

Sinnbilder nach DIN 66261 [\[Bearbeiten\]](#)

Die meisten der nachfolgenden Strukturblöcke können ineinander geschachtelt werden. Das aus den unterschiedlichen Strukturblöcken zusammengesetzte Struktogramm ist im Ganzen rechteckig, also genauso breit wie sein breiter Strukturblock.

Linearer Ablauf (Sequenz) [\[Bearbeiten\]](#)

Anweisung 1	Jede Anweisung wird in einen rechteckigen Strukturblock geschrieben.
Anweisung 2	Die Strukturblöcke werden nacheinander von oben nach unten durchlaufen.
Anweisung n	Leere Strukturblöcke sind nur in Verzweigungen zulässig.

Verzweigung (Alternative) [\[Bearbeiten\]](#)

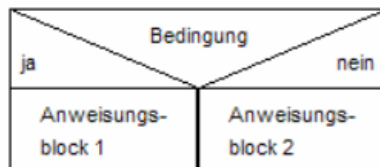
Einfache Auswahl [Einfache Verzweigung] [\[Bearbeiten\]](#)



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).

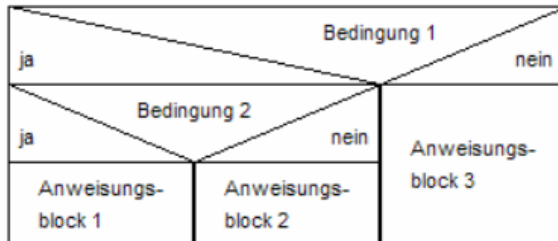
Zweifache Auswahl [Alternative Verzweigung] [\[Bearbeiten\]](#)



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 [\[Bearbeiten\]](#)



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

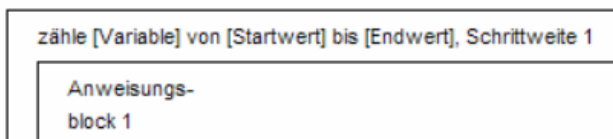
Fallauswahl [\[Bearbeiten\]](#)



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.

Wiederholung (Iteration) [\[Bearbeiten\]](#)

Zählergesteuerte Schleife [\[Bearbeiten\]](#)



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 inkrementiert (bzw. bei negativer Schrittweite dekrementiert) und mit dem Endwert verglichen. Ist der Endwert überschritten bzw. unterschritten, wird die Schleife verlassen.

Abweisende (vorprüfende – kopfgesteuerte) Schleife [\[Bearbeiten\]](#)



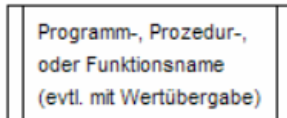
Wiederholungsstruktur mit vorausgehender Bedingungsprüfung (`while`). Der Schleifenkörper (Anweisungsblock 1) wird nur durchlaufen, wenn (und solange) die Bedingung zutreffend (wahr) ist. Diese Symbolik wird auch für die Zählschleife (Anzahl der Durchläufe bekannt) benutzt.

Nicht abweisende (nachprüfende – fußgesteuerte) Schleife [\[Bearbeiten\]](#)



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.

Aufruf [\[Bearbeiten\]](#)



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.

Füllregeln [\[Bearbeiten\]](#)

Allgemeingültigkeit [\[Bearbeiten\]](#)

Struktogramme sollten keine programmiersprachenspezifische Befehlssyntax enthalten. Sie müssen so programmiersprachenunabhängig formuliert werden, dass die dargestellte Logik einfach zu verstehen und als Codivorschrift in jede beliebige Programmiersprache umzusetzen ist.

Deklaration [\[Bearbeiten\]](#)

Weil sie ursprünglich für prozedurale Programmiersprachen entwickelt wurden, bildete man in Struktogrammen nur die Prozedur und keine Deklarationsbereiche von Variablen und Konstanten ab (einfaches Struktogramm). Dadurch ist jedoch nicht sofort deutlich, welcher Datentyp einer Variablen zugeordnet werden muss. Die Deklaration von Variablen und Konstanten ist im ersten Anweisungsblock vorzunehmen. Diese Nassi-Shneiderman-Diagramme bezeichnet man dann als erweiterte Struktogramme.

Exklusivität [\[Bearbeiten\]](#)

Jede Anweisung erhält einen eigenen Strukturblock (Sinnbilder nach [DIN 66261](#)). Selbst mehrere Anweisungen gleicher oder ähnlicher Art dürfen nicht in einem Strukturblock zusammengefasst werden.

Jede Anweisung muss mindestens aus einer Zuweisung bestehen (beispielsweise $Zielvariable \leftarrow Zielvariable * AndereVariable$). Eine Zuweisung wird durch einen nach links gerichteten Pfeil dargestellt. Ältere Struktogramme benutzen alternativ aus alten [Pascal](#)-Zeiten als Zuweisungszeichen den Doppelpunkt gefolgt vom Gleichheitszeichen ($Zielvariable := Zielvariable * AndereVariable$). Das Ziel einer Anweisung steht immer links vom Zuweisungszeichen. Rechts davon steht die Quelle.

Über jedes Struktogramm gehört ein Name, um die Identifikation durch Ereignis- oder (Unter-)Programmaufrufe gewährleisten zu können.