

# *Softwareentwicklung 1*

*Prof. Dr. Hanspeter Mössenböck*

<https://ssw.jku.at/Teaching/Lectures/SW1/VL/>



H. Mössenböck: *Sprechen Sie Java?*  
5. erweiterte Auflage,  
dpunkt.verlag, 2014

# 1. Einführung

## 1.1 Grundbegriffe

1.2 Algorithmen und Ablaufdiagramme

1.3 Struktogramme

1.4 Grammatiken



# Worum geht es?

## Programmieren

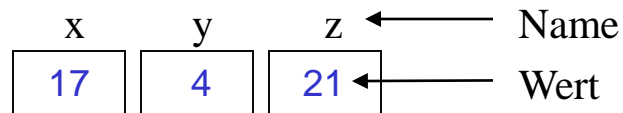
ein Problem so exakt beschreiben, dass es ein Computer lösen kann

- F kreative Tätigkeit
- F Ingenieurtätigkeit
- F Nur wenige Leute können gut programmieren

Programm = Daten + Befehle

# Daten und Befehle

**Daten** Menge adressierbarer Speicherzellen



Daten sind binär gespeichert (z.B. 17 = 10001)

Binärspeicherung ist universell (Zahlen, Texte, Bilder, Ton, ...)

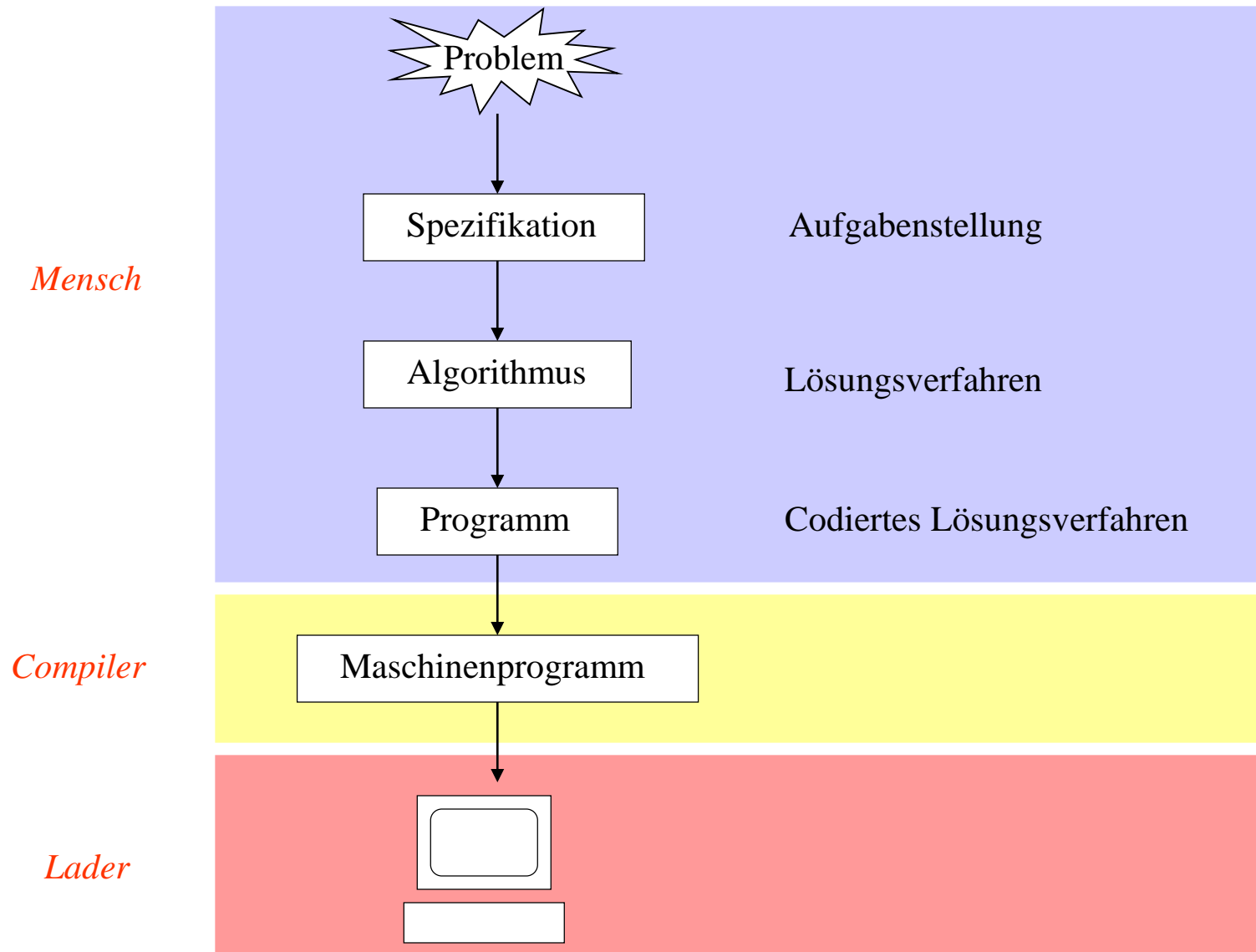
1 Byte = 8 Bit

1 Wort = 4 Byte

**Befehle** Operationen mit den Speicherzellen

<i>Maschinensprache</i>	<i>Hochsprache</i>
ACC ← x // Lade Zelle x	z = x + y;
