4 Schleifen

Wählen Sie jeweils den am besten geeigneten Schleifentyp

4.1. Implementieren Sie die Struktogramme von Aufgabe 4.1 a) und b) von Lsg_Struktogramm.pdf.

4.2. a) Ein Programm soll solange Temperaturmesswerte einlesen, bis der Wert 9999 als Endekennung eingegeben wird. Aus den Messwerten soll der Mittelwert berechnet werden, wobei generell nur die Messwerte berücksichtig werden sollen, die im Bereich zwischen - 40 und + 50 liegen. Alle anderen Werte werden als Fehlmessungen betrachtet.

b) Wie a) allerding soll jetzt aus den gültigen Messwerten der Maximalwert ermittelt werden.
Hinweis: Überlegen zu zunächst, wie Sie (ohne Computer) vorgehen würden, wenn Sie in einer viele Seiten langen Zahlenspalte den höchsten Wert ermitteln sollen.

```java
/**
 * @author stk
 * Kurzbeschreibung: Lese Messwerte solange bis Wert 9999 als Endekennung
 *                   eingegeben wire. Berechne den Mittelwert und den
 *                   Maximalwert
 */
public class A0402Messreihe
{
    /**
     * @param args
     */
    public static void main(String[] args)
    {
        double dMessw = 0;
        double dMittelw = 0;
        double dMax;
        int iAnzahl = 0;

        dMax = -40;
        System.out.println("Geben Sie die Messwerte ein. Beenden mit \"9999\"\n");
        do
        {
            dMessw = Eingabe.getDouble("Messwert " + (iAnzahl+1) + " =");
            if (dMessw >= -40 && dMessw <= 50)
            {
                iAnzahl++;
                dMittelw = dMittelw + dMessw;
                if (dMessw > dMax)
                {
                    dMax = dMessw;
                }
            }
        } while (dMessw != 9999);

        if (iAnzahl > 0)
        {
            dMittelw = dMittelw / iAnzahl;
            System.out.printf("Der Mittelwert = %6.2f\n", dMittelw);
            System.out.printf("Der Maxwert = %6.2f\n", dMax);
        }
    }
}
```

4.3. Die Eulersche[1] Zahl e = 2,7182818284523… spielt in der Mathematik und vielen anderen Wissenschaften eine große Rolle. Diese Zahl kann mit Hilfe einer Reihensumme berechnet werden: **`e = 1 + 1/1 + 1/(1*2) + 1/(1*2*3) + ... + 1/(n!)`**

Möchte man in einem Programm die Zahl näherungsweise berechnen, sollte man sich vorher genau überlegen, wie die einzelnen Reihenglieder berechnet werden können. Am besten man schaut sich zunächst die ersten paar Schritte an und analysiert, welche Größen sich von Schritt zu Schritt verändern und wie Schritt 2 mit Schritt 1 und Schritt 3 mit Schritt 2 zusammenhängt. Das Ergebnis dieser Überlegung ist dann die allgemeine Angabe von Schritt i. Eine Formel, bei der ein Schritt auf den vorherigen in dieser Weise aufbaut, nennt man Rekursionsformel.

```
Startwerte:     Summe <- 1,0
                Summand <- 1,0/1

Schritt i=1:    Summe <- 1,0 + 1,0          (= 2,0)
                Summand <- 1,0/2            (= 0,5)

Schritt i=2:    Summe <- 2,0 + 0,5          (= 2,5)
                Summand <- 0,5/3            (= 0,1666...)

Schritt i=3:    Summe <- 2,5 + 0,1666       (= 2,666...)
                Summand <- 0,1666... /4     (= 0,041666...)


Allgemein:
Schritt i:    Summe <- Summe + Summand
          Summand <- Summand/(i+1)
```

Erstellen Sie ein Programm, das die Eulersche Zahl berechnet, wobei der Benutzer die Zahl der Schritte vorgibt.

[1] Sie wurde nach dem Schweizer Mathematiker Leonhard Euler benannt, der zahlreiche Eigenschaften von e beschrieb.

```java
/* A0403EulerZahl.java
package strProgAufg;

import input.Eingabe;

/**
 * @author stk
 *
 * Kurzbeschreibung: Die Eulersche Zahl wird mit einer Reihensumme berechnet
 *                   Die Anzahl Schritt gibt der Benutzer vor!
 */
public class A0403EulerZahl
{
    /**
     * @param args
     */
    public static void main(String[] args)
    {
        int i;
        int iSchritte;
        double dSumme = 1;
        double dSummand = 1;

        iSchritte = Eingabe.getInt("Wie viele Schritte sollen berechnet werden? ");

        for (i=1; i <= iSchritte; i++)
        {
            dSumme = dSumme + dSummand;
            dSummand = dSummand / (i+1);
        }

        System.out.printf("Nach %d Rechenschritten ist e = %17.15f", i - 1, dSumme);
    }
}
```

4.4. Erstellen Sie ein Programm, das die Kreiszahl π nach folgender Summenformel berechnet: **π = 4 * (1 – 1/3 + 1/5 – 1/7 + ...**

Gehen Sie für den Algorithmus-Entwurf wie in Aufgabe 4.3 vor.

