1. Aufgabe

Die folgende Java-Prozedur ist syntaktisch korrekt und erfüllt einen Zweck, die Kontrollstrukturen wurden jedoch unzweckmäßig verwendet. Betten Sie diese Prozedur in ein startbares Programm ein, das diese Prozedur aufruft. Verbessern Sie das Programm durch Verwendung angemessenerer Kontrollstrukturen.

```
public static void durchlaufeKontrollstrukturen(boolean Bedingung, int Wert)
{
    int i;
    i = 1;
    while(i < 10)
    {
        System.out.println(i);
        i = i + 1;
    }

    while(Bedingung == true)
    {
        System.out.println("Bedingung ist wahr");
        break; // Beendet while-Schleife
    }

    if(Wert == 0)
    {
        System.out.println("Wert ist 0");
    }
    else if(Wert == 1)
    {
        System.out.println("Wert ist 1");
    }
    else if(Wert == 2)
    {
        System.out.println("Wert ist 2");
    }
    else
    {
        System.out.println("Wert ist weder 0, noch 1, noch 2");
    }
}
```

2. Aufgabe

In einer Sparkasse wird eine Tabelle benötigt, die angibt, auf welchen Betrag ein Anfangskapital nach einem Jahr bei Verzinsung nach verschiedenen Zinssätzen anwächst. Schreiben Sie einen Algorithmus in Java, der für ein Anfangskapital von 100,- € und für die Zinssätze von 3 Prozent bis 10 Prozent (in Schritten von 0,5 Prozent steigend) das Endkapital nach folgender Formel berechnet und in Tabellenform ausgibt:

Endkapital = Anfangskapital * (1 + Zinssatz in % / 100)

Erweitern Sie anschließend den Algorithmus so, dass hintereinander Tabellen für ein Startkapital von 100,- €; 250,- € und 750,- € berechnet und ausgegeben werden.

Hinweis: Stellen Sie bei der Ausgabe Anfangskapital, Zinssatz und Endkapital jeweils übersichtlich dar.

Prinzip der Ausgabedarstellung:

Einfach

```
anfangskapital  endkapital  zinssatz
100,00          103,00      3,00
100,00          103,50      3,50

  . . .           . . .       . . .

100,00          110,00      10,00
E N D E Berechnung
```

erweitert

```
Berechnung mit Anfangskapital: 100.0
anfangskapital  endkapital  zinssatz
100,00          103,00      3,00

  . . .           . . .       . . .

100,00          110,00      10,00
E N D E Berechnung

Berechnung mit Anfangskapital: 250.0
anfangskapital  endkapital  zinssatz
250,00          257,50      3,00

  . . .           . . .       . . .

250,00          275,00      10,00
E N D E Berechnung

Berechnung mit Anfangskapital: 750.0
anfangskapital  endkapital  zinssatz
750,00          772,50      3,00

  . . .           . . .       . . .

750,00          825,00      10,00
E N D E Berechnung
```

Denken Sie bei der erweiterten Variante über den geschickten Einsatz eines Feldes und geschachtelter Schleifen und/oder Prozeduraufrufen nach.

3. Aufgabe

Schreiben Sie ein Windows- Konsole- Programm „Logik“, das folgendes leistet:

- Es definiert ein Feld a vom Typ **int** sowie eine Hilfsvariable w vom Typ **boolean**. Weitere Hilfsvariable dürfen nach Bedarf definiert werden. Die Länge von a wird eingelesen. Getestet wird mit der Feldlänge 4 (siehe Testbeispiele). Es sollen aber beliebige Feldlängen möglich sein.
- Es lässt die Werte der Komponenten von a über die Tastatur eingeben.
- Es berechnet der Reihe nach die Wahrheitswerte der folgenden Aussagen und legt diese in der Hilfsvariablen w ab. Nach jeder Berechnung wird w mit geeigneter Beschriftung am Bildschirm ausgegeben
  1. Aussage: Mindestens ein Feldelement von a hat einen Wert größer als 1.
  2. Aussage: Mindestens ein, jedoch höchstens drei Feldelemente von a haben einen Wert größer als 1.
  3. Aussage: Genau ein Feldelement von a ist negativ.
  4. Aussage: Alle Feldelemente, die größer als 0 sind, sind auch größer als 10.
  5. Aussage: Für jedes Feldelement von a gilt: Es ist entweder größer als 10 oder es ist kleiner als -5.

