

- Kursive Ausdrücke sind allgemeine Platzhalter, optionale Erweiterungen stehen in senkrechten Strichen.
- Kontrollstrukturen (des gleichen oder unterschiedlichen Typs) können miteinander verschachtelt werden.

Operator im Struktogramm	Beschreibung seiner Verwendung
Deklaration: <i>variable</i> [als <i>datentyp</i>] z. B.: <div>Deklaration: alter als Ganzzahl</div>	Deklaration einer Variablen, optionale Angabe des Datentyps
Initialisierung: <i>variable</i> = <i>wert</i> z. B.: <div>Initialisierung: guthaben = 10</div>	Initialisierung einer Variablen mit einem Ausgangswert
Deklaration und Initialisierung: <i>variable</i> [als <i>datentyp</i>] = <i>wert</i> z. B.: <div>Deklaration und Initialisierung: anzahl als Ganzzahl = 0</div>	Kombination von Deklaration und Initialisierung
Zuweisung: <i>element</i> = <i>wert</i> z. B.: <div>Zuweisung: qm = laenge * breite</div>	Zuweisung eines Wertes (oder des Ergebnisses einer Berechnung o. ä.) zu einem Element, das eine Variable oder ein anderes Element (z. B. ein Ausgabefeld) sein kann
Einlesen: <i>variable</i> [als <i>datentyp</i>] z. B.: <div>Deklaration und Einlesen: betrag als Dezimalzahl</div>	Einlesen einer Eingabe, z. B. aus einem Eingabefeld, einer Kommandozeile, ...; auch in Kombination mit Deklaration möglich
Ausgabe: <i>inhalt</i> z. B.: <div>Ausgabe: "Die Fläche beträgt " + qm + " Quadratmeter. "</div>	Verwendung einer Ausgabeoption (z. B. Meldungsfenster, Konsole), die eine Variable, einen Array, einen Text oder eine Kombination aus diesen ausgibt
Zeilenweise Ausgabe: <i>inhalt</i> z. B.: <div>Ausgabe: "Hallo!" + Zeilenumbruch</div>	Ausgabe mit Zeilenumbruch
Rückgabe: <i>inhalt</i> z. B.: <div>Rückgabe: strecke</div>	Anweisung zur Rückgabe eines Wertes innerhalb einer Funktion/Methode/Prozedur
Aufruf: <i>methode/unterprogramm</i> (<i>parameter</i>) z. B.: <div>Aufruf: sortiereListe()</div> oder: <div>Aufruf: einzahlen(betrag)</div>	Aufruf einer Funktion/Methode/Prozedur/Unterprogramm, auch in Kombination mit anderen Operatoren (wie Zuweisungen oder Bedingungen), wenn Rückgabewerte verwendet werden sollen
Wenn <i>bedingung</i> , dann [...] , sonst [...] z. B.: <div> <div>alter > 18 AND geschlecht == "w"</div> <div> <div>J</div> <div>N</div> <div>...</div> <div>...</div> </div> </div>	Verzweigungs- bzw. Mehrfachauswahlbedingung

Wiederhole solange bedingung z. B.: <table><tr><td>Wiederhole solange inhalt > 10</td></tr><tr><td>...</td></tr></table>	Wiederhole solange inhalt > 10	...	Schleife mit vorausgehender Bedingungsprüfung („while-Schleife“)			
Wiederhole solange inhalt > 10						
...						
Wiederhole von startwert solange bedingung, Schrittweite schrittweite z. B.: <table><tr><td>Wiederhole von i = 0 solange i < 5, Schrittweite 1</td></tr><tr><td>...</td></tr></table>	Wiederhole von i = 0 solange i < 5, Schrittweite 1	...	Zählergesteuerte Schleife („for-Schleife“): Alternative Darstellung 1			
Wiederhole von i = 0 solange i < 5, Schrittweite 1						
...						
Zähle zählvariable von startwert bis endwert, Schrittweite schrittweite z. B.: <table><tr><td>Zähle i von 0 bis 4, Schrittweite 1</td></tr><tr><td>...</td></tr></table>	Zähle i von 0 bis 4, Schrittweite 1	...	Alternative Darstellung 2			
Zähle i von 0 bis 4, Schrittweite 1						
...						
Deklaration und Initialisierung: array [als Array] = [Arrayelement1, Arrayelement2, ...] z. B.: <table><tr><td>Deklaration und Initialisierung: personen = ["Sven", "Tina", "Anja"]</td></tr><tr><td>Deklaration und Initialisierung: zahlen als Array = []</td></tr></table>	Deklaration und Initialisierung: personen = ["Sven", "Tina", "Anja"]	Deklaration und Initialisierung: zahlen als Array = []	Kombination aus Deklaration und Initialisierung eines Arrays Deklaration und Initialisierung eines „leeren“ Arrays			
Deklaration und Initialisierung: personen = ["Sven", "Tina", "Anja"]						
Deklaration und Initialisierung: zahlen als Array = []						
Zuweisung: array[feldindex] = wert z. B.: <table><tr><td>Zuweisung: personen[2] = "Kay"</td></tr></table>	Zuweisung: personen[2] = "Kay"	Zuweisung eines Wertes (oder des Ergebnisses einer Berechnung o. ä.) zu einem Arrayelement, dessen Index in den eckigen Klammern angegeben wird				
Zuweisung: personen[2] = "Kay"						
Anhängen an ein Array Zuweisung: array[feldindex] = wert z. B.: <table><tr><td>Zuweisung: personen[3] = "Milo"</td></tr><tr><td>Zuweisung: zahlen[0] = 5</td></tr></table>	Zuweisung: personen[3] = "Milo"	Zuweisung: zahlen[0] = 5	Anhängen eines Arrayelements an einen Array, indem ein Wert einem Index zugeordnet wird, der noch nicht existiert			
Zuweisung: personen[3] = "Milo"						
Zuweisung: zahlen[0] = 5						
Anzahl der Elemente eines Arrays array z. B.: <table><tr><td>Deklaration und Initialisierung: laenge = Anzahl der Elemente des Arrays personen</td></tr></table>	Deklaration und Initialisierung: laenge = Anzahl der Elemente des Arrays personen	Zuweisung der Anzahl der Arrayelemente zu einer Variablen Auch in Kombination (bspw. mit Schleifen und Ausgaben) möglich				
Deklaration und Initialisierung: laenge = Anzahl der Elemente des Arrays personen						
Ausgabe: array[feldindex] z. B.: <table><tr><td>Ausgabe: zahlen[1]</td></tr><tr><td>Deklaration und Initialisierung: i = 0</td></tr><tr><td>Ausgabe: personen[i]</td></tr><tr><td>Ausgabe: personen[i + 2]</td></tr><tr><td>Ausgabe: zahlen</td></tr></table>	Ausgabe: zahlen[1]	Deklaration und Initialisierung: i = 0	Ausgabe: personen[i]	Ausgabe: personen[i + 2]	Ausgabe: zahlen	Ausgabe einzelner Arrayelemente Der gewünschte Index kann direkt, durch eine Berechnung oder durch eine Variable angegeben werden Ausgabe eines Arrays Der ganze Array kann über den Namen ausgegeben werden.
Ausgabe: zahlen[1]						
Deklaration und Initialisierung: i = 0						
Ausgabe: personen[i]						
Ausgabe: personen[i + 2]						
Ausgabe: zahlen						