

1. Einführung in die Informatik

2. Algorithmen

- Definition, Eigenschaften, Entwurf
- **Darstellung von Algorithmen**
- Beispiele

formale Darstellung von Algorithmen

Bei der Darstellung von Algorithmen liegt der Schwerpunkt vor allem auf der Darstellung der zeitlichen Ablauffolge, dem sogenannten **Steuerfluss** und nicht auf der Darstellung der einzelnen elementaren Anweisung an sich.

Theorem von Böhm und Jacopini

Mit nur drei logischen Elementarstrukturen

- **Sequenz**
- **Selektion**
- **Zyklus**

lässt sich jedes algorithmierbare Problem darstellen.

formale Darstellung von Algorithmen

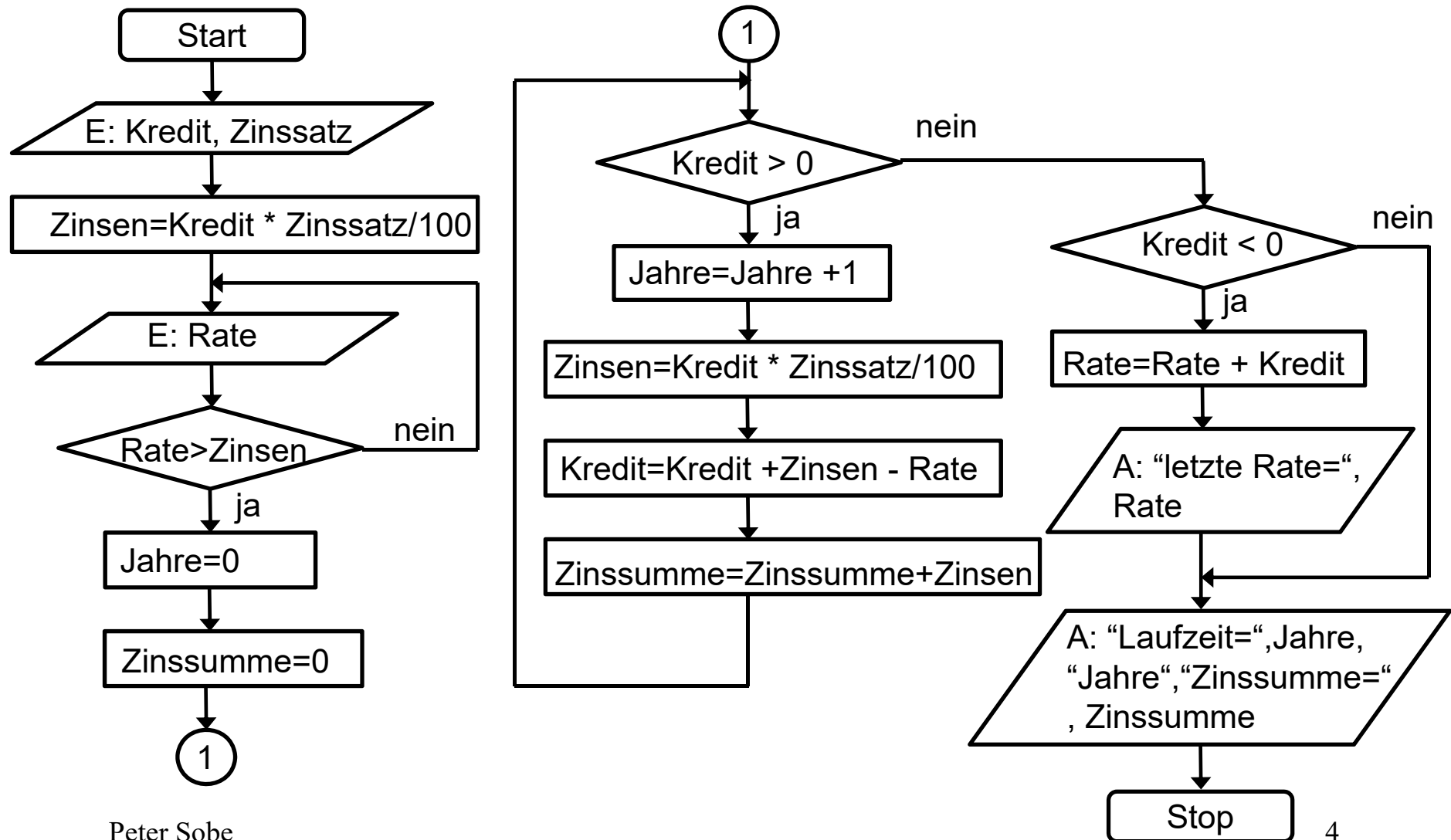
Es wurden zwei verschiedene Darstellungsformen entwickelt, die hauptsächlich in Gebrauch sind und auf grafische Symbole zurückgreifen:

- **Programmablaufplan (abk. PAP)**
nach DIN 66001 (vgl. Bild 1)
- **Struktogramm** nach Nassi und Shneiderman
nach DIN 66261 (vgl. Bild 2)

Beide repräsentieren eine sogenannte **Syntax**, die die Ausdrucksformen der Beschreibung formal spezifiziert.

