

Kontrollstrukturen, dargestellt im Nassi-Shneiderman-Diagramm

Linearer Ablauf (Sequenz)



Jede Anweisung wird in einen rechteckigen Strukturblock geschrieben. Die Strukturblocke werden nacheinander von oben nach unten durchlaufen. Leere Strukturblocke sind nur in Verzweigungen zulässig.

Verzweigung (Alternative)

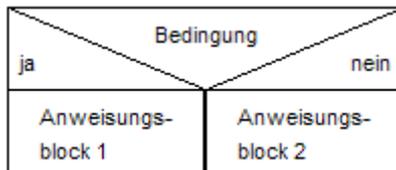
Einfache Auswahl [Einfache Verzweigung]



Alternativ: bedingte Verarbeitung, Selektion, einfache Selektion (if)

Nur wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Trifft die Bedingung nicht zu (falsch), wird der Durchlauf ohne eine weitere Anweisung fortgeführt (Austritt unten).

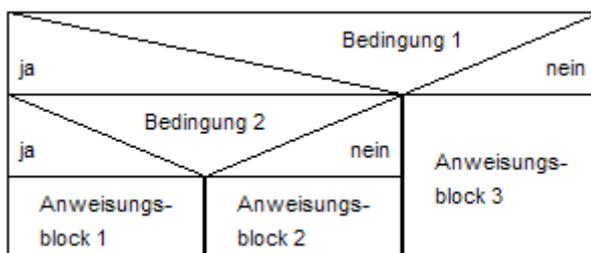
Zweifache Auswahl [Alternative Verzweigung]



Alternativ: alternative Verarbeitung, alternative Verzweigung (if then else)

Wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Trifft die Bedingung nicht zu (falsch), wird der Anweisungsblock 2 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Austritt unten nach Abarbeitung des jeweiligen Anweisungsblocks.

Verschachtelte Auswahl



Es folgt eine weitere Bedingung. Die Verschachtelung ist ebenso im Nein-Fall (noch) möglich.

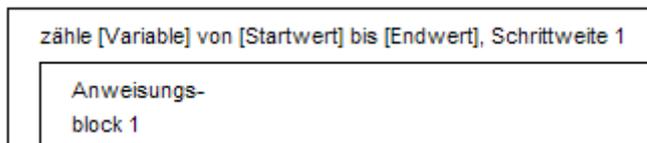
Fallauswahl

Besonders bei mehr als drei abzutestenden Bedingungen geeignet: Der Wert von „Variable“ kann bedingt auf Gleichheit wie auch auf Bereiche (größer/kleiner bei Zahlen) geprüft werden und der entsprechend zutreffende „Fall“ mit dem zugehörigen Anweisungsblock wird durchlaufen (switch, select). Eine Fallauswahl kann stets in eine verschachtelte Auswahl umgewandelt werden – etwa wenn die später eingesetzte Programmiersprache Fallauswahlen nicht kennt.

Wert(ebereich) 1	Wert(ebereich) 2	Wert(ebereich) 3	Wert(ebereich) n	Variable sonst
Anweisungs- block 1	Anweisungs- block 2	Anweisungs- block 3	Anweisungs- block n	Alternativ- block (optional)

Wiederholung (Iteration)

Zählergesteuerte Schleife (FOR-Schleife)



Wiederholungsstruktur, bei der die Anzahl der Durchläufe festgelegt ist (for). Als Bedingung muss eine Zählvariable angegeben und mit einem

Startwert initialisiert werden. Ebenso muss ein Endwert und die (Zähl-)Schrittweite angegeben werden. Nach jedem Durchlauf des Schleifenkörpers (Anweisungsblock 1) wird die Zählvariable um die Schrittweite vergrößert (bzw. bei negativer Schrittweite verkleinert) und mit dem Endwert verglichen. Ist der Endwert überschritten bzw. unterschritten, wird die Schleife verlassen.

Abweisende (vorprüfende – kopfgesteuerte) Schleife



Wiederholungsstruktur mit vorausgehender Bedingungsprüfung (while). Der Schleifenkörper (Anweisungsblock 1) wird nur durchlaufen, wenn (und solange) die Bedingung zutreffend (wahr) ist. Diese Symbolik wird auch für die Zählschleife (Anzahl der Durchläufe bekannt) benutzt.

Nicht abweisende (nachprüfende – fußgesteuerte) Schleife



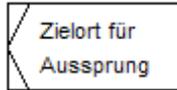
Wiederholungsstruktur mit nachfolgender Bedingungsprüfung für den Abbruch (loop). Der Schleifenkörper (Anweisungsblock 1) wird mindestens einmal durchlaufen, auch wenn die Bedingung von Anfang an nicht zutreffend (falsch) war.

Endlosschleife



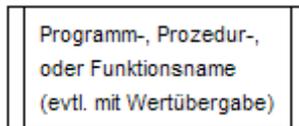
Wiederholungsstruktur, die allenfalls durch einen Aussprung (`break`) verlassen werden kann.

Aussprung



Der Aussprung (`break`) stellt die Beendigung eines Programmteils dar. Er sollte nicht mit der unbedingten Sprunganweisung (`goto`) verwechselt werden, die Nassi und Shneiderman mit den Struktogrammen vermeiden wollten.

Aufruf

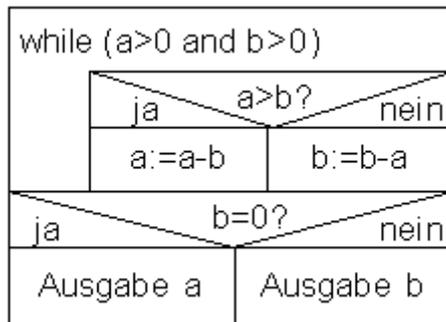


Symbol für den Aufruf eines Unterprogramms bzw. einer Prozedur, Funktion oder Methode. Nach deren Durchlauf wird zu der aufrufenden Stelle zurückgesprungen und der nächstfolgende Strukturblock durchlaufen. Dieses Symbol ist nicht genormt.

Einfaches Struktogramm

Das folgende Beispiel zeigt den Ablauf des euklidischen Algorithmus zur Berechnung des größten gemeinsamen Teilers zweier Zahlen.

als Nassi-Shneiderman-Diagramm



solange $(a > 0)$ und $(b > 0)$ ist, wiederhole
 ist $a > b$?
 wenn ja,
 dann weise a zu: $a - b$,
 wenn nein, dann weise b zu: $b - a$
 ist $b = 0$?
 wenn ja, gib aus a ,
 wenn nein, gib aus b