

Arbeitsblatt Nr. 6	TAF 11.3: Strukturierte Programmierung		B S G G
Datum:	Thema: Algorithmus, Kontrollstrukturen und Symbolik		
Seite 1 von 4	Name:		

Algorithmus

Der Begriff „Algorithmus“ ist bereits in der Einführung gefallen. Ein Algorithmus ist erst mal nichts anderes als eine Handlungsanweisung, die für ein bestimmtes Problem eine Lösung liefert.

Ein klassisches Beispiel für einen Algorithmus ist ein Kochrezept. Das Problem ist die Herstellung eines bestimmten Gerichtes (z.B. Nudeln kochen) und das Rezept ist der Algorithmus, der mir dieses Problem löst, in dem ich die Nudeln gemäß dem Rezept koche.

In der Informatik ist das ähnlich, nur sind die Handlungsanweisungen eben Befehle an den Computer, wie er aus bestimmten Eingabedaten ein Ergebnis berechnet. Hierbei haben Algorithmen in der Informatik noch bestimmte Besonderheiten.

Ein Beispiel: Die Anweisungen in einem Computer-Algorithmus sind nicht interpretierbar, d.h. jede Anweisung ist eindeutig. In einem Rezept könnte stehen, dass ein gehäufter Teelöffel Salz an die Nudeln zu geben ist. Das ist nicht eindeutig. Wie groß ist denn der Teelöffel? Wie groß ist der Haufen? All das ist unbestimmt und eben interpretierbar!

Jeder Algorithmus lässt sich durch die nachfolgend genannten drei Kontrollstrukturen umsetzen.

Kontrollstrukturen und Symbolik

Prinzipiell existieren in Software-Programmen drei grundlegende Kontrollstrukturen¹:

1. Sequenz
2. Bedingte Anweisung oder Entscheidung
3. Wiederholung

Der Ablauf eines Software-Programms kann grafisch dargestellt werden. Hierzu gibt es zwei wichtige genormte Darstellungsformen, die noch aus der nicht-objektorientierten Welt herrühren:

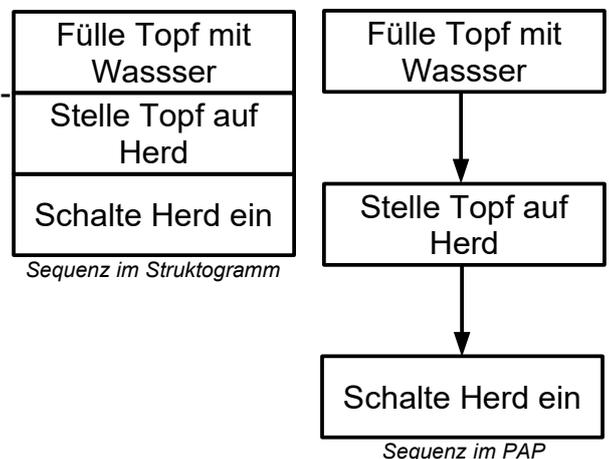
1. Programmablaufplan (PAP)
2. Nassi-Shneiderman-Diagramm (NSD) oder auch Struktogramm genannt

Die jeweilige Symbolik wird zumeist benutzt, um die Abfolge von Anweisungen, z.B. bei Algorithmen, darzustellen.

Sequenz

In den bisherigen recht einfachen Programmen, haben Sie diese grundlegende Struktur bereits kennen gelernt. Die Kontrollstruktur „Sequenz“ ist nichts anderes als eine Abfolge von Befehlen.

Sowohl im PAP als auch im Struktogramm wird die Abfolge grafisch durch ein Rechteck dargestellt. IM PAP sind die einzelnen Befehle durch Pfeile, die die Ablaufrichtung angeben, verbunden.



