Sie Ihre Körpergröße in m ein");
25
26          dBmi = iGewicht / (dGroesse * dGroesse);
27
28          System.out.printf("BMI = %5.1f%n", dBmi);
29
30      }
31  }
```

```
2+ * A0202Verbrauch.java
5 package strProgAufg;
6
7 import input.Eingabe;
8
9- /**
10 * @author stk
11 * Kurzbeschreibung: Berechnet den Durchschnittsverbrauch eines
12 *                   Fahrzeugs in Liter/100 km
13 */
14 public class A0202Verbrauch
15 {
17+ * @param args
20- public static void main(String[] args)
21 {
22     int iStrecke; // Zurückgelegte Strecke in km
23     double dVerbrauchAbs; // Absoluter Verbrauch in l
24     double dVerbrauchMittel; // Verbrauch in l/100km
25
26     iStrecke = Eingabe.getInt("Geben Sie die gefahrene Strecke in km ein");
27     dVerbrauchAbs = Eingabe.getDouble("Geben Sie den Verbrauch in l ein");
28
29     dVerbrauchMittel = dVerbrauchAbs / iStrecke * 100;
30
31     System.out.printf("Verbrauch = %5.1f l/100km%n", dVerbrauchMittel);
32 }
33 }
```

- 2.2. Ein kleines Programm soll den Durchschnittsverbrauch eines Autos berechnen. Der Benutzer kann die gefahrene Strecke in km und den dabei verbrauchten Treibstoff in Liter eingeben. Das Programm gibt dann den Durchschnittsverbrauch Liter/(100 km) aus. Formatieren Sie die Ausgabe so, dass die Ausgabe auf eine Nachkommastelle gerundet wird.
- 2.3. Auf der 7. Etappe der Tour de France 2011 benötigte der Sieger 5:38:53 Std. Wie viele Stunden war er unterwegs? Das Programm soll die Dauer in Stunden als eine Dezimalzahl berechnen und am Bildschirm ausgeben.  
Diese Etappe war 218,5 km lang. Wie hoch war die Durchschnittsgeschwindigkeit des Siegers in km/h? Das Programm soll sie berechnen und auf 2 Nachkommastellen genau am Bildschirm ausgeben.  
Die Gesamtlänge der Tour betrug 3430 km mit insgesamt 21 Etappen.
- Wie lang war eine Etappe im Schnitt?
  - Wie viel Prozent der Gesamtstrecke war das?
  - Es wurden 22 Teams eingeladen, jedes Team besteht aus 9 Fahrern. 32 Fahrer schieden vorzeitig aus. Wieviel Prozent der Fahrer erreichten das Ziel in Paris?

```
20 * A0203TourDeFrance.java
5 package strProgAufg;
6
8 * @author stk
13 public class A0203TourDeFrance
14 {
15     /**
16      * @param args
17      * Kurzbeschreibung:
18      */
19     public static void main(String[] args)
20     {
21         double dLaengeAbs;
22         double dLaengeRel;
23         double dDauerStd;
24         double dVMittel;
25         double dFahrerErfolgRel;
26
27         dDauerStd = 5 + 38 / 60.0 + 53/3600.0;
28         System.out.printf("Zeitdauer für die Etappe= %8.5f h%n", dDauerStd);
29         dVMittel = 218.5 / dDauerStd;
30         System.out.printf("Die Durchschnittsgeschwindigkeit des Siegers betrug:"
31             + "%6.2f km/h%n", dVMittel);
32         dLaengeAbs = 3430 / 21.0;
33         System.out.printf("Mittlere Länge einer Etappe= %8.2f%n", dLaengeAbs);
34         dLaengeRel = dLaengeAbs / 3430 * 100;
35         System.out.printf("Eine Etappe entspricht im Mittel %6.2f %% der Gesamtlänge%n",
36             dLaengeRel);
37         dFahrerErfolgRel = (1 - 32/(9.0*22)) * 100;
38         System.out.printf("%5.2f %% der Fahrer kamen ans Ziel%n", dFahrerErfolgRel);
39     }
40 }
```

- 2.4. Schreiben Sie ein Programm, das vom Benutzer die Eingabe eines Großbuchstabens anfordert. Das Programm gibt auf dem Bildschirm aus, der wievielte Buchstabe im Alphabet der eingegebene ist.

**Hinweis:** In Java kann mit Char- Werten gerechnet werden, d.h. über die Differenz kann der Abstand zwischen zwei Buchstaben ermittelt werden.

**Erweiterung:** Das Programm soll in jedem Fall funktionieren, egal ob Groß- oder Kleinbuchstaben eingegeben werden. Hilfreich ist hierfür die Methode `Character.toUpperCase(char Zeichen)`, die als Parameter ein Zeichen erwartet und als Rückgabewert den entsprechenden Großbuchstaben liefert.

Bsp.: `char cGroß; cGroß = Character.toUpperCase('a') // cGroß ← 'A'`