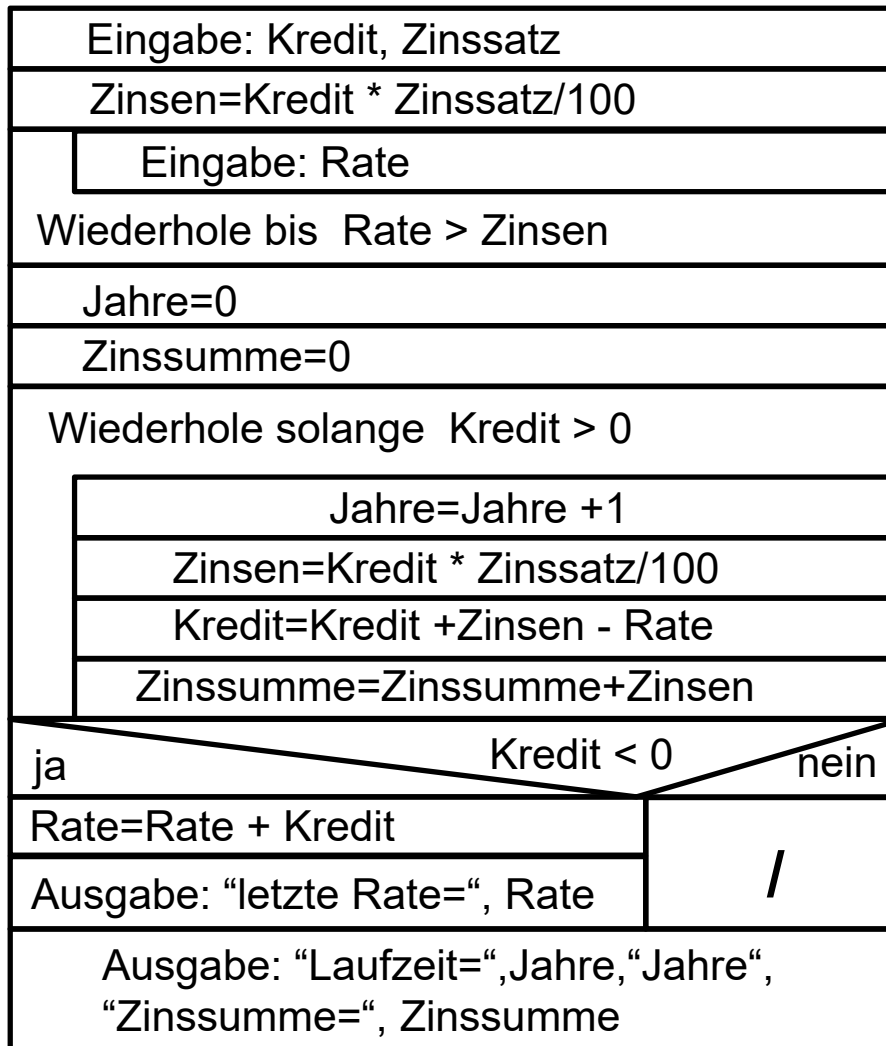
Die Bedeutung der Darstellung wird als **Semantik** bezeichnet.

Ein Algorithmus als Programmablaufplan



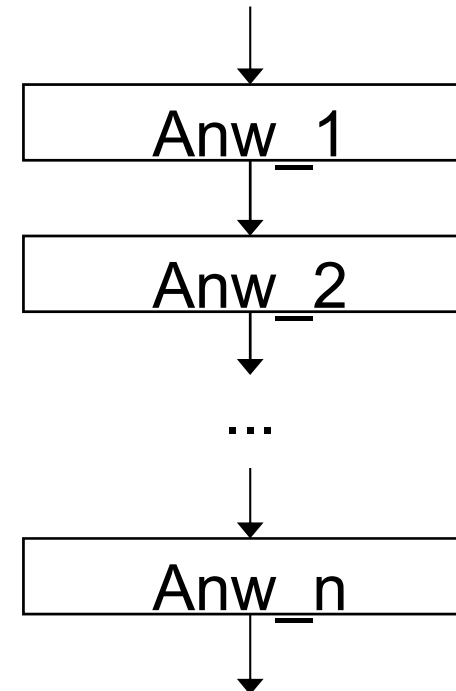
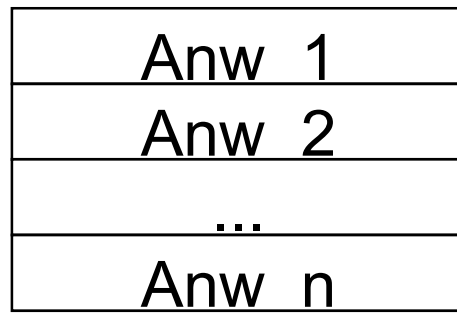
Algorithmus als Struktogramm

Struktogramm



Sequenz

Struktogramm Programmablaufplan

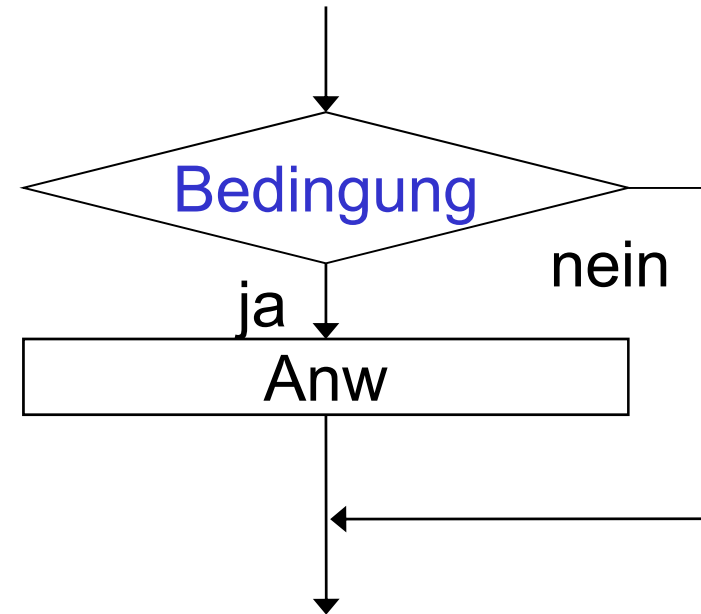
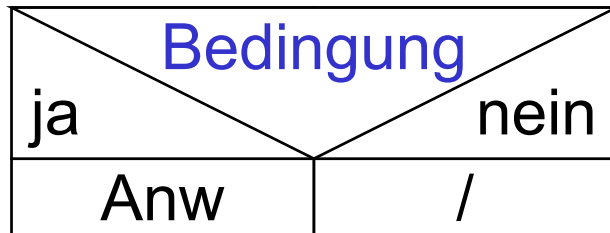


Das Wesentliche einer Sequenz besteht darin, dass sie nach außen wiederum nur eine einzige Aktion/Anweisung repräsentiert, obwohl sie intern aus einer Folge von Anweisungen (Anw) besteht.

einseitige Selektion

Struktogramm

Programmablaufplan

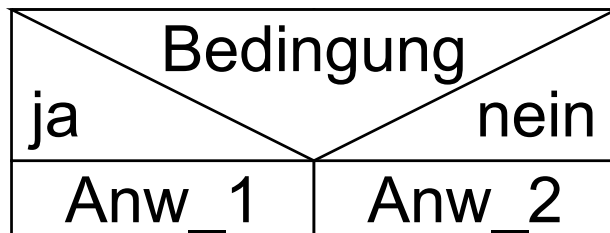


Die einseitige Selektion gestattet die Ausführung einer Anweisung *Anw* (welche wiederum eine Sequenz sein kann), wenn die angegebene *Bedingung* wahr (ja) ist.