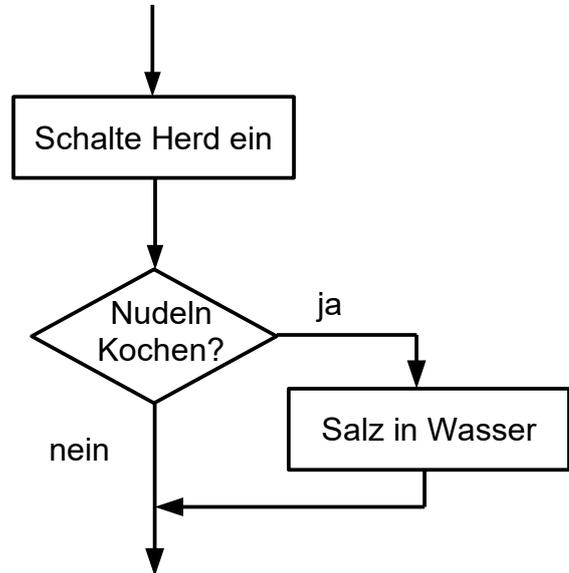
¹ So nennt z.B. <http://oszhdl.be.schule.de/gymnasium/faecher/informatik/algo-progr/kontrollstrukturen.html> drei Kontrollstrukturen während bei <http://de.wikipedia.org/wiki/Kontrollstruktur> nur von zwei Kontrollstrukturen gesprochen wird.
© Uwe Homm Version vom 25. Februar 2020 06 Algorithmen und Kontrollstrukturen.odt

Bedingte Anweisung oder Entscheidung

Bei der bedingten Anweisung wird aufgrund einer Bedingung eine Folge zusätzlicher Befehle ausgeführt.

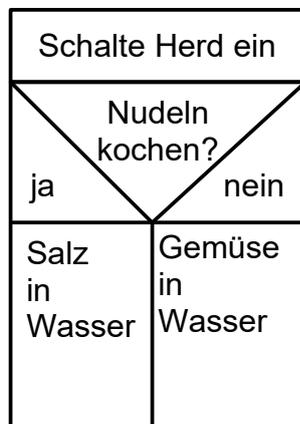


Bedingte Anweisung in Struktogramm

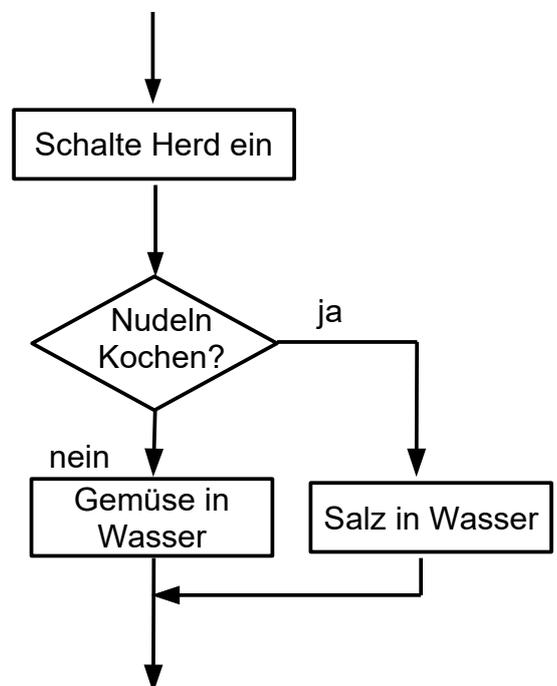


Bedingte Anweisung in PAP

Bei der Entscheidung werden je nach erfüllter Bedingung alternative Anweisungen ausgeführt:



Entscheidung in Struktogramm



Entscheidung in PAP

Wiederholung

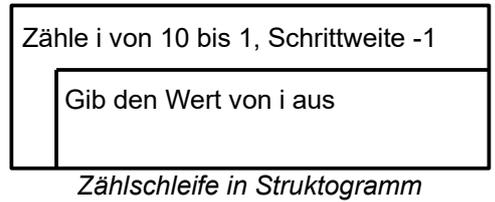
Bei der Wiederholung oder auch Schleife wird eine Befehlsfolge mehrfach wiederholt. Die Anzahl der Wiederholungen kann entweder bereits fest definiert (Zähl-schleife) oder an eine Bedingung geknüpft sein.

In zweiten Fall ist die Anzahl der Schleifendurchläufe im Vorhinein nicht festgelegt!

Für Wiederholungen gibt es im PAP keine eigenen Symbole. Hierzu benutzt man die Symbolik der Entscheidungen.

