

## Grundelemente von Algorithmen

**Operatoren** (geordnet nach fallender Priorität **P**)

P	Beschreibung	Operator
1	Klammern	$(x+y) * z$
2	Potenz	$x^y$
3	Vorzeichen: Plus, Minus	$+x$ $-x$
	Negation: logisch	$\neg x$ $!x$ $\sim x$
4	Multiplikation, Division	$x * y$ $x / y$
5	Addition, Subtraktion	$x + y$ $x - y$
6	größer-als, kleiner-als	$x > y$ $x < y$
	größer-gleich, kleiner-gleich	$x \geq y$ $x \leq y$ $x \geq y$ $x \leq y$
7	Gleichheit	$x = y$ $x == y$
	Ungleichheit	$x \neq y$ $x != y$
8	Konjunktion, logisches UND	$m \wedge n$ $m \& \& n$
9	Disjunktion, logisches ODER	$m \vee n$ $m    n$

### Arithmetischer Ausdruck

- Rechenvorschrift für arithmetische Werte
- Verknüpfung von Zahlen, Variablen, Konstanten, Funktionen mittels arithmetischer Operatoren
- abweichende Darstellung zu Termen der Mathematik:
  - Komma durch Punkt ersetzen bei rationalen/reellen Zahlen,
  - Multiplikationsoperator  $*$  immer schreiben,
  - einzeilige Schreibweise,
  - nur  $()$  zulässig für Änderung der Prioritätenreihenfolge

### Logischer Ausdruck

- Vergleich oder Verknüpfung von Vergleichen oder Wahrheitswerten mittels logischer Operatoren
- Wertebereich: zweiwertig  $\rightarrow$  wahr/falsch, true/false, ja/nein, 1/0

### Ergibtanweisung

- neuer Wert der Variablen ergibt sich aus dem Wert des Ausdrucks

<b>Variable = Ausdruck</b>	<b>Variable := Ausdruck</b>
----------------------------	-----------------------------

### Laufbereich

- zuerst Zuweisung des Anfangswertes  $A$  an die Laufvariable  $L$
- dann schrittweise Erhöhung von  $L$  um Schrittweite  $S$ , solange Endwert  $E$  nicht über- ( $S > 0$ ) bzw. nicht unterschritten ( $S < 0$ ) wird

<b><math>L = A(S)E</math></b>	<b><math>L := A(S)E</math></b>
-------------------------------	--------------------------------

### Eingabeanweisung

- Wertzuweisung an eine Variable durch Eingabe

<b>EINGABE: Variable</b>
--------------------------

### Ausgabeanweisung

- Ausgabe des Wertes einer Variablen oder eines Textes

<b>AUSGABE: "Text", Variable</b>
----------------------------------

## Darstellung von Algorithmen

**Theorem von Böhm und Jacopini:** Mit nur drei logischen Elementarstrukturen **Sequenz**, **Selektion**, **Zyklus** läßt sich jedes algorithmierbare Problem darstellen. D.h., jeder beliebige Algorithmus läßt sich ohne unbedingte Sprunganweisungen formulieren.

## Struktogramm (Nassi-Shneiderman-Diagramm)

- Darstellung eines Algorithmus in Form eines geschlossenen Blocks  $\rightarrow$  im Ganzen rechteckig, d.h., so breit wie der breiteste Strukturblock
- stellt eine Folge von Blöcken dar, wobei jeder Block eine abgeschlossene funktionelle Bedeutung besitzt.

### → Eigenschaften der Blöcke:

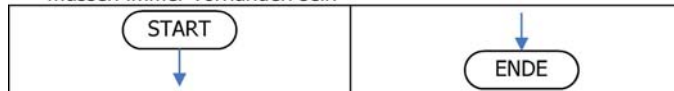
- je Block nur 1 Ein- und 1 Ausgang,
- Steuerfluß von oben nach unten,
- am Eingang Datenaustausch mit Vorgängerblock,
- am Ausgang Datenaustausch mit Nachfolgerblock,
- Aneinanderreihen und Schachteln von Blöcken ergibt neuen Block.