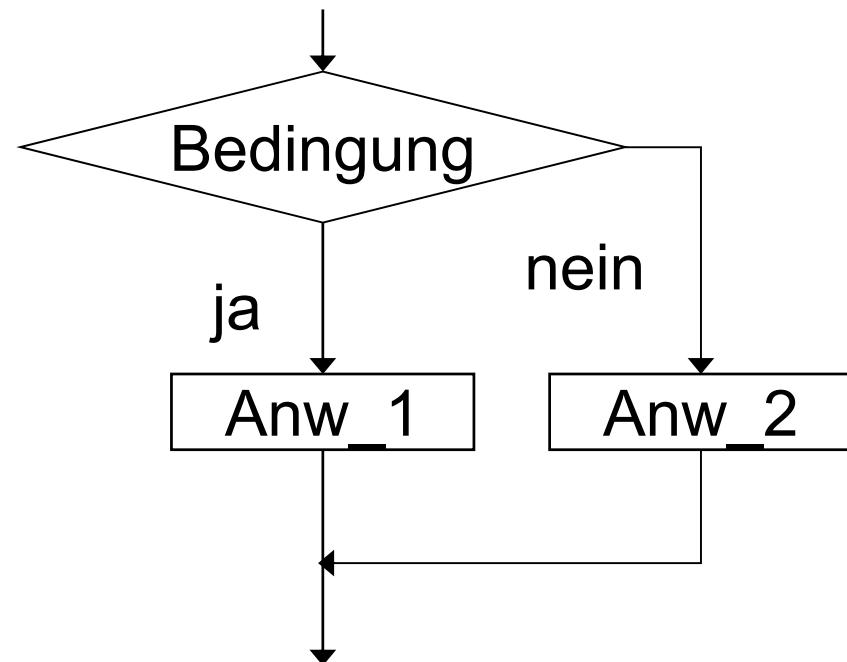
Die Bedingung ist ein **aussagenlogischer Ausdruck**.

zweiseitige Selektion

Struktogramm



Programmablaufplan



Die zweiseitige Selektion gestattet die Auswahl einer Anweisung (welche wiederum eine Sequenz sein kann) aus zwei angegebenen Sequenzen *Anw_1* bzw. *Anw_2*.

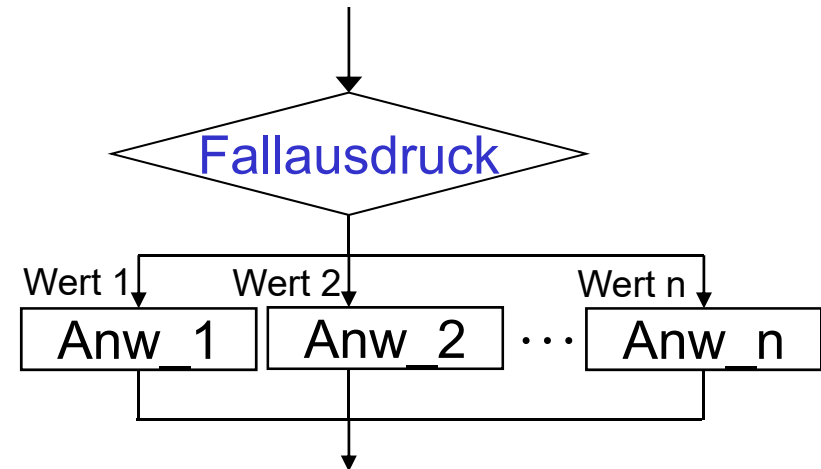
Wenn die angegebene **Bedingung** wahr (ja) ist, wird *Anw_1* ausgeführt, sonst *Anw_2*.

mehrfache Selektion ohne Sonst- Zweig

Struktogramm

Programmablaufplan

Fallausdruck			
Wert 1	Wert 2	...	Wert n
Anw_1	Anw_2	...	Anw_n

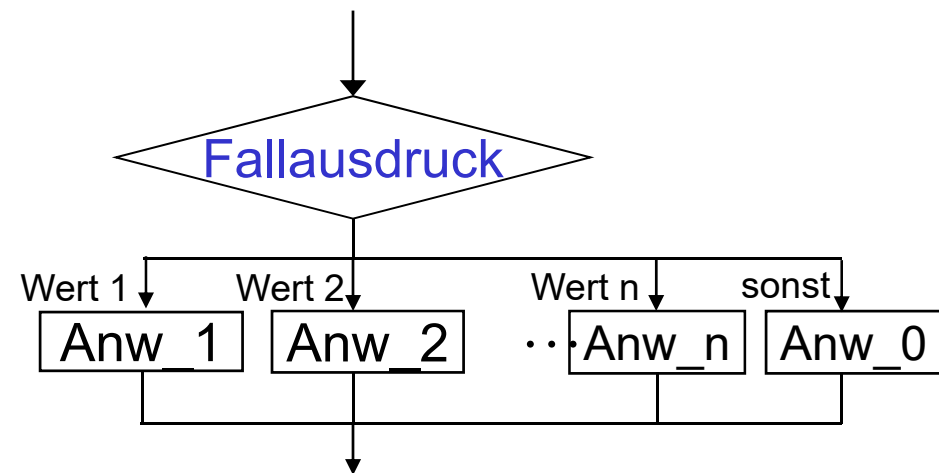


Der Fallausdruck kann auch einfache Variable repräsentieren, z.B. für Zahlenwerte, ein Zeichen oder eine Zeichenkette. Für alle zu behandelnden Fälle wird eine entsprechende Konstante als *Wert 1*, *Wert 2* usw. angegeben. Entsprechend dem zutreffenden Wert wird eine angegebene Anweisung ausgeführt, also *Anw_1* oder *Anw_2* usw.

mehrfache Selektion mit Sonst-Zweig

Struktogramm Programmablaufplan

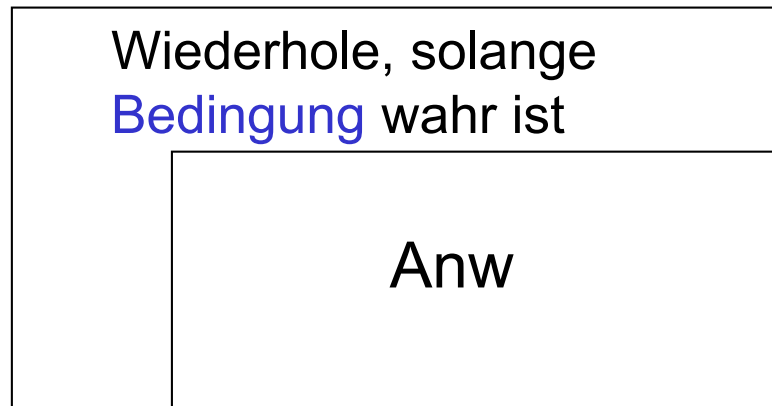
Fallausdruck				
Wert 1	Wert 2	...		sonst
Anw_1	Anw_2	...	Anw_n	Anw_0



