

b) Die do-while-Schleife:

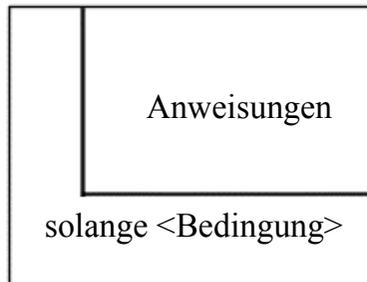
Bei der do-while-Schleife wird erst am Schleifenende überprüft, ob eine Bedingung erfüllt ist:

```
Führe aus
  Anweisung 1
  Anweisung 2
  ...
  Anweisung n
solange Bedingung
} Schleifenrumpf
```

Dies steht im Gegensatz zur while-Schleife, denn dort bei ihr wird schon zu Beginn der Schleife eine Bedingung geprüft. In beiden Fällen gilt, dass solange die Bedingung erfüllt, also wahr (true), ist, die Schleife durchlaufen wird. Wenn die Bedingung irgendwann nicht mehr erfüllt, also falsch (false) ist, wird die Schleife verlassen.

Auch bei der do-while-Schleife muss eine entsprechende Anweisung im Schleifenrumpf sicherstellen, dass die Abbruchbedingung nach endlich vielen Schleifendurchläufen den Wert false annimmt. Eine do-while-Schleife wird mindestens einmal durchlaufen, da die Schleifenbedingung erst nach jedem Schleifendurchlauf geprüft wird. Man nennt die do-while-Schleife daher eine **annehmende Schleife**.

Struktogramm:



Formulierung in Java:

```
do
  Anweisung;
while (Bedingung)
bzw.
do{
  Anweisung1;
  Anweisung2;
  ...
}
while (Bedingung)
```

So wie in Java üblich gilt, dass wenn die Anweisung aus mehreren Einzelanweisungen zusammengesetzt ist, müssen diese durch { und } zusammengefasst werden.

Beispiele:

```
public class DoWhileDemo{

  public static void main(String args[]){
    // Initialisierung
    int x=20;

    // Schleife
    do{
      // Ausgabe des aktuellen Wertes von x
      System.out.println(x);
      x=x/2;
    }
  }
}
```

Do-While-Schleifen

© Wittye 2004

```
}
while (x>1);
}
```

Das Programm DoWhileDemo gibt die Zahlen 20,10,5,2 in jeweils einer neuen Zeile aus und entspricht damit dem Beispiel WhileDemo auf dem Blatt zur while-Schleife.

Beispiel: Summieren von Zahlen

Das folgende Programm summiert die Zahlen 1 bis 50:

```
public class SummeBsp{

  public static void main(String args[]){
    int x, summe;

    // Initialisierung
    x=1;
    summe=0;

    // Aufsummieren durch do-while-Schleife
    do{
      summe=summe+x;
      x++;
    }
    while (x<=50);
    // Ausgabe des Ergebnisses
    System.out.println(summe);
  }
}
```

Aufgaben:

1. Löse die Aufgaben 1 vom letzten Übungsblatt unter Verwendung der do-while-Schleife.
2. Löse die Aufgaben 2 vom letzten Übungsblatt unter Verwendung der do-while-Schleife.
3. Löse die Aufgaben 4 vom letzten Übungsblatt unter Verwendung der do-while-Schleife.
4. Schreibe ein Java-Programm, das eine Wertetabelle für den Kosinus ausgibt.
5. Eine Tasse Kaffee, die gerade auf den Tisch gestellt worden ist, hat eine Temperatur von 85 °C. Die Zimmertemperatur beträgt 21°C. In jeder Minute verringert sich die Temperatur des Kaffees um ein Zehntel der Differenz von Kaffeetemperatur und Zimmertemperatur. Schreibe ein Java-Programm, das die Kaffeetemperatur nach 0,1,2,3,4,... Minuten ausgibt.
6. Zur näherungsweisen Berechnung aus einer Quadratwurzel aus einer Zahl a gibt es folgendes Verfahren, das nach Heron von Alexandria (60 n. Chr.) benannt ist:
 - Gib dir eine Zahl a vor (Einlesen von der Tastatur!). Halbiere die Zahl a und nenne das Ergebnis x.
 - Dividiere die Zahl a durch die Zahl x und nenne das Ergebnis z
 - Wiederhole, solange bis $(z-x)(z-x) < 1/100000$:
 - Addiere zu der Zahl x die Zahl z und nenne die Hälfte der Summe x.
 - Dividiere die Zahl a durch die Zahl x und nenne das Ergebnis z.
 - Gebe die Zahl z aus.

Schreibe ein Java-Programm, das die Quadratwurzel nach diesem Verfahren berechnet und zur Kontrolle die Quadratwurzel mit der entsprechenden Java-Funktion Math.sqrt ausgibt.

Do-While-Schleifen

© Wittye 2004