## Programmablaufplan (PAP), Flußdiagramm

- Darstellung eines Algorithmus in Form mit gerichteten Pfeilen verbundener Teilelemente

### → Grenzstelle(n)

- kennzeichnen Anfang/Ende eines Programmablaufplans
- müssen immer vorhanden sein



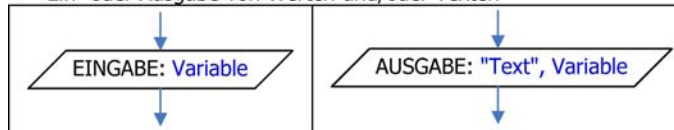
### → Anschlußstelle (Verbinder, Konnektor)

- verbindet zwei Programmablaufplanteile



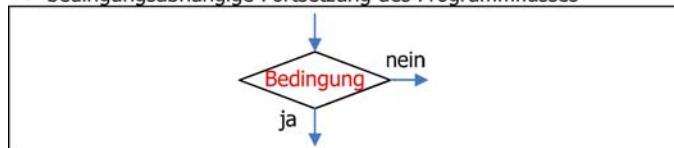
### → Ein-/Ausgabe

- Ein- oder Ausgabe von Werten und/oder Texten



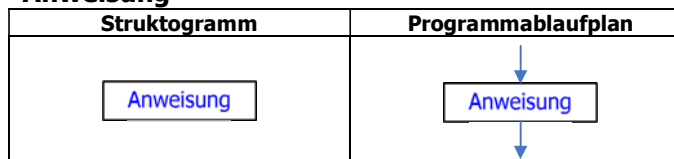
### → Verzweigung

- bedingungsabhängige Fortsetzung des Programmflusses



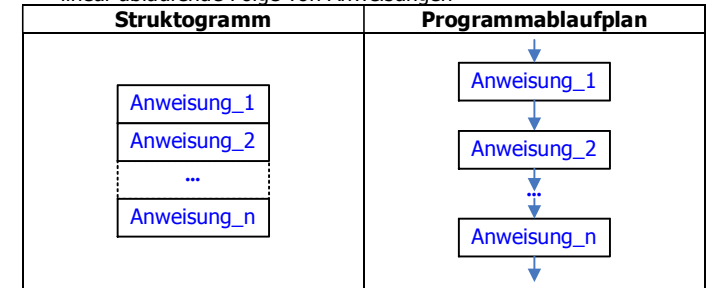
## Bausteine von Algorithmen

### Anweisung



## Sequenz

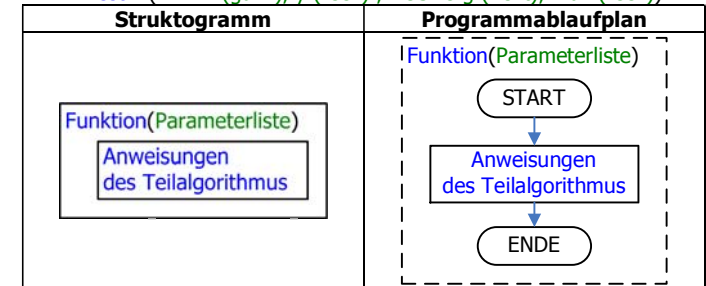
- linear ablaufende Folge von Anweisungen



## Prozedur

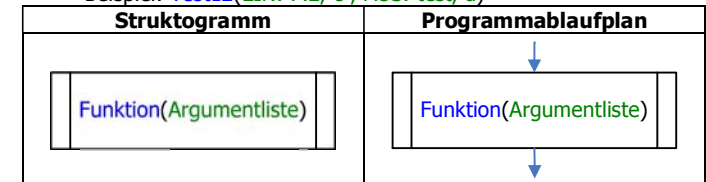
### → Deklaration

- Funktion:** Name der Funktion/Prozedur
- Parameterliste:**
  - alle Größen, die der Algorithmus aus der Umgebung zur Abarbeitung benötigt
  - Eingangs- und Ausgangsgrößen müssen unterschieden werden.
  - Beispiel: **Test12(EIN: x (ganz), y (reell) ; AUS: erg (Text), max (reell))**



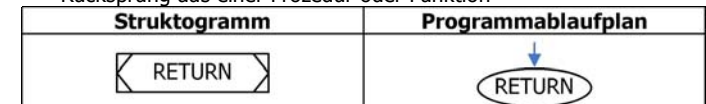
### → Aufruf

- Aufruf eines Unterprogramms bzw. einer Prozedur oder Funktion
- Nach Abarbeitung wird zur aufrufenden Stelle zurückgesprungen.
- Argumentliste:**
  - Argumente:** Die in einer Umgebung gültigen Objekte, die für die Parameter beim Aufruf eingesetzt werden.
  - Eingangs- und Ausgangsgrößen müssen unterschieden werden.
  - Beispiel: **Test12(EIN: 7.2, c ; AUS: test, d)**