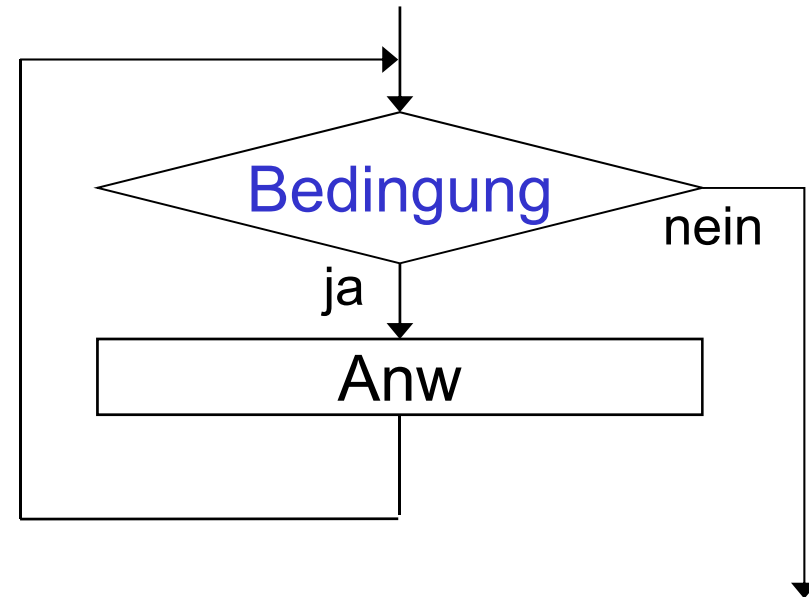
Der Ablauf entspricht im Prinzip der mehrfachen Selektion ohne Sonst-Zweig mit der Erweiterung, dass wenn der Fallausdruck keinen der angegebenen Werte *Wert 1*, *Wert 2*, ... ergibt, wird dann die Anweisung *Anw_0* ausgeführt.

Zyklus abweisend

Struktogramm



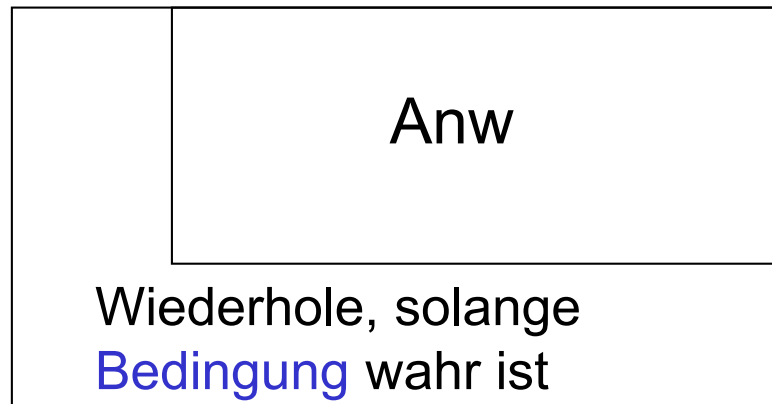
Programmablaufplan



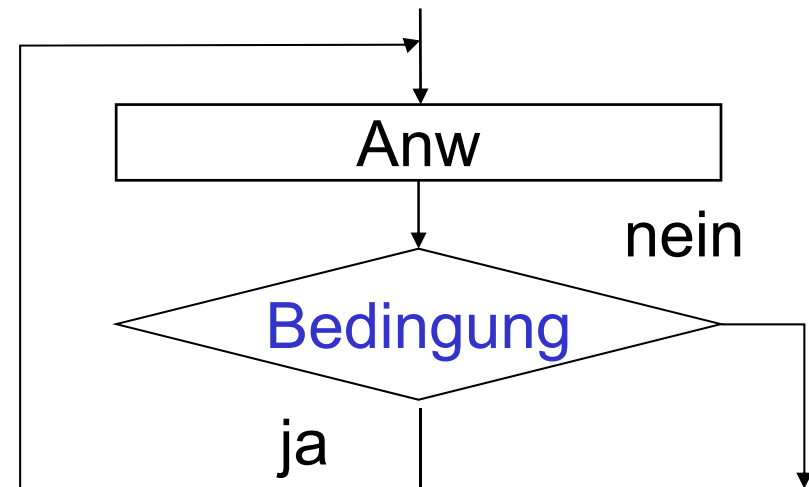
Die Ausführung einer Anweisung *Anw* (welche wiederum eine Sequenz sein kann), wird solange wiederholt ausgeführt, solange die angegebene **Bedingung** wahr (ja) ist. Abweisend bedeutet, dass wenn die Bedingung bei Eintritt bereits falsch ist, wird *Anw* überhaupt nicht ausgeführt.

Zyklus nicht abweisend

Struktogramm



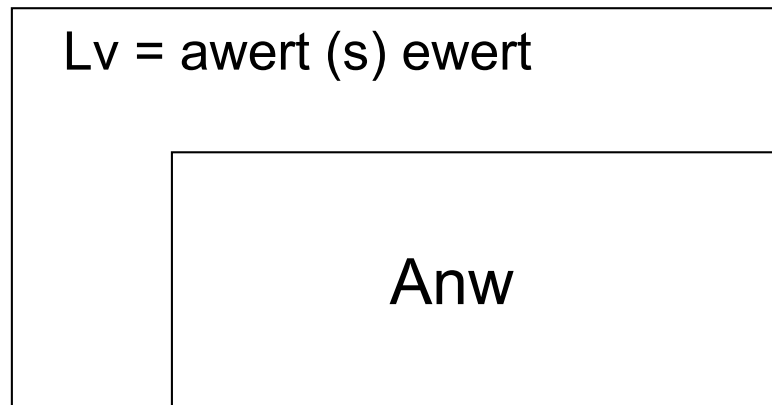
Programmablaufplan



Die Ausführung des Zyklus beginnt mit der Ausführung der Anweisung Anw (welche wiederum eine Sequenz sein kann), ohne Prüfung der Bedingung (deshalb „**nicht** abweisend“). Erst nach der Ausführung von *Anw* wird die angegebene **Bedingung** geprüft. Ist die Bedingung wahr, wird die Anweisung *Anw* wieder ausgeführt. Das wird wiederholt, solange die angegebene **Bedingung** wahr (ja) ist.