ACC ← ACC + y // Addiere Zelle y	
z ← ACC // Speichere Ergebnis in Zelle z	

# Programmerstellung



# 1. Einführung

1.1 Grundbegriffe

1.2 Algorithmen und Ablaufdiagramme

1.3 Struktogramme

1.4 Grammatiken

# Algorithmus



Schrittweises, präzises Verfahren zur Lösung eines Problems

Name

Parameter

**Summiere Zahlen von 1 bis  $max$**  ( $\downarrow max$ ,  $\uparrow sum$ )

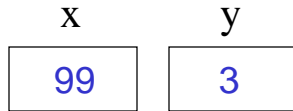
1.  $sum \leftarrow 0$
2.  $zahl \leftarrow 1$
3. Wiederhole, solange  $zahl \leq max$ 
  - 3.1  $sum \leftarrow sum + zahl$
  - 3.2  $zahl \leftarrow zahl + 1$

Folge von Schritten

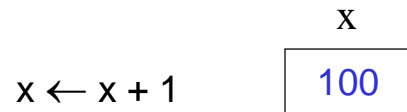
Programm = Beschreibung eines Algorithmus in einer Programmiersprache

# Variablen

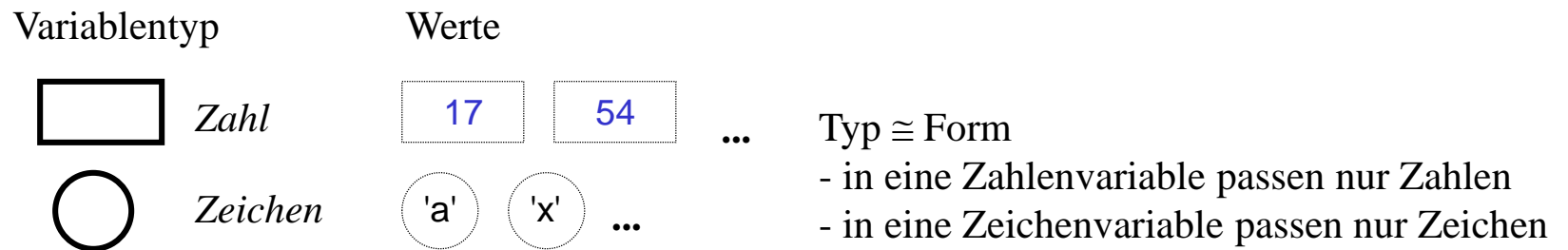
Sind benannte Behälter für Werte



Können ihren Wert ändern



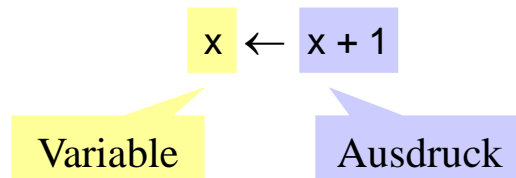
Haben einen Datentyp = Menge erlaubter Werte