Zählschleife (FOR-Schleife)

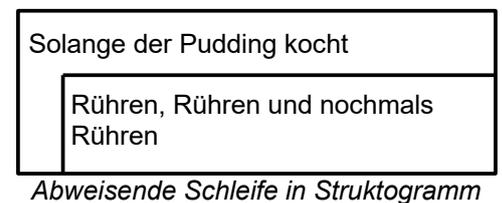
Bei der Zählschleife, die zumeist auch FOR-Schleife genannt wird, da der entsprechende Befehl in vielen Programmiersprachen FOR lautet, wird für eine Zählvariable ein Startwert festgelegt, der solange verändert wird, bis ein bestimmter Zielwert erreicht ist. Dann wird die Schleife verlassen. Im Schleifenrumpf wird eine Folge von Anweisungen definiert, die bei jedem Durchlauf ausgeführt wird.



Die Zählvariable darf im Schleifenrumpf nicht verändert werden!

Abweisende Schleife (WHILE-Schleife)

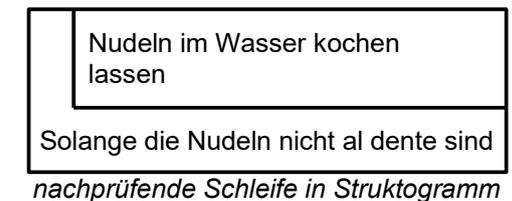
Die abweisende Schleife prüft vor dem Schleifenrumpf, ob die angegebene Bedingung erfüllt ist. Unabdingbar für diese (und auch die nachfolgende) Schleife ist, dass die Bedingung im Schleifenrumpf verändert wird.



Bei dieser Art von Wiederholung **kann** es sein, dass der Schleifenrumpf nicht ein einziges Mal durchlaufen wird, wenn die Bedingung bereits am Anfang nicht erfüllt ist!

Nicht abweisende Schleife (DO-WHILE-Schleife)

Die abweisende Schleife prüft nach mindestens einmaligem Durchlauf des Schleifenrumpfes, ob die angegebene Bedingung erfüllt ist. Somit wird hier der Schleifenrumpf mindestens ein Mal durchlaufen!



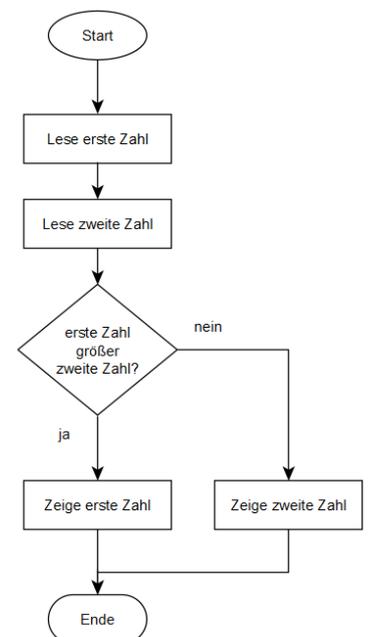
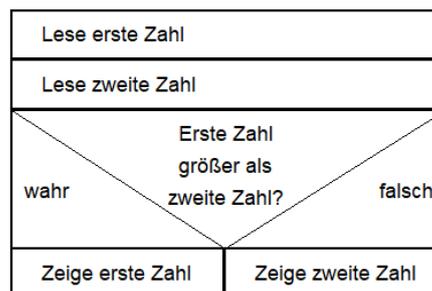
Beispiele

Zwei Zahlen sollen miteinander verglichen werden. Ist die erste Zahl größer als die zweite Zahl, soll die erste Zahl angezeigt werden; ansonsten die zweite Zahl.

Links wird der Algorithmus als Struktogramm dargestellt und rechts davon als Programmablaufplan.