Zähl-Zyklus

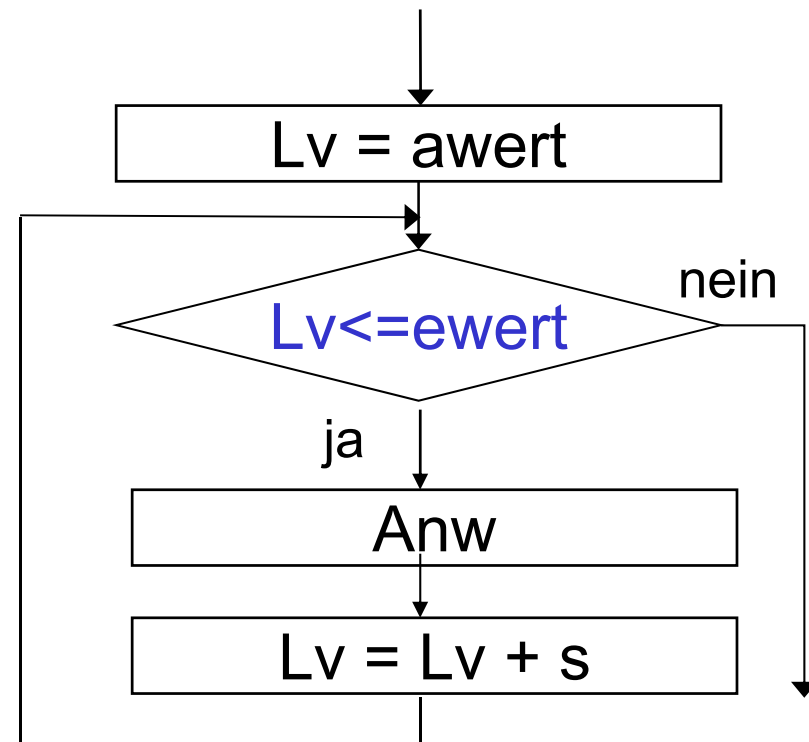
Struktogramm



Lv die ist sogenannte Laufvariable

- **awert** ist der Anfangswert von Lv
- **s** ist die Schrittweite
- **ewert** ist der Endwert von Lv

Programmablaufplan

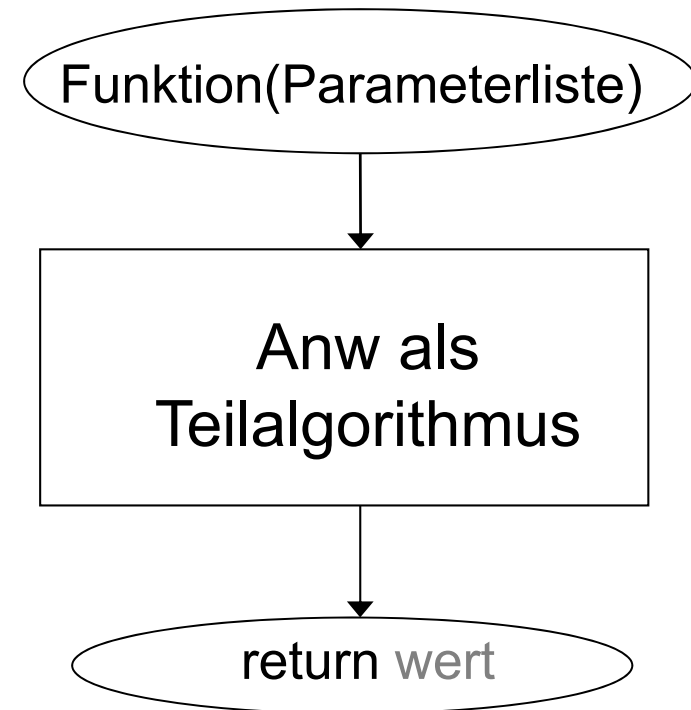
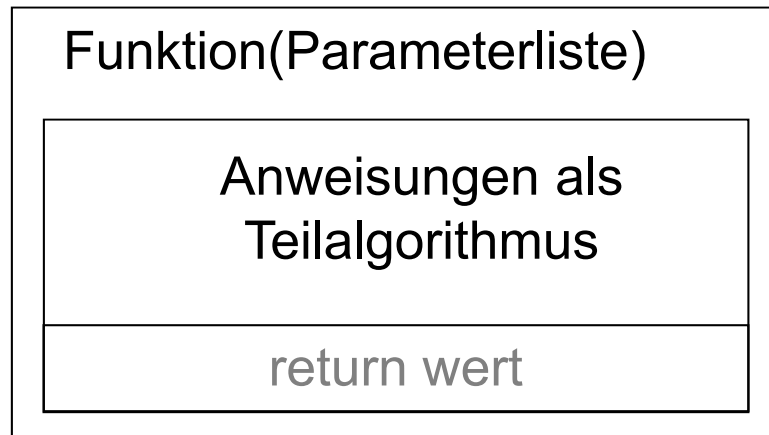


Die Anweisung *Anw* erfährt eine bestimmte Anzahl von Wiederholungen, die sich aus den Größen *awert*, *s* und *ewert* ergibt.

Unterprogramme

Struktogramm

Programmablaufplan



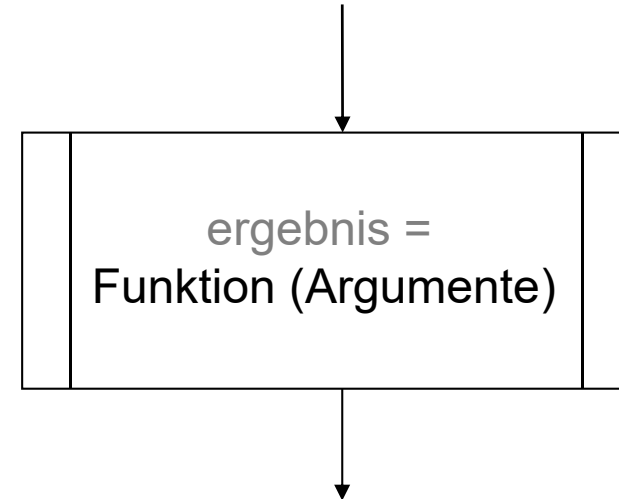
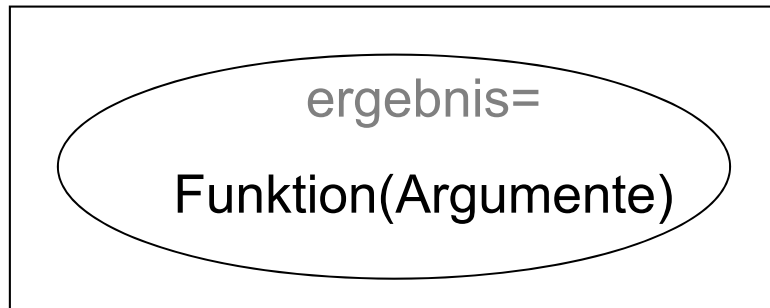
Funktion ist der Name eines Teilalgorithmus. Alle Größen, die der Algorithmus aus der Umgebung zur Abarbeitung benötigt, müssen in der Parameterliste definiert werden.

