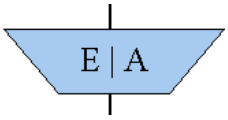
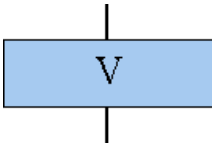
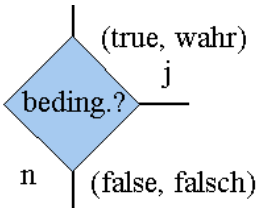
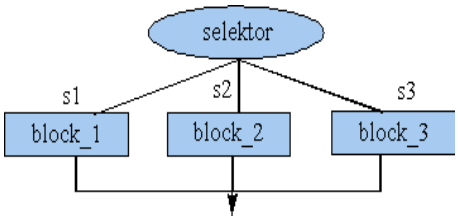
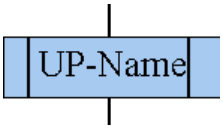
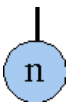
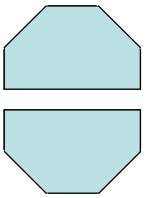
Flussdiagramm / Programmablaufplan (PAP)

Basissysmbole	
	**Grenzstelle** (Anfang, Zwischenhalt oder Ende des Programms/Algorithmus)
	**Verbindung** Zur Verdeutlichung der Ablaufrichtung werden Linien mit einer Pfeilspitze benutzt.
	**Ein- oder Ausgabe**
	Verarbeitungsschritt, sog. Block (einzelne Operation oder Gruppe von Operationen)
	**Alternative** **(Verzweigung auf Grund einer Entscheidung)** Ergebnis der Bedingung: j: ja (true, wahr) oder n: nein (false, falsch)
	**Selektor** (Mehrfachauswahl)
	**Unterprogramm** (an anderer Stelle festgelegte, strukturierte Menge von Operationen)
	**Konnektor** (Übergang von / zu einer anderen Stelle des Programmsystems)



Schleifenbegrenzer

Zur Darstellung von Programmwiederholungen werden diese 2 Symbole benutzt, die den Anfang und das Ende eines Schleifenrumpfes kennzeichnen.

Beispiel:

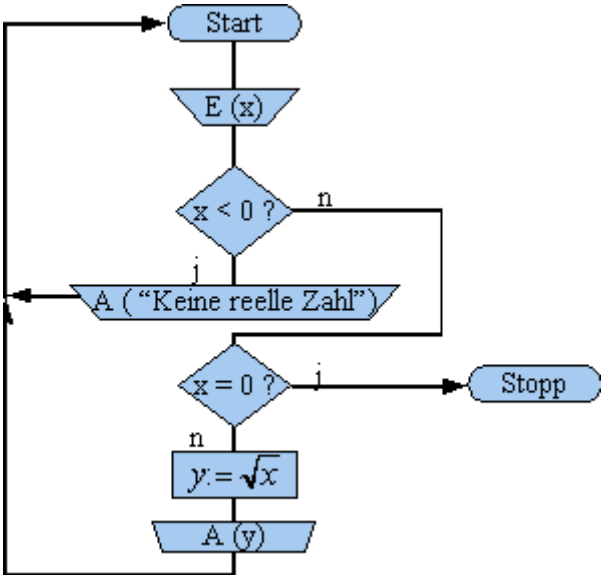
Aufgabe: $y := \sqrt{x}$

Eingabe einer Zahl
Test auf x > 0

Algorithmus:

Verarbeitungsschritte
x > 0 : Berechnung und Ausgabe y
x < 0 : Ausgabe "Keine reelle Zahl"
Eingabe der nächsten Zahl
Abbruch bei Eingabe x = 0

Lösung:



Wertzuweisung „:=“
(Ergibtanweisung)

Struktogramm

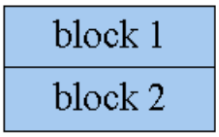
Allgemeines

- Grafisches Darstellungsmittel für logische Steuerstrukturen, sowie Algorithmen
- basiert auf den Grundelementen (Blockstruktur) der strukturierten Programmierung: *Sequenz, Selektion, Iteration*

Grundelemente des Struktogramms

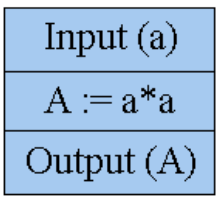
- *Sequenz* (Folge, sequentielle Reihung)
- *Selektion* (Alternative, Verzweigung, Auswahl)
  einfache (einseitige) Alternative
  vollständige (zweiseitige) Alternative
  Fallunterscheidung (CASE-Klausel, Mehrfachverzweigung)
- *Iteration* (Wiederholung, Schleife)
  Zählschleife
  Bedingungsschleife (abweisend, nicht-abweisend)

