

3 Verzweigung

- 3.1. Implementieren Sie das Struktogramm von Aufgabe 2.1 von Lsg_Struktogramm.pdf. Achten Sie auf die Wahl eines geeigneten Datentyps.

```
2 * A0301LagerBestand.java
5 package strProgAufg;
6
7 import input.Eingabe;
8
10 * @author stk
15 public class A0301LagerBestand
16 {
17     /**
18      * @param args
19      * Kurzbeschreibung: Ermittelt wird die Bestellmenge in Abhängigkeit
20      * vom Lagerbestand und vom Mindestbestand
21      */
22     public static void main(String[] args)
23     {
24         int iBestand; // aktueller Lagerbestand
25         int iMinBestand; // Mindestbestand
26         int iBestellmenge;
27
28         iBestand = Eingabe.getInt("Aktueller Lagerbestand = ");
29         iMinBestand = Eingabe.getInt("Mindestbestand = ");
30
31         if (iBestand < iMinBestand)
32         {
33             iBestellmenge = 2 * iMinBestand - iBestand;
34         }
35         else
36         {
37             iBestellmenge = 0;
38         }
39         System.out.println("Bestellmenge = " + iBestellmenge);
40     }
41 }
```

- 3.2. Implementieren Sie das Struktogramm von Aufgabe 2.2 von Lsg_Struktogramm.pdf. Formatieren Sie die Ausgabe so, dass die Ausgabe auf zwei Nachkommastelle gerundet wird.

```

13 public class A0302berechneBestellwert
14 {
15     /**
16      * @param args
17      * Kurzbeschreibung:
18      */
19     public static void main(String[] args)
20     {
21         double dWert; // Bestellwert
22         double dRabatt;
23
24         dWert = Eingabe.getDouble("Geben Sie den Bestellwert ein ");
25
26         if (dWert >= 500)
27         {
28             dRabatt = dWert * 0.1; // 10 % Rabatt
29         }
30         else
31         {
32             if (dWert >= 200)
33             {
34                 dRabatt = dWert * 0.05; // 5 % Rabatt
35             }
36             else
37             {
38                 dRabatt = 0;
39             }
40         }
41
42         dWert = dWert - dRabatt;
43
44         System.out.printf("Rabatt =           %8.2f €\n", dRabatt);
45         System.out.printf("Rechnungsbetrag = %8.2f €\n", dWert);
46     }
47 }

```

3.3. Um die Auswertung von Wettkampfergebnissen für ein Sportabzeichen im Weitsprung besser vergleichen zu können, sollen die Sprungergebnisse mit Hilfe eines Computerprogramms ausgewertet werden:

	Gold	Silber	Bronze
männlich	über 5,70 Meter	über 4,70 bis 5,70 Meter	über 3,50 bis 4,70 Meter
weiblich	über 5,10 Meter	über 4,20 bis 5,10 Meter	über 3,10 bis 4,20 Meter

- Der Benutzer soll zunächst eingeben, ob er männlich (m) oder weiblich (w) ist. Gibt der Benutzer hier etwas Falsches ein (d.h. etwas anderes als 'm', 'M', 'w' oder 'W'), dann erhält er eine Fehlermeldung.
- Dann gibt der Benutzer seine Sprungweite ein und das Programm zeigt an, welche Auszeichnung der Sportler erhält.

```
6 import input.Eingabe;
7 /**
8  * @author stk
9  *
10  *
11  * Kurzbeschreibung: Abhängig vom Weitsprungergebnis wird eine
12  *                   Medaille vergeben.
13  */
14 public class A0303Weitsprung
15 {
16     * @param args[]
17     public static void main(String[] args)
18     {
19         char cGeschlecht;
20         double dSprungWeite;
21         double dGold, dSilber, dBronze;
22
23         cGeschlecht =
24             Eingabe.getChar("Geben Sie w für weiblich oder m für männlich ein:");
25         // Damit auch die Eingabe von W oder M funktioniert wird die
26         // Eingabe in Kleinbuchstaben umgewandelt
27         cGeschlecht = Character.toLowerCase(cGeschlecht);
28         if (cGeschlecht == 'w' || cGeschlecht == 'm')
29         {
30             dSprungWeite =
31                 Eingabe.getDouble("Sprungweite? ");
32             if (cGeschlecht == 'm')
33             {
34                 dGold = 5.7;
35                 dSilber = 4.7;
36                 dBronze = 3.5;
37             }
38             else
39             {
40                 dGold = 5.1;
41                 dSilber = 4.2;
42                 dBronze = 3.1;
43             }
44             if (dSprungWeite > dGold)
45             {
46                 System.out.println("Gold Medaille");
47             }
48             else
49             {
50                 if (dSprungWeite > dSilber)
51                 {
52                     System.out.println("Silber Medaille");
53                 }
54                 else
55                 {
56                     if (dSprungWeite > dBronze)
57                     {
58                         System.out.println("Bronze Medaille");
59                     }
60                 }
61             }
62         }
63         else
64         {
65             System.out.println("Nur m oder w eingeben!");
66         }
67     }
68 }
```

3.4. Es soll ein Programm erstellt werden, bei dem der Benutzer aufgefordert wird, ein Zeichen einzugeben. Anschließend soll ausgegeben werden, ob das Zeichen

- eine Dezimalziffer
- ein Buchstabe oder
- ein anderes Zeichen ist.