In der Parameterliste müssen Eingangs- und Ausgangsgrößen gekennzeichnet werden.

Unterprogramm-Aufruf

Struktogramm

Programmablaufplan

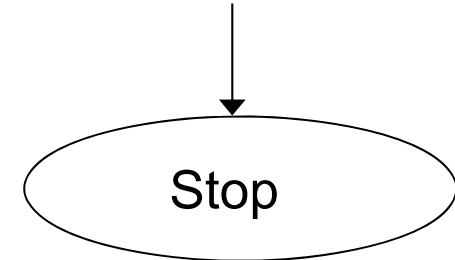
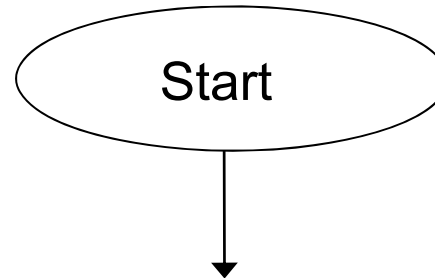


Die Argumente sind die in einer Umgebung gültigen Variablen und Konstanten, die für die Parameter beim Aufruf eingesetzt werden.

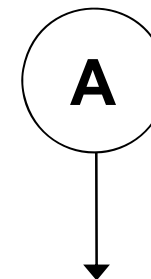
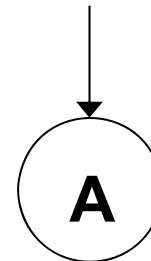
Beispiele für Unterprogramme und deren Aufruf werden an der Tafel demonstriert.

Hilfsgrößen bei Programmablaufplänen

Anfang / Ende
eines Hauptprogrammes

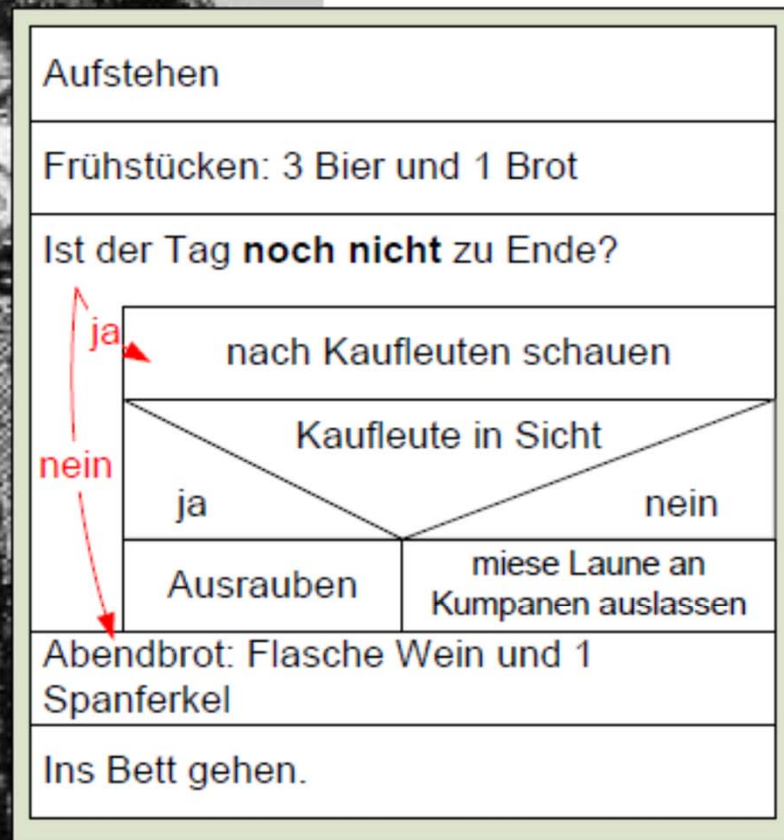
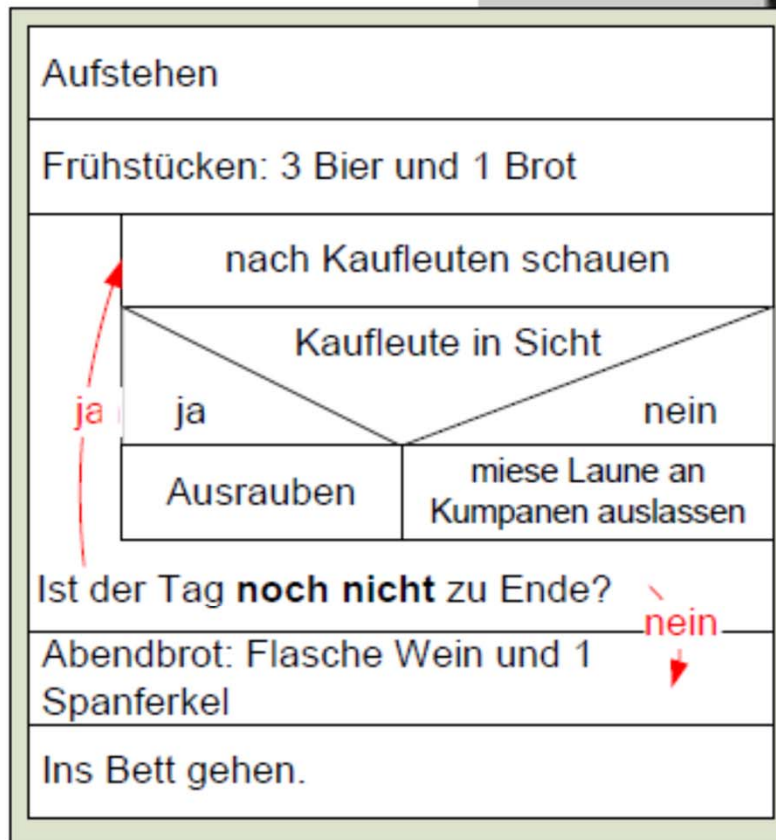