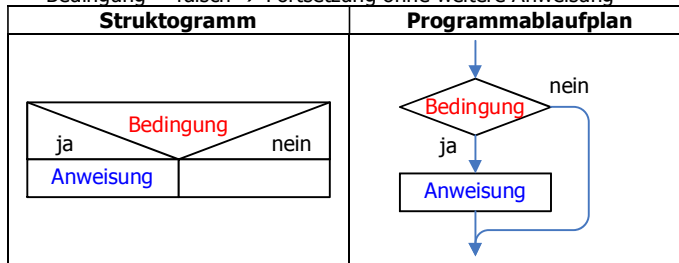
### → Rücksprung

- Rücksprung aus einer Prozedur oder Funktion

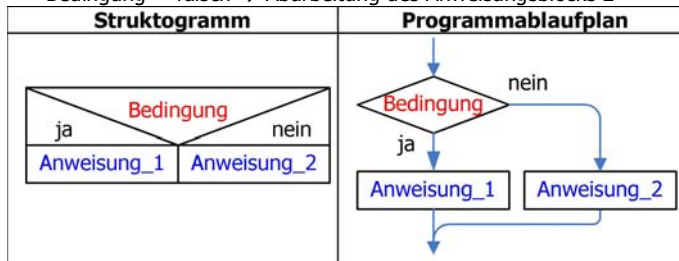


**Selektion (Verzweigung, Alternative, Auswahl)****→ einseitig (einfache Auswahl, bedingte Verarbeitung)**

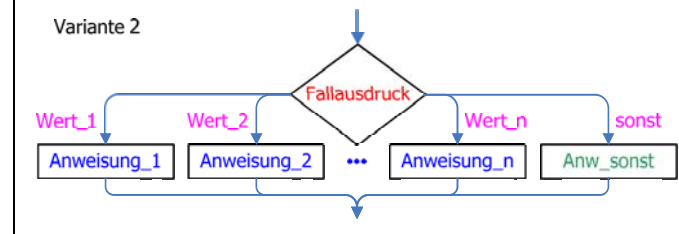
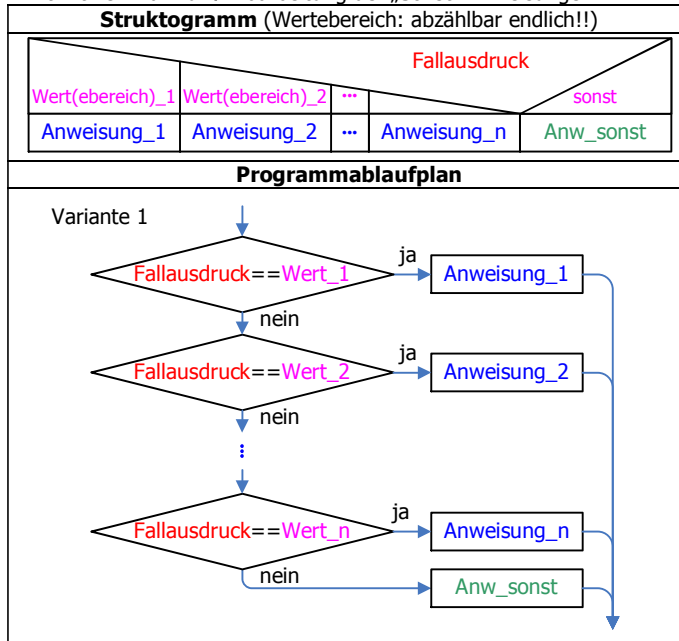
- Bedingung==wahr → Abarbeitung des Anweisungsblocks
- Bedingung==falsch → Fortsetzung ohne weitere Anweisung

**→ zweiseitig (zweifache Auswahl, alternative Verarbeitung)**

- Bedingung==wahr → Abarbeitung des Anweisungsblocks 1
- Bedingung==falsch → Abarbeitung des Anweisungsblocks 2

**→ mehrfach (Fallauswahl)**

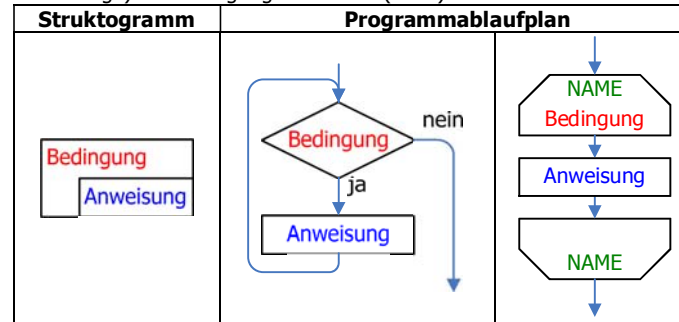
- Verzweigung in Abhängigkeit des integralen Fallausdruck-Wertes
- trifft kein Fall zu → Abarbeitung der „Sonst“-Anweisungen

