

# Kontrollstrukturen in Programmen III

## Zyklen (Schleifen, Wiederholungen) mit Bedingungsprüfung

Beim bedingten Zyklus steht die Anzahl der Durchläufe nicht fest. Der Durchlauf dieses Zyklus erfolgt so oft, wie die Bedingung erfüllt ist.

**Abweisender Zyklus (kopfgesteuerte Schleife)**

Die Anweisungen innerhalb der Schleife werden ausgeführt (durchlaufen), solange die vorangestellte Bedingung erfüllt ist. Ist sie nicht mehr erfüllt, wird die Aktion nicht mehr ausgeführt, die Wiederholung ist insgesamt beendet.

*Anders ausgedrückt:*

Die Anweisungen innerhalb der Schleife werden so lange durchlaufen, bis die vorangestellte Bedingung zum ersten Mal nicht mehr erfüllt ist. D.h. jedes mal bevor der Zyklus abgearbeitet werden soll, wird die Bedingung geprüft. Ist sie erfüllt, werden die Anweisungen ausgeführt - wenn nicht, wird der Zyklus einfach übersprungen.

### Darstellungsform (vereinfacht)

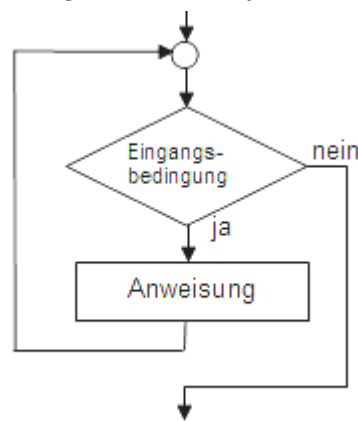
*Sprachliche Formulierung:*

**Solange die Bedingung erfüllt ist, führe aus:**

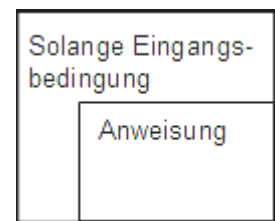
- Anweisung 1
- Anweisung 2
- Anweisung 3
- ...
- ...

**Ende**

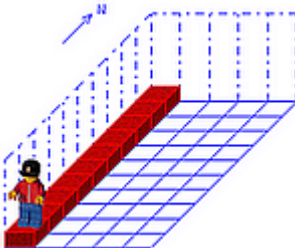
*Programmablaufplan:*



*Struktogramm:*



Beispiel (Programmiersprache):

Karol	Pascal (vereinfacht)
<pre> LinksDrehen Schritt LinksDrehen LinksDrehen Hinlegen Schritt LinksDrehen solange NichtIstWand tue   Hinlegen   Schritt *solange                     </pre> 	<pre> <b>WHILE</b> Eingabe &lt;&gt; 'Stopp' <b>DO</b>   BEGIN     WRITE ('Geben Sie "Stopp" ein, um das            Programm zu beenden: ');     READLN (Eingabe);   END; <b>END.</b>                     </pre>

### Aufgabe:

Karol soll dieses Fundament bauen!  
(7 Programmzeilen wären optimal!)

*Hinweis:* Kombiniere den Zählzyklus mit dem abweisenden Zyklus!