Erstellen Sie das Programm, wobei die Anzahl der Schritte nicht fest vorgegeben ist. Die Schleife soll solange laufen, bis sich die Summe nur noch in der 10^{-9} Stelle verändert. Wie viele Schleifendurchläufe sind notwendig?

```
Startwerte:     Summe <- 0
                Nenner <- 1
                Vz <- +1

Schritt i=1:    Summe <- 0 + Vz*1/1      (= 1,0)
                Nenner <- 1 + 2    (= 3)
                Vz <- (-1)* Vz     (= -1)

Schritt i=2:    Summe <- 1,0 + Vz*1/3    (= 0,6666...)
                Nenner <- 3 + 2    (= 5)
                Vz <- (-1)* Vz     (= +1)

Schritt i=3:    Summe <- 0,1666... + Vz*1/5    (= 0,8666)
                Nenner <- 5 + 2    (= 7)
                Vz <- (-1)* Vz     (= -1)
```

```java
 * A0404Kreiszahl.java
package strProgAufg;

/**
 * @author stk
 * Kurzbeschreibung: Programm berechnet die Kreiszahl Pi
 */
public class A0404Kreiszahl
{
     * @param args
    public static void main(String[] args)
    {
        int i = 0;
        int iVz = 1;
        double dSumme = 1;
        int iNenner = 1;

        do
        {
            i++;
            iVz = iVz * (-1);
            iNenner = iNenner +  2;
            dSumme = dSumme + iVz * (1.0 / iNenner);
//          System.out.printf("Nach %d Rechenschritten ist Pi = %17.15f\n", i, 4*dSumme);
        } while (iNenner <= 1e9);
        System.out.printf("Nach %d Rechenschritten ist Pi = %17.15f\n", i, 4*dSumme);

    }
}
```

4.5. Implementieren Sie das Struktogramme von Aufgabe 4.14 von Lsg_Struktogramm.pdf. Benutzen Sie die formatierte Ausgabe, so dass eine Tabelle angezeigt wird:

```
Bitte geben Sie den Einzahlungsbetrag in Euro ein:
1000

          |    1.Jahr  |    2.Jahr  |    3.Jahr  |    4.Jahr  |    5.Jahr
----------+-----------------------------------------------------------------
2% Zinsen |    1020,00 |    1040,40 |    1061,21 |    1082,43 |    1104,08
3% Zinsen |    1030,00 |    1060,90 |    1092,73 |    1125,51 |    1159,27
4% Zinsen |    1040,00 |    1081,60 |    1124,86 |    1169,86 |    1216,65
5% Zinsen |    1050,00 |    1102,50 |    1157,63 |    1215,51 |    1276,28
6% Zinsen |    1060,00 |    1123,60 |    1191,02 |    1262,48 |    1338,23
```

```java
/**
 * A0405ZinsesZinsTab.java
 */
package strProgAufg;

import input.Eingabe;

/**
 * @author stk
 */
public class A0405ZinsesZinsTab
{
    /**
     * @param args
     */
    public static void main(String[] args)
    {
        // Variablendeklaration:
        double betrag, guthaben;
        int zinsen, jahr;

        // Eingabe
        System.out.println("* * * PROGRAMM ZUR ZINSESZINSBERECHNUNG * * * \n");
        betrag = Eingabe.getDouble("Bitte geben Sie den Einzahlungsbetrag in Euro ein: ");

        // Ausgabe des Tabellenkopfes:
        System.out.printf("\n          ");
        for (jahr = 1; jahr <= 5; jahr++)
        {
            System.out.printf("|    %d.Jahr  ", jahr);
        }
        System.out.println();

        // Ausgabe der gestrichelten Linie:
        System.out.println("----------+----------------------------------------------------------");

        for (zinsen = 2; zinsen <= 6; zinsen++)
        {
            guthaben = betrag;
            System.out.printf("%d%% Zinsen ", zinsen);

            for (jahr = 1; jahr <= 5; jahr++)
            {
                guthaben = guthaben + guthaben * (zinsen / 100.0);
                System.out.printf("|%11.2f ", guthaben);
            }
            System.out.println();
        }
    }
}
```

