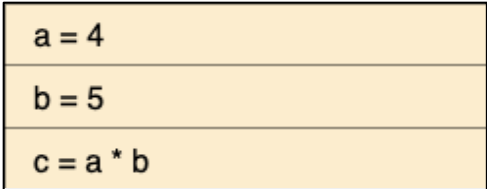
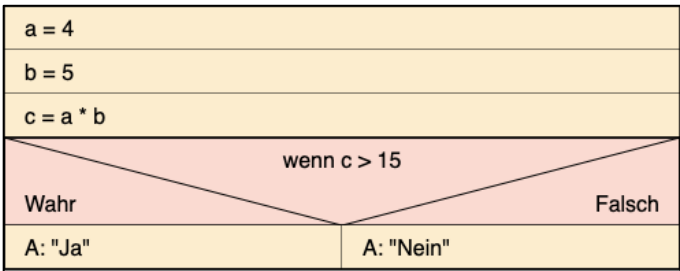


Struktogramme (Nassi-Shneiderman-Diagramme) mit Codebeispielen

Lineare Struktur		
		Ausführung der Anweisungen im rechteckigen Block nach und nach (sequenziell)
Code-Beispiele		
Python	Powershell	Pseudocode
<pre>a = 4 b = 5 c = a * b</pre>	<pre>\$a = 4 \$b = 5 \$c = \$a * \$b</pre>	<pre>a := 4 b := 5 c := a * b</pre>

Verzweigung		
		Wenn eine vorher definierte Bedingung zutrifft, wird der ja-Block durchlaufen. Ansonsten der nein-Block (if-else-Anweisung). Der nein-Block kann dabei auch leer bleiben (if-Anweisung).
Code-Beispiele		
Python	Powershell	Pseudocode
<pre>a = 4 b = 5 c = a * b if c > 15: print("Ja") else: print("Nein")</pre>	<pre>\$a = 4 \$b = 5 \$c = \$a * \$b if (\$c -gt 15) { Write-Host "Ja" } else { write-host "Nein" }</pre>	<pre>a := 4 b := 5 c := a * b wenn (c > 15) dann ausgabe "Ja" sonst ausgabe "Nein" Ende wenn</pre>

--	--	--

Fallauswahl – Mehrfachauswahl

tag = 3							
0	1	2	3	4	5	6	tag Sonst
ergebnis = "Sonntag"	ergebnis = "Montag"	ergebnis = "Dienstag"	ergebnis = "Mittwoch"	ergebnis = "Donnerstag"	ergebnis = "Freitag"	ergebnis = "Samstag"	ergebnis = ""
A: ergebnis							

Der Zustand einer Variable bestimmt den ausgeführten Fall. Es können auch mehrere Fälle ausgeführt werden. Trifft keiner der Fälle zu, so kann ein Alternativblock ausgeführt werden. (z.B. switch-case)

Code-Beispiele

Python	Powershell	Pseudocode
<pre> tag = 3 if tag == 0: print("Sonntag") elif tag == 1: print("Montag") elif tag == 2: print("Dienstag") elif tag == 3: print("Mittwoch") elif tag == 4: print("Donnerstag") elif tag == 5: print("Freitag") elif tag == 6: print("Samstag") else: print("") # ab Python 3.10 switch/match möglich </pre>	<pre> \$tag = 3 switch (\$tag) { 0 {\$ergebnis = "Sonntag"} 1 {\$ergebnis = "Montag"} 2 {\$ergebnis = "Dienstag"} 3 {\$ergebnis = "Mittwoch"} 4 {\$ergebnis = "Donnerstag"} 5 {\$ergebnis = "Freitag"} 6 {\$ergebnis = "Samstag"} default {\$ergebnis = ""} } Write-Host \$ausgabe # Powershell kann auch elseif </pre>	<pre> tag := 3 wenn (tag = 0) dann ausgabe "Sonntag" wenn (tag = 1) dann ausgabe "Montag" wenn (tag = 2) dann ausgabe "Dienstag" wenn (tag = 3) dann ausgabe "Mittwoch" wenn (tag = 4) dann ausgabe "Donnerstag" wenn (tag = 5) dann ausgabe "Freitag" wenn (tag = 6) dann ausgabe "Samstag" sonst ausgabe "" Ende wenn </pre>

