Algorithmen mit Wiederholungsanweisungen

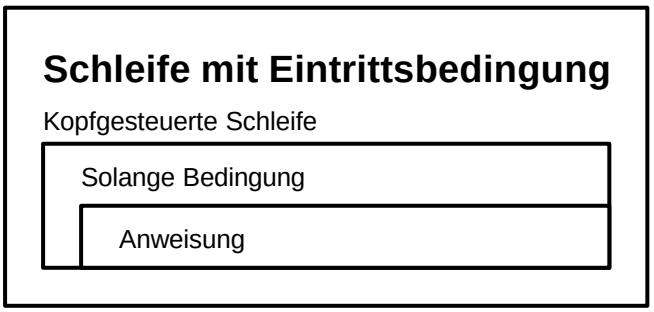
Die *Wiederholungsanweisung* oder *Schleife* ist neben den Entscheidungen ein wichtiges Mittel zur Ablaufsteuerung von Algorithmen. Prinzipiell unterscheidet man zwei verschiedene Arten von Schleifen, je nachdem ob die Anzahl der Wiederholungen von vornherein feststeht (*Zählschleife*), oder ob sich die Anzahl der notwendigen Schleifendurchläufe erst während der Ausführung ergibt.
Der Ausdruck Schleife stammt übrigens noch aus der Zeit des deutschen Computerpioniers KONRAD ZUSE (1910 – 1995), der seine Programme über Lochstreifen eingelesen hat. Musste eine Wiederholungsanweisung durchgeführt werden, so klebte er einfach die Enden des Lochstreifens zusammen wodurch die Maschine ständig die gleichen Anweisungen ausführte.

1. Wiederholung mit Ein – und Austrittsbedingung

Jede Schleife besteht aus einer Anweisung (oder einer Anweisungsfolge), die mehrfach auszuführen ist (*Schleifenkörper*) und einer *Abbruchbedingung*, die bei Erfüllung den Abbruch der Wiederholung bewirkt. Je nachdem wo die Abbruchbedingung steht unterscheidet man zwei verschiedene Arten von Schleifen:

- Eine Schleife mit *Eintrittsbedingung*. Hier steht die Abbruchbedingung im Kopf der Schleife und regelt somit das Eintreten in dieselbige.
- Eine Schleife mit *Austrittsbedingung*. Die Abbruchbedingung steht hier im Fuß der Schleife und regelt somit das Austreten aus der Schleife.

Die Schleife mit Eintrittsbedingung



Solange <Bedingung> **wiederhole**
<Anweisung>
Ende – wiederhole

Solange der Bedingungsteil den Wahrheitswert **`true`** besitzt, wird der Wiederholungsteil ausgeführt.

`While` <Bedingung> **`do`** <Anweisung>

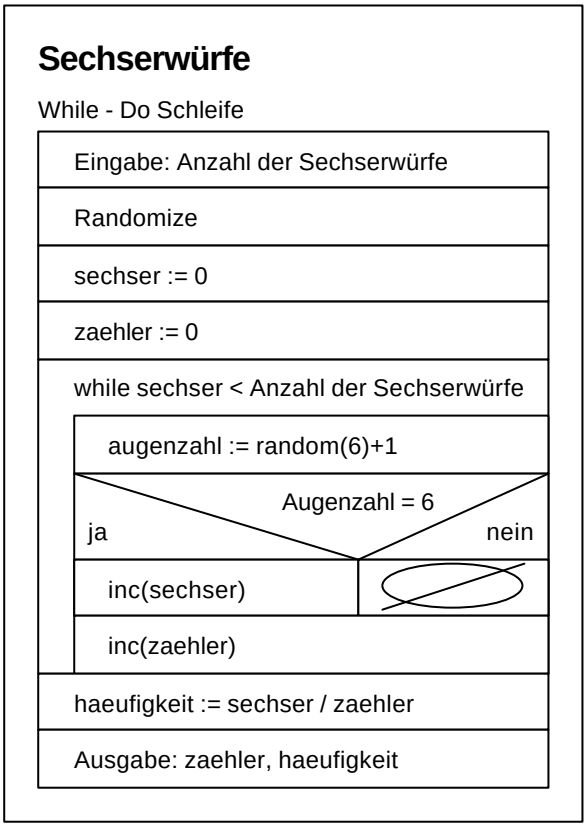
Hinweise:

- Besteht der Wiederholungsteil aus mehreren Anweisungen, so ist dieser mit **`begin`** und **`end`** zu klammern.
- Ist die Bedingung schon zu Beginn des Schleifenaufrufs nicht erfüllt, so wird die Schleife nicht ein einziges mal durchlaufen.
- Jede Schleife besteht im allgemeinen aus folgenden Elementen:
  - Variable auf Anfangswert setzen (*initialisieren*)
  - Abbruchbedingung prüfen
  - Anweisungen des Schleifenkörpers ausführen (dies beeinflusst die Veränderung mindestens einer Variablen)