Sequenz (Folge, Reihung)



Sequentielle Ausführung

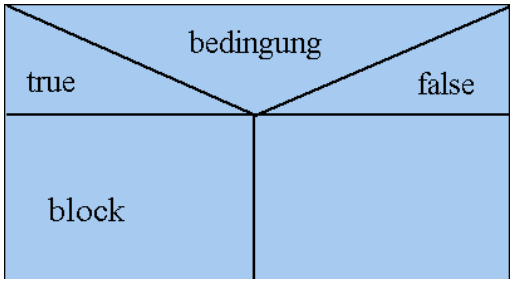
Beispiel: Berechnung des Flächeninhalts eines Quadrates ($A = a^2$)



Wertzuweisung für Variable a
(Eingabe: äußere Wertzuweisung

Selektion

1. einfache (einseitige) Alternative

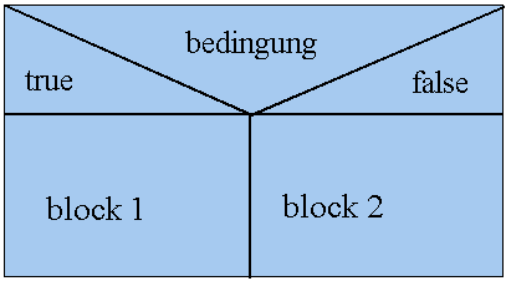


wenn "bedingung"

true - block ausführen

false - Keine Operation

2. vollständige (zweiseitige) Alternative

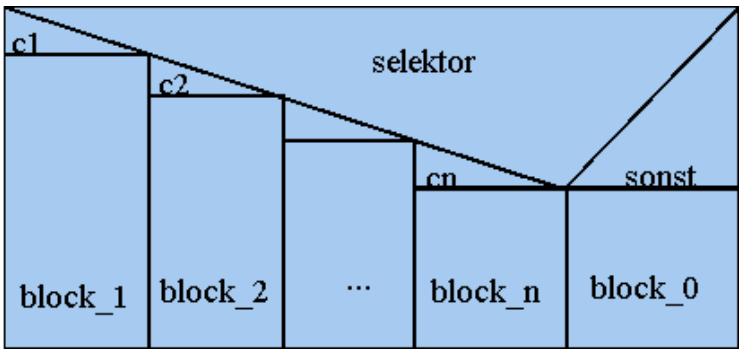


wenn "bedingung"

true - block 1 ausführen

false - block 2 ausführen

3. Fallunterscheidung (CASE-Klausel), Mehrfachverzweigung)

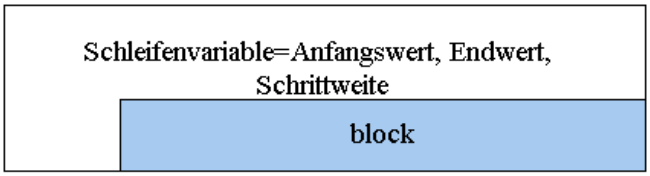


selektor: Selektor (Verteiler)

wenn einer der Selektoren c_i erfüllt ist, wird zugehöriger block_i ausgeführt,
sonst block_0 oder (falls sonst nicht angegeben) keine Anweisung; "sonst" kann demnach auch entfallen

Iteration

1. Zählschleife: Die Zyklenanzahl n ist bei Schleifeneintritt bekannt, das Ende der Schleife wird durch den Zähler bestimmt.



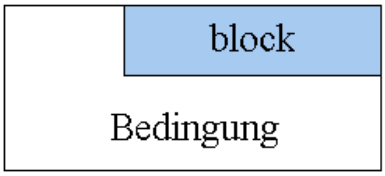
"block“ wird solange ausgeführt, bis Schleifenvariable den Endwert überschritten / erreicht (entspr. Sprachfestlegung) hat
Bei jedem Durchlauf wird der Wert der Schleifenvariable um die Schrittweite erhöht / erniedrigt (Standard: Schrittweite=1)

2. Bedingungsschleife: Die Anzahl der Zyklen ist bei Schleifeneintritt unbekannt. Das Ende der Schleife ist von einer Bedingung abhängig.

2a. abweisende Schleife: Zuerst wird die Bedingung geprüft und dann ggf. (wenn Bedingung *true*) der Stukturblock durchlaufen (Anzahl Schleifendurchläufe n $\geq$ 0).

