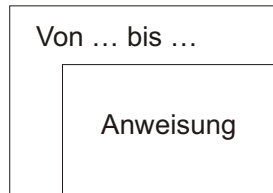


c) Die for-Schleife:

Wenn bereits bei der Erstellung eines Programms die Anzahl der notwendigen Schleifendurchläufe bekannt ist, kann (!) man eine FOR-Schleife benutzen. Deshalb wird die for-Schleife auch als Zählschleife bezeichnet. Sie ist eine **abweisende** Schleife (vgl. while-Schleife).

Das Struktogramm für eine Zählschleife ist:



Leider ist der Aufbau in Java wesentlich komplizierter und kann z.T. durch das Struktogramm nicht vollständig erfasst werden.

Formulierung in Java:

```
for(Initialisierung; Ausdruck; Zählanweisung)
    Anweisung;
```

Ist die Anweisung aus mehreren Einzelanweisungen zusammengesetzt, so müssen diese durch { und } zusammengefasst werden, wie bei den anderen Schleifen oder Verzweigung.

Betrachten wir nun die for-Schleife etwas genauer:

- Im Teil *Initialisierung* wird die Zählvariable initialisiert, d.h. auf den Startwert gesetzt. Dieser Teil wird nur einmal durchlaufen. Außerdem wird häufig sogar die Zählvariable deklariert. Dadurch ist sie nur innerhalb der Schleife verfügbar und kann außerhalb nicht verwendet werden.
- Der Teil *Ausdruck* steht ein logischer Ausdruck, der entweder *true* oder *false* ist. Solange er *true* ist, wird die Schleife durchlaufen.
- Im Teil *Zählanweisung* wird der Wert der Zählvariable geändert.

Beispiel:

Da diese allgemeine Erläuterungen recht kompliziert erscheinen, schauen wir uns am besten einige Code-Beispiele an:

```
public class ForDemo{

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

        // Schleife
        for(x=20;x>1;x=x/2){
            System.out.println(x);
        }
    }
}
```

Das erste Beispiel liefert die gleiche Ausgabe wie das Beispiel WhileDemo oder DoWhileDemo von den letzten Blättern.

An diesem Beispiel erkennt man schon den großen Vorteil der for-Schleifen bei Aufgaben, bei denen man irgendwelche Dinge zählen muss. Durch sie ist eine wesentlich kompaktere Formulierung möglich. Außerdem braucht sich man im Schleifenrumpf um das Verändern der Zählvariable kümmern.

Aufgabe: Ändere das Programm so, dass die Zahlen 30,27,24,21,18,15,12,9,6,3 ausgegeben werden.

Beispiel: Summieren bis zu einer vorgegebenen natürlichen Zahl

```
import Prog1Tools.IOTools;
```

```
public class Summieren{

    public static void main(String args[]){
        int EndZahl;
        int summe=0;
        EndZahl=IOTools.readInteger("Bitte Zahl eingeben: ");

        // Schleife
        for(int i=1;i<=EndZahl;i++){
            summe=summe+i;
        }

        // Ergebnis ausgeben
        System.out.println("Summe = "+summe);
    }
}
```

Aufgaben:

1. Berechne die Summe aller ganzen Zahlen zwischen einer Anfangszahl und einer Endzahl mit Hilfe einer for-Schleife.
2. Schreibe ein Java-Programm, das eine einzugebende ganzzahlige Potenz einer vorgegebenen Zahl mit Hilfe einer Schleife berechnet.
3. a) Erstelle mit Hilfe eines Computerprogrammes für die Funktion $f(x) = 3 - \sin(0,5 \cdot (x+2))$ eine Wertetabelle für das Intervall $[-2; 11]$; die Schrittweite für x sei 1.
b) Wie muss das Programm für eine Schrittweite für x von 0,5 geändert werden?
c) Wie muss das Programm für eine flexible Schrittweite geändert werden?
4. Eine natürliche Zahl Z soll eingegeben werden. Der Computer hat zu untersuchen, ob diese Zahl eine Primzahl ist oder nicht.
Überlege: An welchen Stellen Ihres Programms lässt sich Rechenzeit einsparen durch Vermeiden unnötiger Berechnungen?
(Eine natürliche Zahl Z ist eine Primzahl, wenn sie größer als 1 ist und nur sich selbst und die Zahl 1 als Teiler besitzt.)
5. Schreibe ein Java-Programm, das für einen eingegeben Geldbetrag, Zinssatz und eine vorgegebene Dauer in Jahren die Zinsen berechnet.
6. Schreibe ein Java-Programm, das alle pythagoräische Tripel bis zu einer oberen Grenze durch Probieren bestimmt, d.h. gesucht sind alle Tripel $(a; b; c)$ natürlicher Zahlen, bei denen die Summe der Quadrate der ersten beiden Zahlen a und b das Quadrat der dritten Zahl c ergibt: $a^2 + b^2 = c^2$.
Achtung: Verwende nicht zu große Werte für die Obergrenze c , da sonst sehr lange Rechenzeiten auftreten!