Testbeispiele:

Eingaben	1. Aussage	2. Aussage	3. Aussage	4. Aussage	5. Aussage
-1, -2, 0, 15	true	true	false	true	false
3, 4, -1, 8	true	true	true	false	false
-6, 15, -8, 11	true	true	false	true	true
2, 15, 18, 20	true	false	false	false	false
-5, -4, -3, -2	false	false	false	true	false
20, 30, 20, 40	true	false	false	true	true

Ein-/Ausgabebeispiel:

```
Feldlänge> 4
a[0]> -1
a[1]> -2
a[2]> 0
a[3]> 15
1. Aussage (Mindestens ein Feldelement hat einen Wert größer als 1): true
2. Aussage (Mindestens ein, höchstens drei Feldelemente haben einen Wert
größer als 1): true
3. Aussage (Genau ein Feldelement ist negativ): false
4. Aussage (Alle Feldelemente, die größer als 0 sind, sind auch größer als
10): true
5. Aussage (jedes Feldelement ist entweder größer als 10 oder es ist kleiner
als -5): false
```

4. Aufgabe

a) Erstellen Sie Übung3, Aufgabe 9, Ausdruck 4 nochmals. Ersetzen Sie dabei die Variablen a0 bis a6 durch ein entsprechendes Feld a. Für die Berechnung des Ausdrucks benötigen Sie dann natürlich eine **for**-Schleife. Machen Sie eine erklärende Ausgabe dazu, so dass man die Inhalte von A und die Rechnung sieht.

b) Wenn Ihr Programm funktioniert erstellen Sie die folgende weitere Variante dazu. Die Feldlänge soll eingegeben werden und das Feld dann mit Zufallszahlen gefüllt werden. Berechnen Sie dann den Ausdruck. Die Zufallszahl pro Feldelement wird folgendermaßen bestimmt: `a[i] = (int)(Math.random()*100)/10d;`

`Math.random()` erzeugt eine `double` Zufallszahl `z` für die gilt: `0 <= z < 1`. Durch obigen Ausdruck wird eine Zahl mit jeweils einer Stelle vor und nach dem Komma erzeugt (Überlegen Sie warum?).

Ein-/Ausgabebeispiel zu a):

```
x> 1.5
1,0*1,50**0 - 2,5*1,50**1 + 0,1*1,50**2 - 0,0*1,50**3 + 0,8*1,50**4 -
0,0*1,50**5 + 1,5*1,50**6 = 18,610938
```

Ein-/Ausgabebeispiel zu b):

```
Feldlänge> 7
x> 1.5

3,7*1,50**0 - 3,1*1,50**1 + 0,8*1,50**2 - 9,1*1,50**3 + 8,5*1,50**4 -
9,7*1,50**5 + 3,1*1,50**6 = -25.179687499999993
```

5. Aufgabe

Führen Sie folgenden Schleifenumformungen durch:

a. Formulieren Sie eine Java-Zählschleife (`for`), die eine gegebene Zahl n mit der Schrittweite -2 auf 3 runter zählt.
b. Formen Sie diese Zählschleife in eine `while`-Schleife um.
c. Formen Sie diese `while`-Schleife in eine `do`-Schleife um.

Erstellen Sie ein Programm welches eine Zahl n einliest und dieses in der jeweiligen Schleifenform runterzählt. Die Schleifen sollen hintereinander in einem Programm durchlaufen werden. Jeder Schleifendurchlauf soll via Ausgabe dokumentiert werden. Experimentieren Sie auch mit n=3 oder kleiner.

Ein-/Ausgabebeispiel:

```
n> 9
for --> n: 9, Zählerstand: 9
for --> n: 9, Zählerstand: 7
for --> n: 9, Zählerstand: 5

while --> n: 9, Zählerstand: 9
while --> n: 9, Zählerstand: 7
while --> n: 9, Zählerstand: 5

do --> n: 9, Zählerstand: 9
do --> n: 9, Zählerstand: 7
do --> n: 9, Zählerstand: 5
```