Der Quelltext kann kürzer werden, wenn man die vorhandene Methode

`Character.toLowerCase(cZeichen)` verwendet. Sie bekommt als Parameter einen `char`-Wert und gibt einen `char`-Wert zurück. Ist `cZeichen` ein Großbuchstabe, so wird dieser in einen Kleinbuchstaben umgewandelt und zurückgegeben. Ansonsten wird einfach `cZeichen` wieder zurückgegeben.

```
20 * A0304ZeichenCheck.java
5 package strProgAufg;
6 import input.Eingabe;
7 /**
8  * @author stk
9  *
10 * Kurzbeschreibung: Ein Zeichen wird eingelesen und geprüft, ob
11 *                   ein Buchstabe, eine Ziffer oder sonstiges Zeichen
12 *                   ist
13 */
14 public class A0304ZeichenCheck
15 {
16     * @param args
17     public static void main(String[] args)
18     {
19         char cZeichen;
20
21         cZeichen = Eingabe.getChar("Geben Sie ein Zeichen ein: ");
22
23         if (cZeichen >= '0' && cZeichen <= '9')
24         {
25             System.out.println("Das ist eine Ziffer");
26         }
27         else
28         {
29             cZeichen = Character.toLowerCase(cZeichen);
30             if (cZeichen >= 'A' && cZeichen <= 'Z'
31                 || cZeichen == 'ä' || cZeichen == 'ö' || cZeichen == 'ü'
32                 || cZeichen == 'ß')
33             // if (Character.isLetter(cZeichen))
34             {
35                 System.out.println("Das ist ein Buchstabe");
36             }
37             else
38             {
39                 System.out.println("Das ist ein sonstiges Zeichen");
40             }
41         }
42     }
43 }
44 }
45 }
46 }
```

3.5. Implementieren Sie das Struktogramm von Aufgabe 2.8 von Lsg_Struktogramm.pdf.

```
public class A0305_EinfacherRechner
{
    * @param args[]
    public static void main(String[] args)
    {
        boolean bFehler = false; // false bedeutet, kein Fehler aufgetreten
        double dOp1;
        double dOp2;
        double dErgebnis = 0;
        char cOperator;

        dOp1 = Eingabe.getDouble("Operand1 = ");
        cOperator = Eingabe.getChar("Operator = ");
        dOp2 = Eingabe.getDouble("Operand2 = ");

        switch (cOperator)
        {
            case '+':
                dErgebnis = dOp1 + dOp2;
                break;
            case '-':
                dErgebnis = dOp1 - dOp2;
                break;
            case '*':
                dErgebnis = dOp1 * dOp2;
                break;
            case '/':
                if (dOp2 != 0)
                {
                    dErgebnis = dOp1 / dOp2;
                }
                else
                {
                    System.out.println("Division durch 0 nicht erlaubt!");
                    bFehler = true;
                }
                break;
            default:
                System.out.println("Nur +|-|*|/ erlaubt!");
                bFehler = true;
                break;
        }
        if (!bFehler)
        {
            System.out.printf("Ergebnis = %17.15f", dErgebnis);
        }
    }
}
```

3.6. Erstellen Sie ein Programm, das eine quadratische Gleichung der Form:

$$ax^2 + bx + c = 0 \text{ löst.}$$

Die Koeffizienten a , b , c werden vom Benutzer eingegeben. Das Programm berechnet die Diskriminante und zeigt diese an. Dann wird x_1 und x_2 bzw. eine Lösung x berechnet

und ausgegeben oder eine der folgenden Meldungen angezeigt:

- „a darf nicht 0 sein!“
- „Die Gleichung hat keine reelle Lösung!“

Achtung: Entwickeln Sie die Lösung zunächst in einem Struktogramm und führen einen Schreibtischtest durch.

(Testwerte: $a = 2$, $b = -8$, $c = 8$; \rightarrow 1 Lösung: $x = 2$

$a = 2$, $b = 4$, $c = 1,5$; \rightarrow 2 Lösungen: $x_1 = -1,5$; $x_2 = -0,5$

$a = 3$, $b = 2$, $c = 4$; \rightarrow keine reelle Lösung!

```
import input.Eingabe;

/* @author stk */
public class A0306QuadrGleichung
{
    /* @param args */
    public static void main(String[] args)
    {
        double a, b, c, d, wu, x1, x2;

        System.out.printf("Start Programm \"Quadratische Gleichung\"\n\n");
        System.out.printf("Bitte die Koeffizienten a, b und c eingeben\n");
        a = Eingabe.getDouble("a = ");

        /* Ist a Null? */
        if (a == 0)
        {
            System.out.printf("\nDie Zahl a muss != 0 sein");
        }
        else
        {
            b = Eingabe.getDouble("b = ");
            c = Eingabe.getDouble("c = ");

            d = b*b-4*a*c; /* Diskriminante berechnen */
            System.out.printf("\nDer Wert der Diskriminante d = %6.3f\n", d);

            if (d < 0) /* Keine reelle Lösung? */
            {
                System.out.printf("\nKeine reelle Loesung.\n");
            }
            else
            {
                if (d == 0) /* Nur eine Lösung? */
                {
                    x1 = -b/(2*a);
                    System.out.printf("x1=x2= %6.3f", x1);
                }
                else /* Zwei Lösungen */
                {
                    wu = Math.sqrt(d);
                    x1 = (-b + wu)/(2*a);
                    x2 = (-b - wu)/(2*a);
                    System.out.printf("x1= %6.3f\n", x1);
                    System.out.printf("x2= %6.3f\n", x2);
                }
            }
        }
        System.out.printf("\nEnde des Programms.\n\n");
    }
}
```