"block" wird nur dann abgearbeitet (und zykl. wiederholt), ***solange*** "Bedingung" ***wahr*** ist

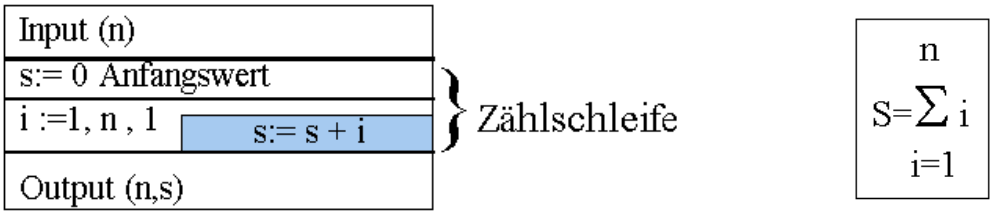
2b. nicht-abweisende Schleife: Zuerst wird der Strukturblock durchlaufen und dann die Bedingung vor erneutem Schleifendurchlauf geprüft. Die Wiederholung des Strukturblockes erfolgt, **bis** Bedingung **true (wahr)** ist (Anzahl Schleifendurchläufe n > 0).



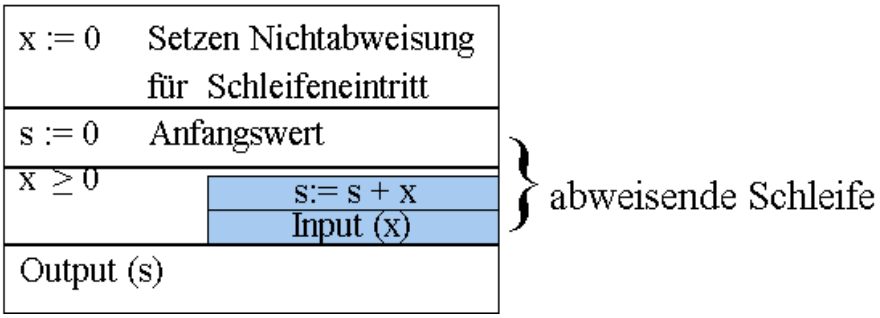
"block" wird wiederholt abgearbeitet, ***bis*** "Bedingung" ***wahr*** ist ("Bedingung" am Schleifenende geprüft)

Beispiel: Struktogramm (Iteration)

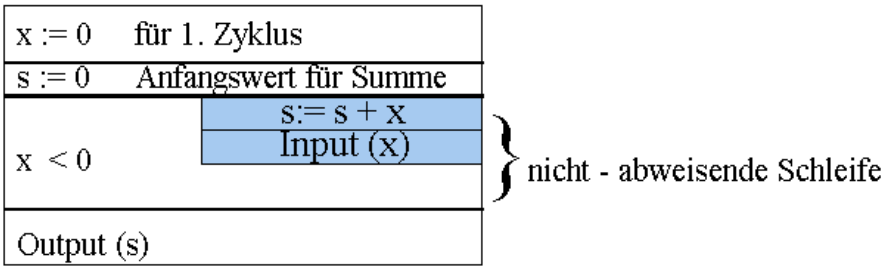
1. Zählschleife FOR { ... }
Gaußsche Schüleraufgabe: Summe der ganzen Zahlen von 1 bis n (n=100)



2. Bedingungsschleife (abweisend) WHILE { ... }
Es sind einzelne Zahlen (x) einzulesen und zu addieren (x ≥ 0). Abbruch, bei x<0



3. Bedingungsschleife (nicht abweisend) DO { ... } WHILE
Es sind einzelne Zahlen (x) einzulesen und zu addieren (x ≥ 0). Abbruch, bei x<0

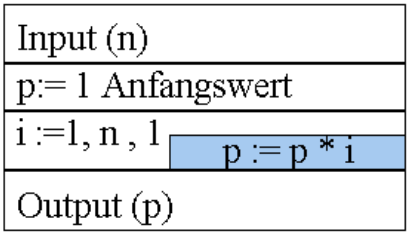


Beispiel: Struktogramm (Fakultät)

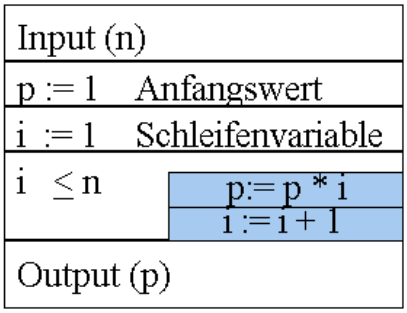
Berechnung der Fakultät von n
Lösung: Rekursive Berechnung
n! = n(n-1)! , 0! = 1

$$p = n! = \prod_{i=1}^{n} i$$

a. Lösung mittels Zählschleife



b. Lösung mittels abweisender Bedingungsschleife (WHILE { ... })
Schleife, solange i ≤ n



c. Lösung mittels nicht-abweisender Bedingungsschleife (DO { ... }WHILE)
Schleife, bis i > n