Hier das Programm dazu:

Zahlenvergleich



```

// Demo für PAP und Struktogramm
// Die Konvertierung ist nicht
// im PAP/Struktogramm enthalten
int zahl1 = Convert.ToInt32(Console.ReadLine());
int zahl2 = Convert.ToInt32(Console.ReadLine());
if (zahl1 > zahl2)
    Console.WriteLine(zahl1);
else
    Console.WriteLine(zahl2);

```

Arbeitsblatt Nr. 6	TAF 11.3: Strukturierte Programmierung	 B S G G
Datum:	Thema: Algorithmus, Kontrollstrukturen und Symbolik	
Seite 4 von 4	Name:	

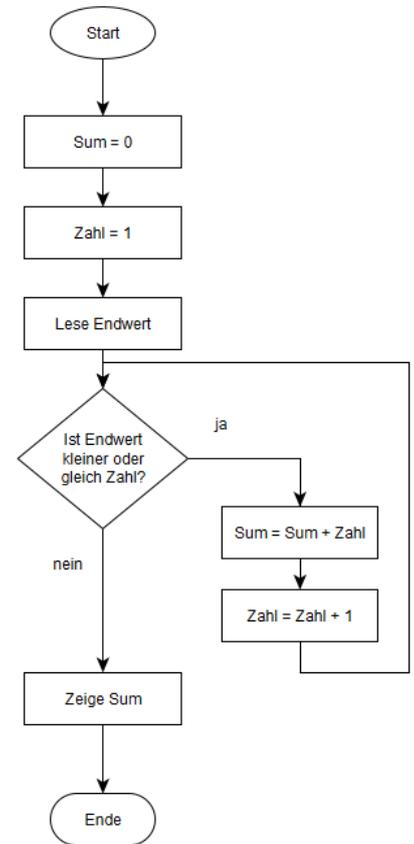
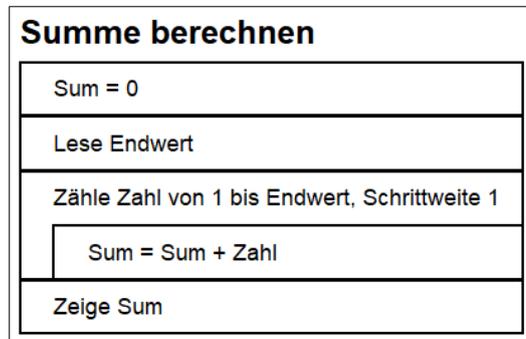
Es soll die Summe aller Zahlen von 1 bis zu einem Endwert, der einzugeben ist, berechnet und ausgegeben werden.

Hier sieht man auch, warum man Algorithmen häufig mittels Struktogramm darstellt.

PAPs werden schnell groß und unübersichtlich. Und außerdem lassen sich Wiederholungsstrukturen nur schlecht darstellen.

Hier das Programm:

```
// Demo für PAP und Struktogramm
// Die Konvertierung ist nicht
// im PAP/Struktogramm enthalten
int sum = 0;
int endwert = Convert.ToInt32(Console.ReadLine());
for (int zahl = 1; zahl <= endwert; zahl++)
{
    sum = sum + zahl;
}
Console.WriteLine(sum);
```



Übungen

Zeichnen Sie für die Aufgaben wahlweise ein PAP oder ein Struktogramm; oder beides :-)

1. Ein Algorithmus soll drei Zahlen (z1, z2 und z3) einlesen und davon die kleinste Zahl ermitteln und ausgeben.

2. Eine Multiplikation ist eigentlich eine mehrfache Addition.

Beispiel: $x \cdot y = y + y + y = p$ $3 \cdot 5 = 5 + 5 + 5 = 15$

Für diesen Algorithmus (Multiplikation als mehrfache Addition) sollen ein Multiplikator (x) und ein Multiplikand (y) eingelesen werden. Anschließend soll das Produkt (p) mittels mehrfacher Addition berechnet und angezeigt werden.

3. In der Mathematik gibt es die Fakultät einer Zahl. Dargestellt wird das durch ein Ausrufezeichen.

Beispiel: $x!$. Hat x z.B. den Wert 5, dann berechnet sich $5!$ wie folgt:

$$5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120.$$

Stellen Sie den Algorithmus zur Berechnung der Fakultät einer eingegebenen Zahl dar.

4. Der wahrscheinlich älteste Algorithmus der Welt ist der „euklidische Algorithmus“, der von zwei ganzen Zahlen den größten gemeinsamen Teiler ermittelt.

Informieren Sie sich über diese Algorithmus und skizzieren Sie diesen als Struktogramm.