4.6. Erstellen Sie ein Programm, das eine Multiplikationstabelle wie nebenstehend am Bildschirm anzeigt

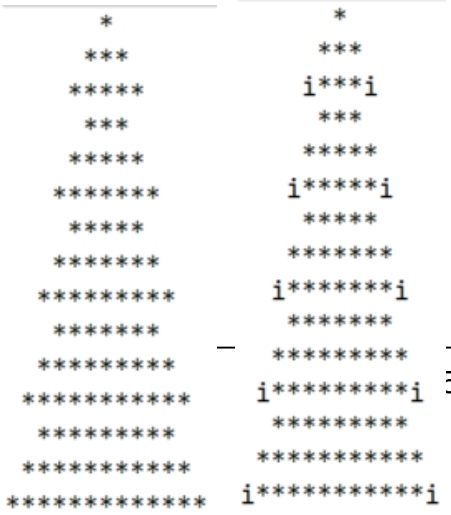
	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

gds2 Gottlieb-Daimler-Schule 2
Technisches Schulzentrum Sindelfingen
mit Abteilung Akademie für Datenverarbeitung

```java
/*
 * A0406MultiplikationsTab.java
 */
package strProgAufg;

/**
 * @author stk
 * Kurzbeschreibung: Ausgabe einer Multiplikationstabelle
 */
public class A0406MultiplikationsTab
{

    /**
     * @param args
     */
    public static void main(String[] args)
    {
        //Variablendeklaration:
        int zeile, spalte;

        //Ausgabe der ersten Zeile der Tabelle:
        System.out.printf("     |");
        for (spalte = 1; spalte <= 10; spalte++)
        {
            System.out.printf("%4d",spalte);
        }
        System.out.println();    //Zeilenumbruch

        //Ausgabe der gestrichelten Linie:
        System.out.printf("-----+");
        for (spalte = 1; spalte <= 4*10; spalte++)
        {
            System.out.printf("-");
        }
        System.out.println();    //Zeilenumbruch

        //Ausgabe der Tabelle mit den Zeilen 1 bis 10
        for (zeile = 1; zeile <= 10; zeile++)
        {
            System.out.printf("%4d |", zeile);
            for (spalte = 1; spalte <= 10; spalte++ )
            {
                System.out.printf("%4d",spalte*zeile);

            }
            System.out.println();    //Zeilenumbruch
        }
    }
}
```

4.7. Weihnachten ist nicht mehr soweit entfernt.
a) Schreiben Sie ein Programm, das nach Eingabe der Anzahl der Äste (Trapeze) und der Anzahl der Zeilen pro Ast einen Weihnachtsbaum ausgibt (linkes Bild).

b) Der Baum soll nun noch mit Kerzen geschmückt



werden (rechtes Bild).

```java
 * @author stk
public class A0407bBaum
{
     * @param args
    public static void main(String[] args)
    {
        int anzAeste = 5;
        int anzZeilen = 3;
        int iAeste;
        int iZeile;
        int iLeer;
        int iStern;

        for (iAeste = 1; iAeste <= anzAeste; iAeste++)
        {
            // Ein Aste besteht aus mehreren Zeilen
            // Die Zeilenzahl ist konstant, aber der Zahlenbereich verschiebt sich
            // Bei ersten Ast: 1..anzAeste, beim zweiten: 2..anzAeste+1, usw.
            for (iZeile = iAeste; iZeile <= anzZeilen + iAeste - 1; iZeile++)
            {
                // Die Leerzeichen hängen ab von anzZeilen und anzAeste und der Zeile
                //  innerhalb eines Astes (iZeile)
                for (iLeer = 1; iLeer <= anzZeilen - iZeile + anzAeste - 1; iLeer++)
                    System.out.printf(" ");
                // Die Sterne hängen von iZeile ab und dadurch indirekt von iAeste
                if (iZeile == anzZeilen + iAeste - 1) // letzte Zeile eines Astes
                {
                    System.out.printf("i");  // statt erstem Stern eine Kerze
                    // 2 Sterne weniger wegen der Kerzen
                    for (iStern = 1; iStern <= 2 * iZeile - 1 - 2; iStern++)
                        System.out.printf("*");
                    System.out.printf("i");  // statt letztem Stern eine Kerze
                    System.out.printf("\n");
                }
                else
                {
                    for (iStern = 1; iStern <= 2 * iZeile - 1; iStern++)
                        System.out.printf("*");
                    System.out.printf("\n");

                }
            }
        }
    }
}
```