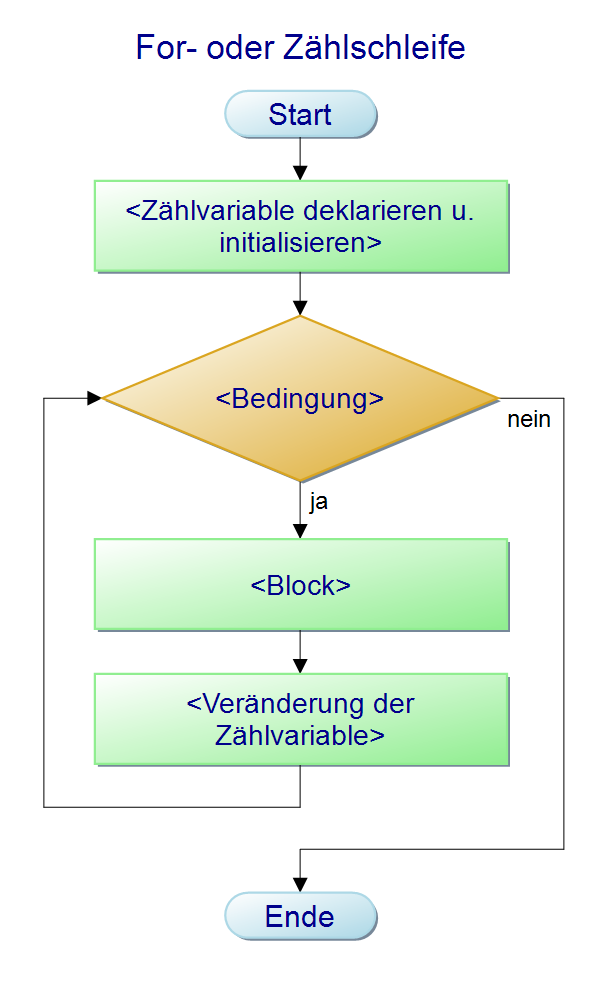
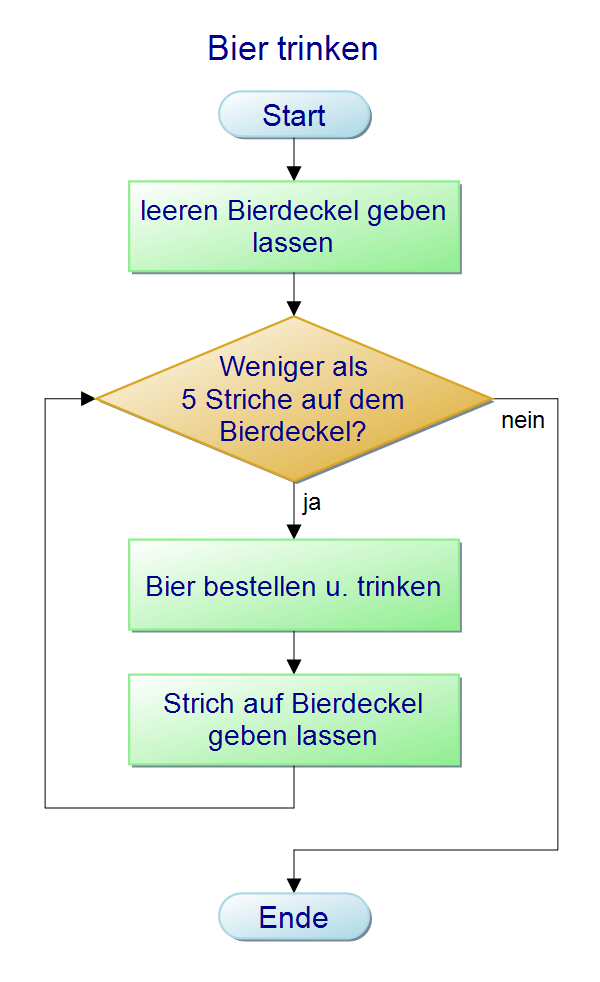
# PAP vom Bier trinken Beispiel zur allgemeinen Zählschleife



Verallgemeinerung für die for- oder Zählschleife:

Der Bierdeckel wird allgemein zur Zählvariablen.

Sie wird entsprechend initialisiert.

Die Frage nach der Anzahl der Strich auf dem Bierdeckel

ist eine allgemeine Bedingung.

Ist sie erfüllt, wird ein Block ausgeführt

u. die Zählvariable verändert.

Ist sie nicht mehr erfüllt,

wird die Struktur verlassen.

Notwendige Infos für die Syntax:

1. Eindeutiges Schlüsselwort
2. Deklaration u. Initialisierung einer Zählvariablen
3. Bedingung
4. Block mit sauberer Trennlinie
5. Veränderung an der Zählvariablen

Wir suchen im Hauptprogramm nach bereits vorhandenen ähnlichen Strukturen,

an denen wir die Syntax der herzuleitenden for-Schleife anlehnen können.

Die Hauptmethode "main" hat abstrahiert folgende Form:

… main ( … ){

<Block>

}

Daran angelehnt wollen wir die Syntax einer for-Schleife herleiten:

* Als Schlüsselwort benutzen wir "for" so wie bei der Hauptmethode "main".
* Wir benötigen einen Block mit sauberer Trennlinie,  
  um zu trennen, ob sich eine Anweisung innerhalb der Schleife befindet  
  u. somit mehrfach ausgeführt wird oder außerhalb.  
  In unserem Beispiel wäre das die Frage,  
  ob wir zu jedem Bier eine Brezel essen  
  oder nur einmalig eine auf dem Weg nach Hause.

Die schließende geschweifte Klammer gibt diese Grenze eindeutig an.

* Die restlichen Punkte packen wir unter dem Namen "Kontrollteil"  
  in die runden Klammern.  
  Damit ergibt sich die abstrakte Form der for- oder Zählschleife:

Abstrakte Form der for-Schleife:

Kontrollteil fasst zusammen, was kontrolliert,   
wie oft die Schleife durchlaufen wird.

for (<Kontrollteil>) {

<Block>

}

Konkrete Form der for-Schleife:

Kontrollteil umfasst die Punkte:

1. Deklaration u. Initialisierung der Zählvariablen, abgekürzt mit <Init>.
2. Bedingung, die erfüllt sein muss,   
   damit der Block der Schleife (weiter) ausgeführt wird,  
   abgekürzt mit <Bed>.
3. Veränderung an der Zählvariablen,   
   um eine Terminierung sicherzustellen,  
   abgekürzt mit <Veränderung>.

Die Teile werden durch einen Semikolon getrennt.

Syntax der for- oder Zählschleife:

**for (<Init>; <Bed>; <Veränderung>) {**

**<Block>**

**}**