Kopfgesteuerte Schleife		
<pre> graph TD Start([Start]) --> i0[i = 0] i0 --> a5[a = 5] a5 --> Cond{solange a < 5 ODER i < 3} Cond --> Body[A: "Nein" i = i+1] Body --> Cond </pre>		Die Anweisung wird so lange ausgeführt, wie die vorher definierte Bedingung(en) zutrifft / zutreffen. (z.B. while-Schleife)
Code-Beispiele		
Python	Powershell	Pseudocode
<pre> i = 0 a = 5 while a < 5 or i < 3: print("Nein") i = i+1 </pre>	<pre> \$i = 0 \$a = 5 while(\$a -lt 5 -or \$i -lt 3) { Write-Host "Nein" \$i = \$i+1 } </pre>	<pre> i := 0 a := 5 solange (a<5 or i<3) ausgabe "Nein" i = i+1 Ende solange </pre>

Fußgesteuerte Schleife		
<pre> graph TD Start([Start]) --> i1[i = 1] i1 --> Body[A: i ".Versuch" i = i+1] Body --> Cond{solange i < 3} Cond --> Body </pre>		Die Anweisung wird zunächst einmal durchlaufen, erst dann folgt die Prüfung einer Bedingung. Sollte die Bedingung erfüllt sein wird die Anweisung so lange durchlaufen, bis die Bedingung nicht erfüllt ist. (z.B. do-while-Schleife)
Code-Beispiele		
Python	Powershell	Pseudocode
<pre> i = 1 condition = True while condition == True: print(i, ". Versuch") i = i+1 if(i>=3): condition = False # do-while existiert in Python nicht, daher hier eine Emulation (auch mit break möglich) </pre>	<pre> \$i = 1 do { Write-host \$i". Versuch" \$i = \$i+1 } while (\$i -lt 3) </pre>	<pre> i := 1 wiederhole ausgabe i ". Versuch" i = i+1 solange (i < 3) </pre>

Zählergesteuerte Schleife		
<div> <div>zähle i von 1 bis 3, i+1</div> <div>A: i ". Versuch"</div> </div>		Die Schleife wird gemäß der Zählervariablen durchlaufen. Im Schleifenkopf wird der Start und Endwert, sowie die Veränderung der Zählervariable definiert. (z.B. for-Schleife)
Code-Beispiele		
Python	Powershell	Pseudocode
<pre>for i in range(1,4): print(i, ". Versuch")</pre>	<pre>for (\$i=1; \$i -le 3; \$i++){ Write-host \$i ". Versuch" }</pre>	<pre>zähle i von 1 bis 3, i+1 ausgabe i ". Versuch" Ende zähle</pre>

Tipp

Modellierungstool für Struktogramme

<https://t1p.de/y7wg3>


Pseudocode – Belegsatz IHK

Die Darstellung soll in allgemein verständlichem Pseudocode erfolgen. Der Code soll für Dritte ohne Kenntnis der verwendeten Programmiersprache lesbar sein. Der Code muss nicht in der geschriebenen Sprache kompilierbar bzw. ausführbar sein.

Pseudocode, prozedurale Programmierung

Schriftart: Courier New (12)

Pseudocode	Beschreibung
wenn [Bedingung] dann [Anweisung] sonst [Anweisung] Ende wenn	Verzweigung
zähle [Variable] von [Startwert] bis [Endwert] [Schleifeninhalt] Ende zähle	Zählschleife (Iteration)
solange [Bedingung] [Schleifeninhalt] Ende solange	Kopfgesteuerte Schleife
wiederhole [Schleifeninhalt] solange < Bedingung > :=	Fußgesteuerte Schleife
:=	Zuweisung
=; <; >; <=; >=; < >	Vergleiche
integer	
double	
string	
array	

Struktogramm – Belegsatz IHK

DIN 66261 Struktogramm