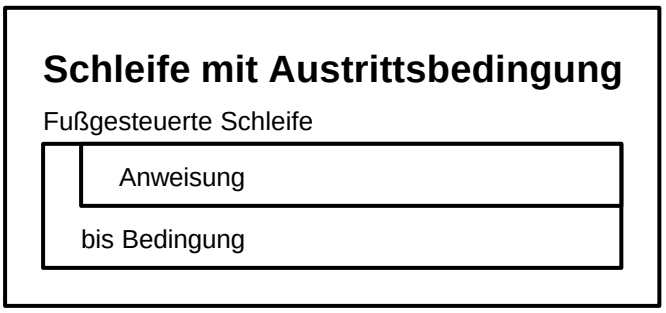
Ein Beispiel:

Es soll solange mit einem Würfel gewürfelt werden, bis eine zuvor eingegebene Zahl von Sechsen erreicht ist. Danach soll die Gesamtanzahl der Würfe und die relative Häufigkeit der Augenzahl 6 ausgegeben werden.[1] Klar ist, dass die Anzahl der durchzuführenden Würfe zunächst nicht bekannt ist. Eine Zählschleife mit fest vorgegebener Anzahl von Wiederholungen scheidet damit aus.

Struktogramm:



Die Schleife mit Austrittsbedingung



Wiederhole
<Anweisung>
bis Bedingung

Der Wiederholungsteil wird solange ausgeführt bis die Bedingung den Wahrheitswert **`true`** annimmt.

`Repeat`
<Anweisung>
`until` <Bedingung>

[1] Junger, Rapp, Rathfelder, Reinelt: *Informatik*, Diesterweg / Sauerländer 1988, Beispiel 2 auf Seite 69

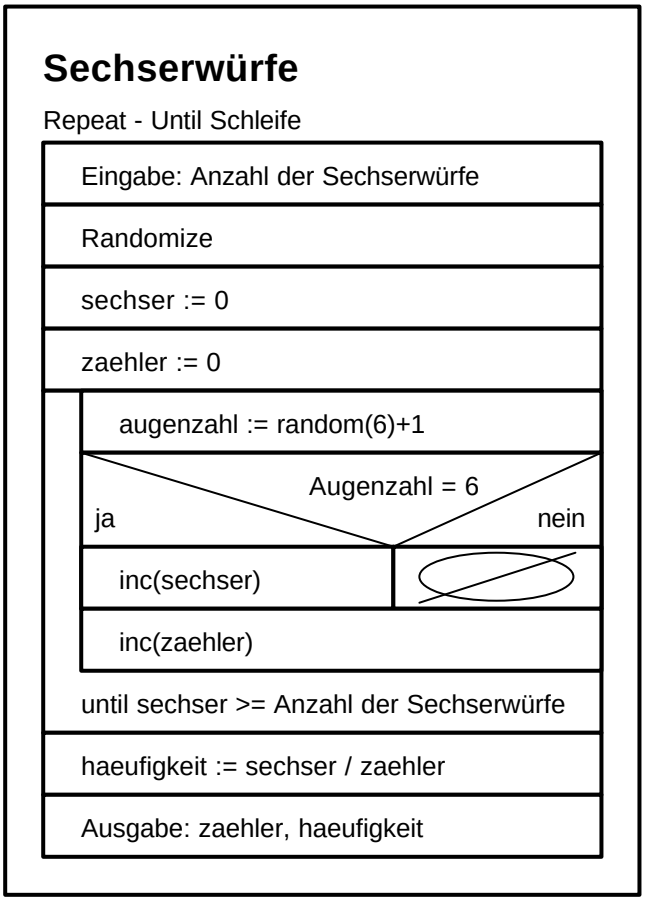
Hinweise:

- Besteht der Schleifenkörper aus mehreren Anweisungen, ist ein Klammern der Anweisungen mit **`begin`** und **`end`** nicht notwendig, da **`repeat ... until`** diese Funktion bereits erfüllen.
- Im Unterschied zur Schleife mit Eintrittsbedingung erfolgt das Prüfen der Bedingung erst im Fuß der Schleife. Dies hat zur Folge, dass der Schleifenkörper mindestens einmal durchlaufen wird!

Ein Beispiel:

Es soll wiederum das Würfeln mit einem Würfel simuliert werden (s.o.).
Überlegen Sie zunächst wie die Abbruchbedingung lauten muss!

Struktogramm:



Wie Sie sehen, kann man eine kopfgesteuerte Schleife in eine fußgesteuerte Schleife überführen, indem man den in der Abbruchbedingung stehenden booleschen Ausdruck `sechser < Anzahl der Sechserwürfe` negiert, also `NOT(sechser < Anzahl der Sechserwürfe)` oder `sechser >= Anzahl der Sechserwürfe` schreibt und sie nach unten verschiebt. Entsprechend kann man natürlich auch eine fußgesteuerte Schleife in eine kopfgesteuerte Schleife umschreiben.

Aufgabe: Erstellen Sie das Programm mit den beiden Varianten (kopfgesteuerte und fußgesteuerte Schleife)!