

Zählschleifen mit **while**, **for** und **do**:

Für Zählschleifen eignet sich die **for** – Schleife am Besten, weil **alle** Schleifenanweisungen übersichtlich in der ersten Zeile untergebracht sind.

for (*Initialisierung* ; *Bedingung* ; *Weiterzählen*)
 Befehl

Beispiel: **for** (i=1 ; i<= n ; i++)
 Befehl;

Das Selbe würde mittels **while** – Schleife realisiert wie folgt aussehen:

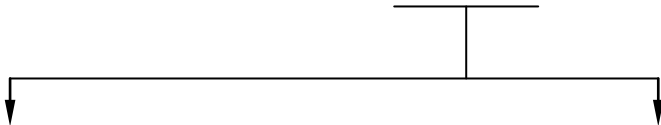
Initialisierung ;
while (*Bedingung*); {
 Befehl;
 Weiterzählen;}

Beispiel: i=1;
 while (i<= n) {
 Befehl;
 i++; }

- 17. Übung:** a) Schreiben Sie ein Programm das bis zu einem **vorgegebenen Endwert zählt**. Der Endwert soll vom Benutzer eingegeben werden. Verwenden Sie in der Zählschleife die **for** - Anweisung.
- b) Realisieren Sie die obige Aufgabe mit der **while** – Schleife.

Benutzung von zwei Zählervariablen:

Für manche Anwendung möchte man **zwei** Zählervariablen benutzen. Zu diesem Zweck werden die Variablen mittels „**Komma**“ getrennt.


for (*Initialisierung1* , *Initialisierung2* ; *Bedingung* ; *Weiterzählen1* , *Weiterzählen2*)

18. Übung: Schreiben Sie ein Programm das unter Verwendung von **zwei Variablen** bis zu einem vorgegebenen Endwert zählt.

Die Variable 1 wird um **1** erhöht, die Variable 2 um **2** erhöht.

Die Ausgabe sieht wie folgt aus:

Bis zu welchem Wert soll gezählt werden? **3**

1 2

2 4

3 6

Benutzung von **do** und **while**:

Ein weitere Möglichkeit für Schleifen ist mit **do** gegeben.

Bei **while** und **for** wird getestet und nur wenn die Bedingung erfüllt ist, wird der Schleifenblock durchlaufen.

```
do {  
    Befehl(e);  
} while (Ausdruck);
```

Der Schleifenblock wird **garantiert einmal ausgeführt**, bevor die Bedingung getestet wird.

19. Übung: Eine Primzahl ist eine Zahl, die nur durch 1 und sich selbst ohne Rest teilbar ist. Diese Definition erhält schon alles was für ein Algorithmus benötigt wird. Es werden alle Zahlen **i** von **2** bis zahl -1 durchprobiert. Wenn sich eine Zahl **i** findet, für die bei der Division kein Rest bleibt, wissen wir, dass es **keine Primzahl** sein kann. Wenn wir keinen solchen Teiler finden, handelt es sich bei um eine **Primzahl**.

Schreiben Sie ein Programm, dass alle Zahlen von 2 bis 499 durchprobiert und die Primzahlen ausgibt.