**Zyklus (Wiederholung, Iteration, Schleife)**

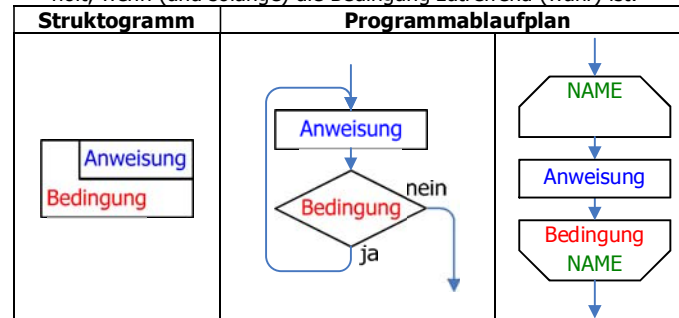
- NAME ist der individuelle Name des jeweiligen Zyklus.

**→ abweisend (vorprüfend, kopfgesteuert)**

- Anweisung wird nur dann ausgeführt (und wiederholt), wenn (und solange) die Bedingung zutreffend (wahr) ist.

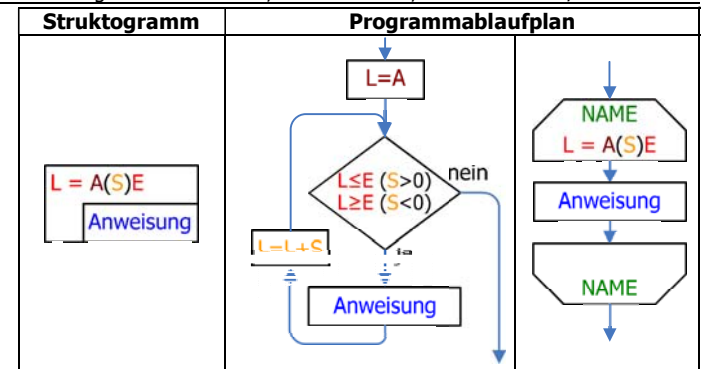
**→ nicht abweisend (nachprüfend, fußgesteuert)**

- Anweisung wird mindestens einmal ausgeführt und danach wiederholt, wenn (und solange) die Bedingung zutreffend (wahr) ist.

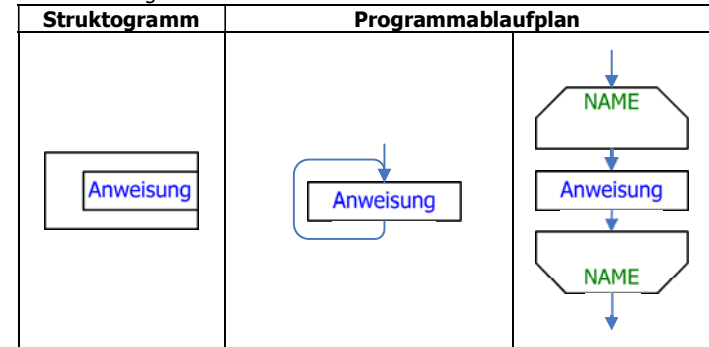
**→ Zähl-Zyklus (zählergesteuerte Schleife)**

- Die Anweisung erfährt eine bestimmte Anzahl von Wiederholungen, die sich aus den Größen **A**, **S** und **E** ergibt.

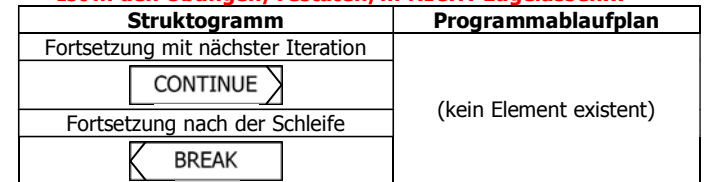
  1. Zuweisung des Anfangswertes **A** an die Laufvariable **L**
  2. Test, ob Endwert **E** über- (**S**>0) bzw. unterschritten (**S**<0) wird, falls ja → Fortsetzung nach der Schleife
  3. Abarbeitung der Anweisung
  4. Erhöhung von **L** um Schrittweite **S** → Schritt 2

**→ Endlos-~ (Wiederholung ohne Bedingungsprüfung)**

- Kann allenfalls durch einen Abbruch verlassen werden.
- Sollte möglichst vermieden werden!

**→ Abbruch**

- Abbruch der Schleifenabarbeitung
- Sollte möglichst vermieden werden!

**• Ist in den Übungen/Testaten/... NICHT zugelassen!!!****Notizen**