Anschlußstelle (Konnektor)
zwischen zwei
Programmablaufplanteilen



Ein erstes Beispiel

Tagesablauf von Lips Tullian:



Hintergrund

wahrer Name:

evtl. Elias Erasmus Schönknecht
oder Philip Mengstein

Räuberhauptmann:

Anführer der Räuberbande „Schwarze Garde“

zeitweiser „Wirkungskreis“: bis Thüringen, Harz, Lausitz,
Osterzgebirge, Prag



u.a. Unterschlupf im Tharandter Wald

geköpft 08. 03. 1715 in Alt-Dresden vor dem Schwarzen Tor im Beisein
von August dem Starken und etwa 20.000 „Schaulustigen“

Quellen:

http://www.sachsen-reisefuehrer.com/dresdner_umland/lips_tullian.htm

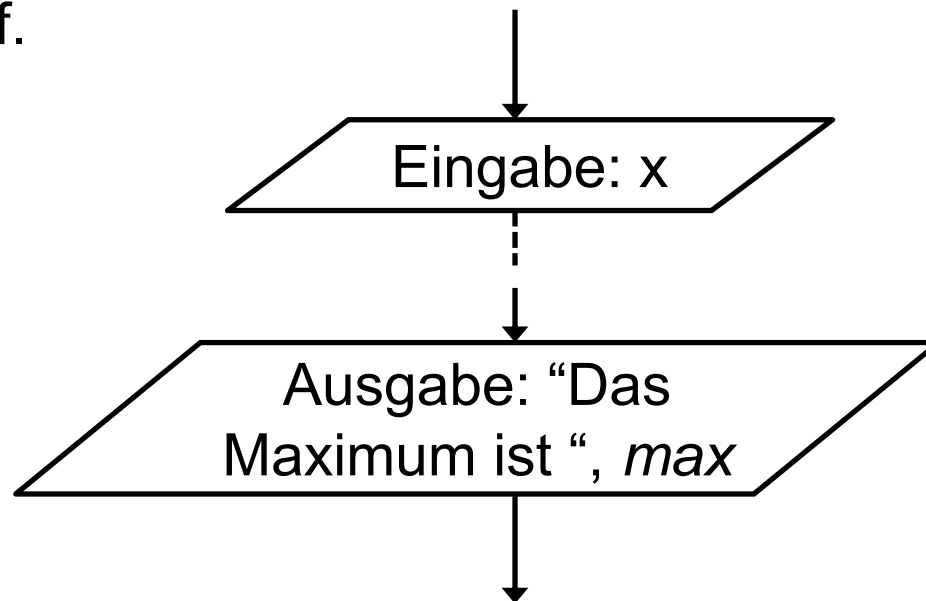
http://de.wikipedia.org/wiki/Lips_Tullian

Anweisungen und Bedingungen für PAP und Struktogramme (1)

Anweisungen:

- Eingabeanweisungen, z.B. Eingabe: x
- Ausgabeanweisungen, z.B. Ausgabe: "Das Maximum ist ", max

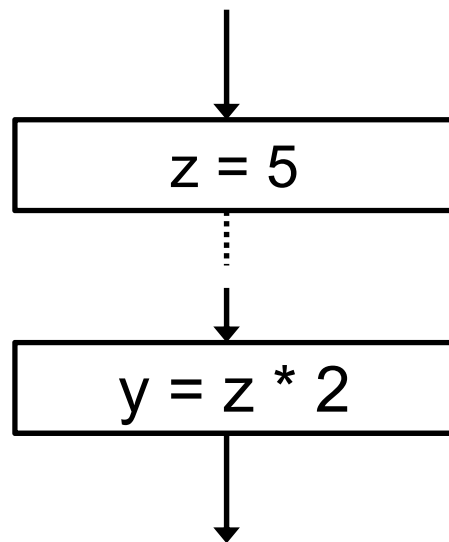
Die Symbole x und max werden hier für Variablen benutzt. Variablen nehmen Daten während der Abarbeitung des Programms auf.



Anweisungen und Bedingungen für PAP und Struktogramme (2)

Anweisungen:

- Zuweisungen konstanter Werte, z.B. $z = 5$
- Zuweisungen im Stil mathematischer Ausdrücke, z.B. $y = z * 2$. Hier wird der Variable y der Wert zugewiesen, der sich aus $z*2$ errechnet. Allgemein wird der links von $=$ stehenden Variablen der Wert des Ausdrucks rechts zugewiesen.



Nicht gültig:

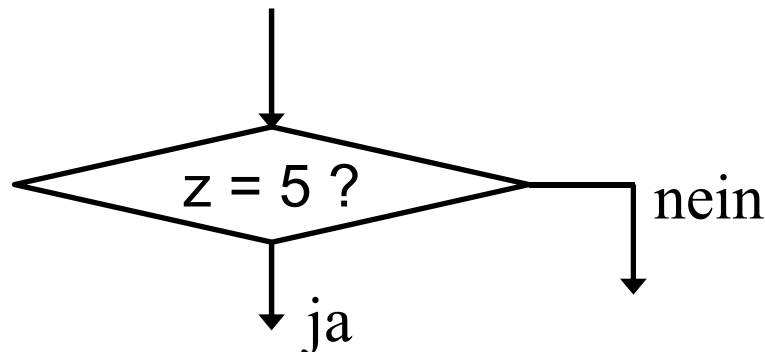
$$x+y = 10*z$$

Das ist keine Zuweisung, obwohl es ein möglicher mathematischer Ausdruck ist.

Anweisungen und Bedingungen für PAP und Struktogramme (3)

Bedingungen:

- Als Frage formuliert, ob eine Variable einen bestimmten Wert aufweist, z.B. $z = 0 ?$. Die Bedingung ist dann erfüllt, wenn z den Wert 0 besitzt. Das „?“ kann (muss nicht) verwendet werden.
- Als Gleichung formuliert, z.B. $y = z * z ?$. Wenn die linke und rechte Seite der Gleichung den gleichen Wert ergeben, dann ist die Bedingung erfüllt.
- Als Ungleichung formuliert, z.B. $y > z ?$. Nur bei ungleicher linker und rechter Seite ist die Bedingung erfüllt.



