# while- oder Bedingungsschleife

Wir haben als ersten Vertreter von Schleifen die for- oder Zählschleife kennen gelernt.

Wir wenden sie an, wenn wir mehrfach einen Block ausführen wollen,

wenn wir von vorneherein wissen wie oft.

Z.B:

* Genau 5 Bier trinken
* Ausgabe der Zahlen von 1 bis 10 oder umgekehrt
* Ausgabe der ersten 100 Vielfachen von 17

Was machen wir in einem Fall,

in dem wir zwar auch mehrfach etwas tun wollen,

aber von vorneherein nicht wissen wie oft?

Fragestellung z.B.:

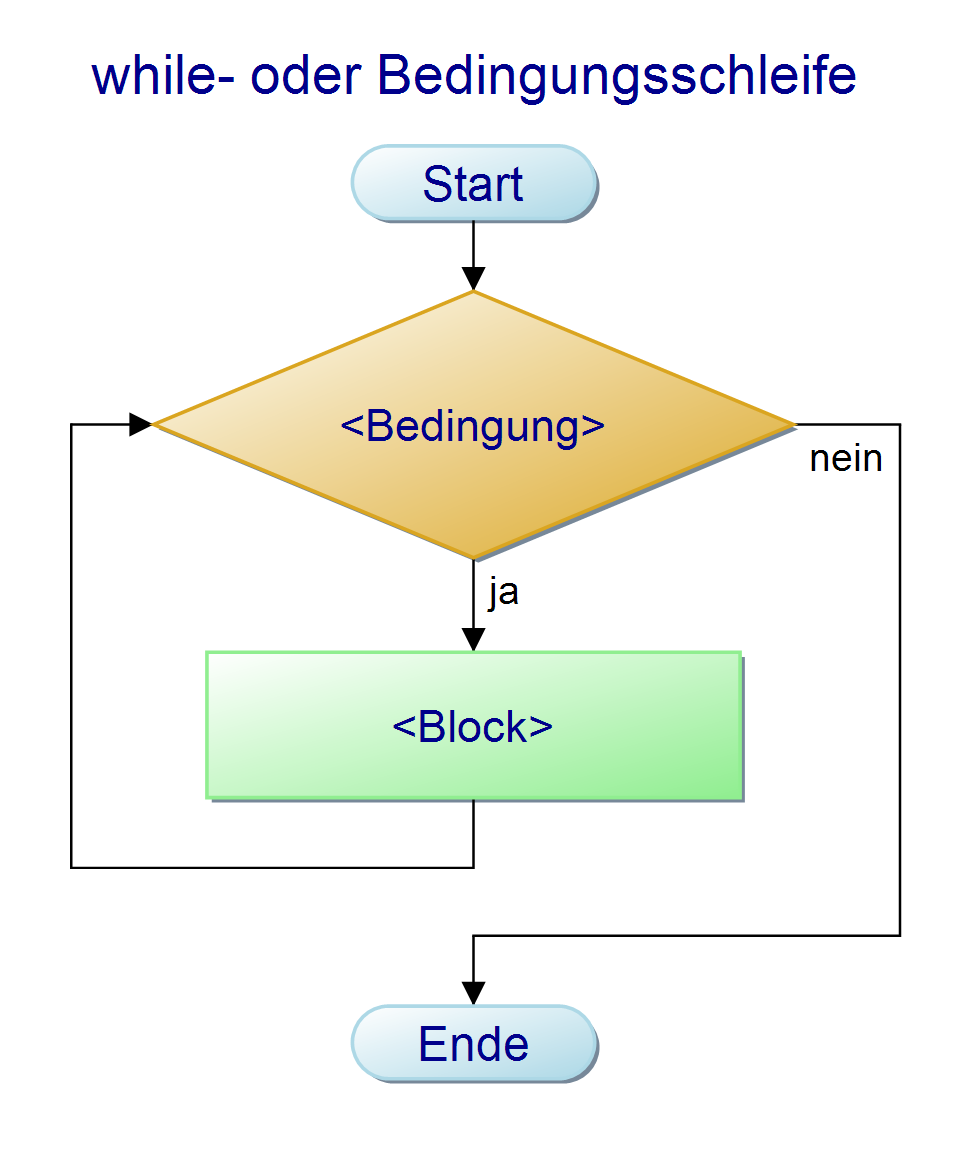
* Ausgabe der Vielfachen von einer Zahl, die kleiner als eine Obergrenze sind
* Welches ist die kleinste Zahl, deren Fakultät größer als 1 Million ist

Dazu brauchen wir eine neue Struktur:

Die while- oder Bedingungsschleife.

Wir leiten sie - wie schon for-Schleife u. bedingte Anweisungen - am PAP her:

PAP einer while- oder Bedingungsschleife:



Notwendige Infos:

1. eindeutiges Schlüsselwort
2. Bedingung
3. Block mit sauberer Grenze

Herleitung der Syntax in Anlehnung an die einseitig bed. Anw.:

if (<Bed>){

<Block>

}

Wir brauchen nur ein anders Schlüsselwort: while

Syntax der while- oder Bedingungsschleife:

while (<Bed>) {

<Block>

}

Vergleich der beiden Schleifen:

while-Schleife: for-Schleife:

**while (<Bed>) {**  **for (<Init>;<Bed>;<Veränd>) {**

**<Block>** **<Block>**

**}** **}**

Die for-Schleife gibt uns mehr Vorgaben:

Neben der Bedingung für die Variable auch

* Deklaration u. Initialisierung einer Zählvariablen
* Veränderung der Zählvariablen,  
  um eine Terminierung sicher zu stellen

Die while-Schleife lässt uns "mehr künstlerische Freiheit",

fordert aber auch mehr Eigenverantwortung,

weil auch bei ihr mit einer Variable gearbeitet werden muss,

die bei jedem Durchlauf eine Veränderung erfahren muss,

um eine Terminierung sicher zu stellen.