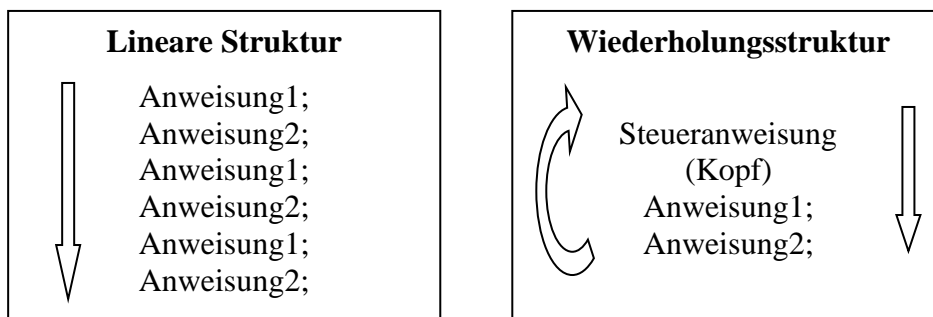


Wiederholungsstruktur

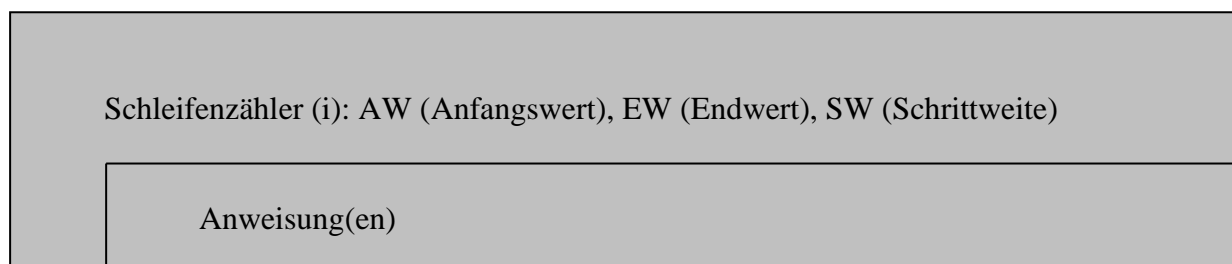
Eine Wiederholungsstruktur, meist als Schleife oder Iteration bezeichnet, besteht aus einer Folge von Anweisungen, die mehrfach durchlaufen werden kann. Ihrem Aufbau nach unterteilt man die Schleife in den Schleifenkopf und den Schleifenrumpf. Der Schleifenkopf enthält die Kontrollinformation für die Anzahl der Schleifendurchläufe; die mehrfach zu durchlaufende Anweisungsfolge heißt Schleifenrumpf. Will man zwei Anweisungen beispielsweise dreimal durchführen, könnte man in einer linearen Struktur die Anweisungen mehrfach hintereinander codieren. Einfacher und flexibler ist es aber, die beiden Anweisungen in einer Schleife nur einmal zu codieren und im Schleifenkopf die Bedingung für die Anzahl der Schleifendurchläufe festzulegen.



Geschlossene Schleife (zählergesteuerte Schleife)

Das wesentliche Kennzeichen der geschlossenen Schleife ist, dass die Anzahl der Schleifendurchläufe zum Zeitpunkt des Eintritts in den Schleifenrumpf in der Codierung bereits feststeht. Um eine solche Schleife zu realisieren verwendet man in C/C++ die For-Schleife.

Die geschlossene Schleife wird in einem Struktogramm entsprechend der folgenden Abbildung dargestellt.



Syntax:

```
for ([Initialisierungsausdruck], [Bedingungsausdruck], [Reinitialisierungsausdruck])  
{  
    Anweisung1;  
    Anweisung2;  
    ...  
}
```

Die Codierung der geschlossenen Schleife beginnt mit dem reservierten Wort *for*. Der Schleifenkopf enthält in runden Klammern – jeweils durch Semikolon getrennt – drei Ausdrücke:

1. Initialisierungsausdruck: Hier wird z.B. die Zählvariable auf einen Initialwert gesetzt. Diese Initialisierung wird lediglich beim ersten Durchlauf der Schleife ausgeführt.
2. Bedingungsausdruck: Im Bedingungsausdruck wird die Schleifenbedingung auf ihren Wahrheitsgehalt geprüft. Ist die Durchführungsbedingung erfüllt, so werden alle Anweisungen im Schleifenrumpf ausgeführt.
3. Reinitialisierungsausdruck: Mit dem dritten Ausdruck des Schleifenkopfes wird die Zählvariable reinitialisiert, d.h. auf einen neuen Wert gesetzt. Im Regelfall handelt es sich hierbei um ein Inkrementieren (=Erhöhen um 1 z.B. `i++`) oder Dekrementieren (=Verringern um 1 z.B. `i--`) des Schleifenzählers.

Solange die Bedingungsüberprüfung den Wahrheitswert *true* liefert, wird die Schleife weiter durchlaufen. Ergibt die Überprüfung der Bedingung einen Wert von 0 (= *false*), so wird die Schleife verlassen und die nächste Anweisung nach dem Schleifenkonstrukt durchgeführt.

Sofern in der Wiederholungsstruktur mehrere Anweisungen notwendig sind, muss der Anweisungsblock in geschweiften Klammern stehen. Ist jedoch lediglich eine Anweisung mehrfach durchzuführen, so kann man auf die Klammern verzichten.