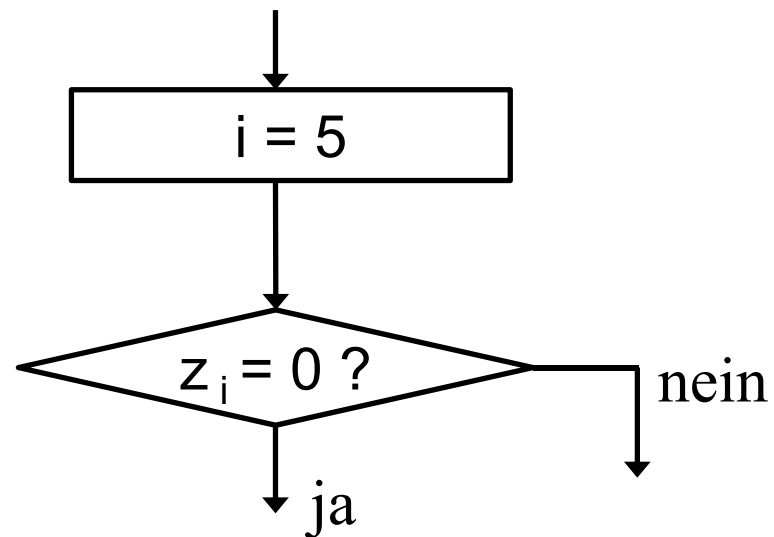
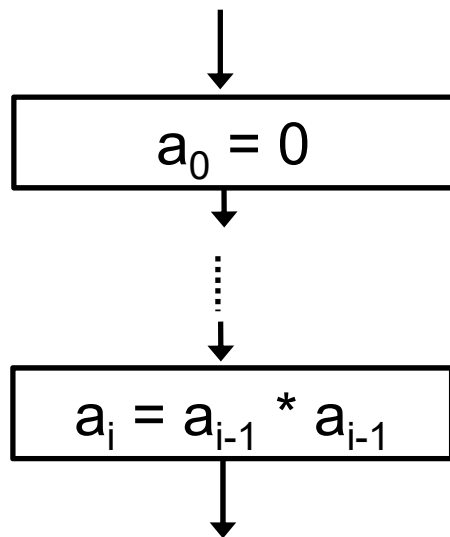
Achtung: Ein Ausdruck wirkt unterschiedlich, je nach dem, ob er in einer Anweisung oder in einer Bedingung verwendet wird.

Anweisungen und Bedingungen für PAP und Struktogramme (4)

Felder:

Algorithmen arbeiten oft auf Feldern, die N einzelne Variable als Folge enthalten, z.B. $z_0, z_1, z_2, \dots, z_{N-1}$

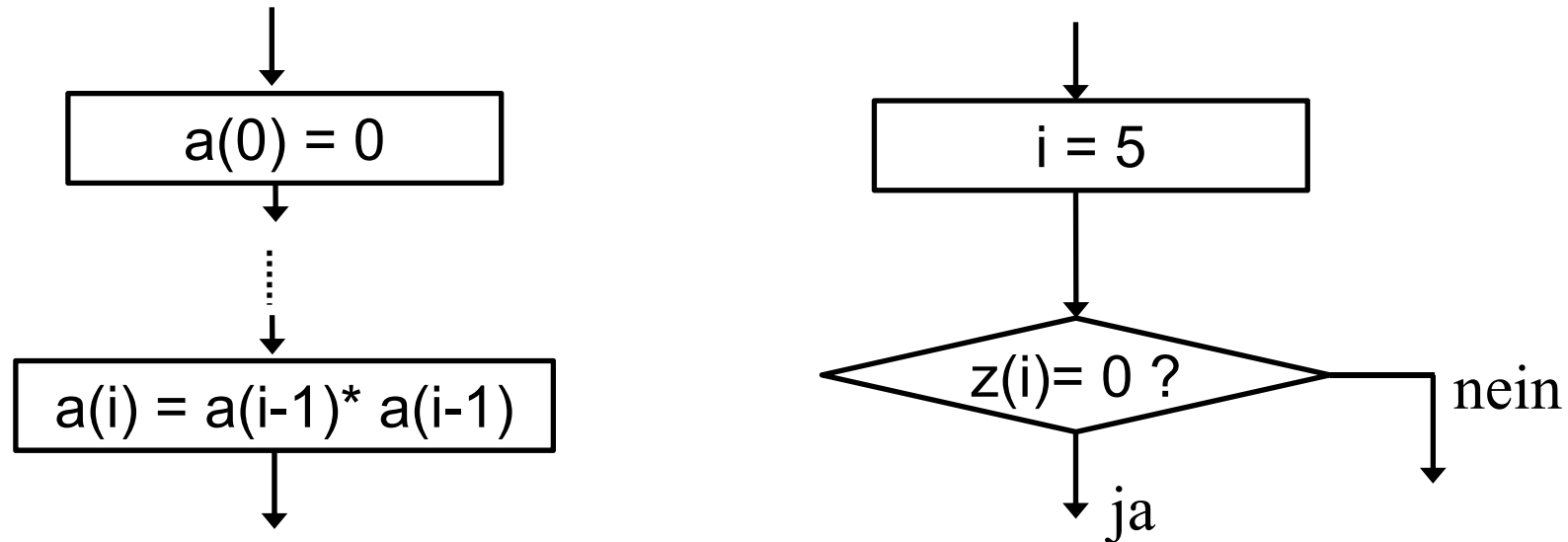
Die einzelnen Elemente können im PAP oder Struktogramm mit Index angegeben werden. Der Index kann ein konkreter Wert sein (z.B. z_0) oder auch als Variable angegeben werden (z.B. z_i).



Anweisungen und Bedingungen für PAP und Struktogramme (5)

Felder (Fortsetzung):

Der Index kann auch mit Indexklammern geschrieben werden, z.B. $z[0]$, $z[i-1]$ oder auch $z(0)$, $z(i-1)$



Die Programmierung von Feldzugriffen in Visual Basic erfolgt mit runden Indexklammern.

Zusammenfassung: PAP und Struktogramme

- Algorithmen werden als Abläufe, bestehend aus einzelnen Anweisungen notiert.
- PAP und Struktogramme dienen zur programmiersprachen-unabhängigen Beschreibung von Algorithmen
 - während Entwurf
 - zur Dokumentation
- Abläufe unterliegen drei prinzipiellen Ablaufweisen (s.g. Steuerflusskonstrukte)
 - Sequenz (Nacheinanderausführung)
 - Selektion
 - Zyklus
- Wiederholt benutzte Teilalgorithmen werden als Unterprogramme geschrieben und dann an mehreren Stellen aufgerufen