```
20 * A0204Alphabet.java
5 package strProgAufg;
6
7 import input.Eingabe;
8
9 /**
10  * @author stk
11  * Kurzbeschreibung: Programm ermittelt die Position eines Buchstabens
12  *                   im Alphabet
13  */
14 public class A0204Alphabet
15 {
16     /**
17      * @param args
18      * Kurzbeschreibung:
19      */
20     public static void main(String[] args)
21     {
22         char cEingabe;
23         int iAbstand;
24         System.out.println("Geben sie einen Buchstaben ein:");
25         cEingabe = Eingabe.getChar();
26
27         iAbstand = Character.toUpperCase(cEingabe) - 'A';
28         System.out.printf("Der eingegebene Buchstabe '%c' ist der %d. im Alphabet.",
29                           cEingabe, iAbstand + 1);
30     }
31 }
```

2.5. Geben Sie die genaue Ausgabe auf dem Bildschirm an oder eine Begründung, warum keine Ausgabe erfolgen kann:

```
int    iZahl = 10    , iErgebnis = 0;
byte   bZahl = 80    , bErgebnis = 0;
long   lZahl = 34;
float  fZahl = 4.5f, fErgebnis = 0.0f;
double dZahl = 12.5;
short  sZahl = 30    , sErgebnis = 0;
```

		Ergebnis	
1	iErgebnis = iZahl / 4;	2	
2	bErgebnis = lZahl - 10;	Type Mismatch	bErgebnis = (byte)(lZahl - 10)
3	fErgebnis = lZahl / iZahl * fZahl;	13.5	
4	fErgebnis = fZahl * lZahl / iZahl;	15.3	
5	fErgebnis = dZahl * lZahl / iZahl;	Type Mismatch	fErgebnis = (float)dZahl * lZahl / iZahl; (42.5)
6	fErgebnis = dZahl * (float)iZahl;	Type Mismatch	fErgebnis = (float)dZahl * iZahl; 125.0
7	iErgebnis = (int)(fZahl/iZahl);	0	
8	sErgebnis = bZahl*dZahl;	Type Mismatch	sErgebnis = (short)(bZahl*dZahl); 1000
9	fErgebnis = (fZahl / (int)dZahl)+10.5;	Type Mismatch	fErgebnis = (fZahl/(int)dZahl)+10.5f; 10.875
10	bErgebnis = (byte)sZahl*80;	Type Mismatch	bErgebnis = (byte)(sZahl*80); 96 (aber falsches Ergebnis (eigentl. 2400), da nur die rechten 8 Bit benutzt werden)

11	<code>iErgebnis = (int)((10/3.0)*10);</code>	33	
12	<code>fErgebnis = fZahl * (lZahl / iZahl);</code>	13.5 (aber falsch!)	<code>fErgebnis = fZahl * (lZahl / (float)iZahl);</code> 15.3

- 2.6. Schreiben Sie ein Programm mit folgender Funktion: Der Benutzer kann eine Kommazahl eingeben, die eine Zeitspanne in Stunden darstellt. Das Programm gibt daraufhin diese Zeitspanne in der Darstellung hh:mm:ss,ms am Bildschirm aus.

```

 9 //**
10  * @author stk
11  * Kurzbeschreibung: Wandelt eine Zeitspanne von Stunden in
12  *                   hh:mm:ss,ms um
13  */
14 public class A0206Zeitspanne
15 {
16     * @param args[]
17     public static void main(String[] args)
18     {
19         double dZeitspanne;
20         double dRest;
21         int iStunden;
22         int iMinuten;
23         double dSekunden;
24
25         dZeitspanne = Eingabe.getDouble("Zeitspanne in h =");
26
27         iStunden = (int)dZeitspanne;
28         dRest = (dZeitspanne - iStunden) * 60;
29         iMinuten = (int) dRest;
30         dSekunden = (dRest - iMinuten) * 60;
31
32         System.out.printf("Die Zeitspanne entspricht %d:%d:%5.3f",
33                             iStunden, iMinuten, dSekunden);
34     }
35 }

```

Beispiel: Zeitspanne in h = 3.8284

Die Zeitspanne entspricht 3h 49' 42,240 s