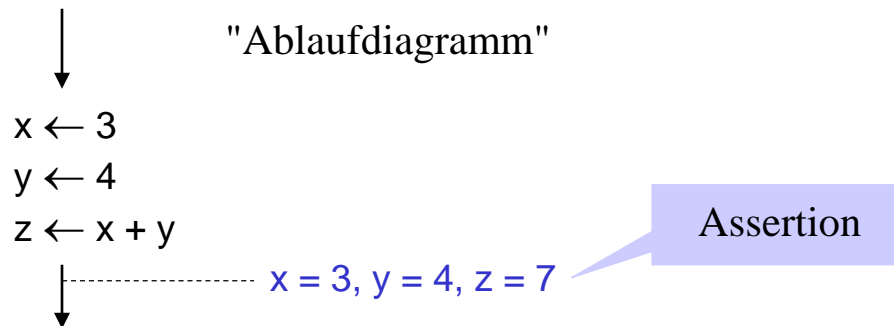
# Anweisungen

## Wertzuweisung



1. werte Ausdruck aus
2. weise seinen Wert der Variablen zu

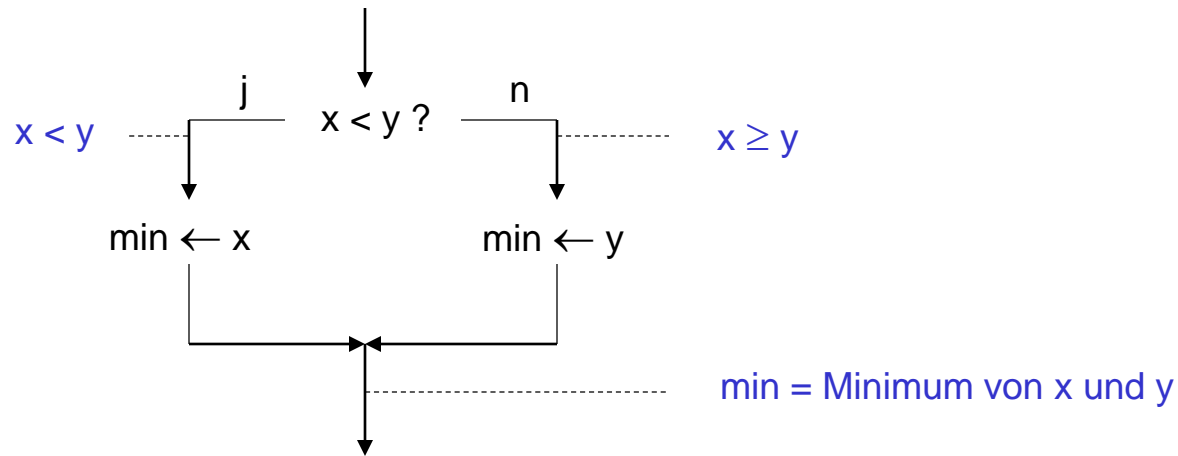
## Anweisungsfolge (auch Sequenz)



*Assertion (Zusicherung)*  
 Aussage über den Zustand des Algorithmus  
 an einer bestimmten Stelle

# Anweisungen

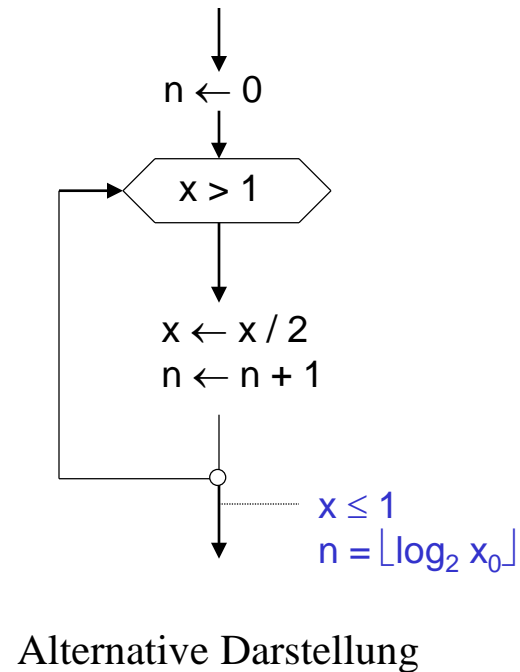
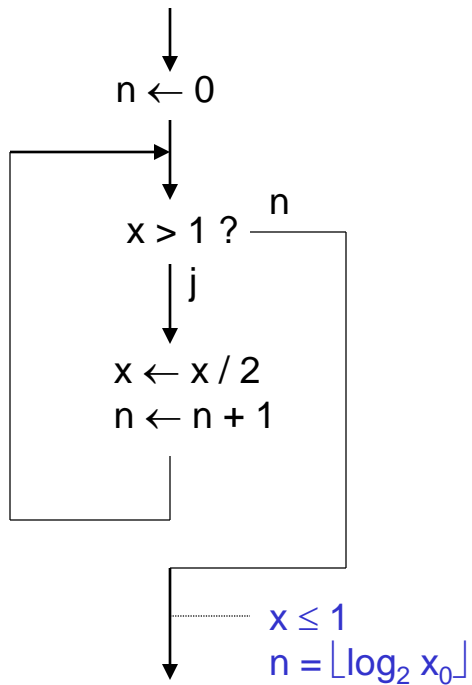
**Auswahl** (auch Verzweigung, Abfrage, Selektion)



# Anweisungen

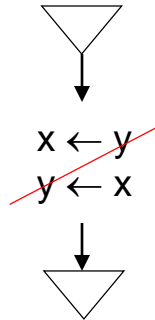


## Wiederholung (auch Schleife, Iteration)

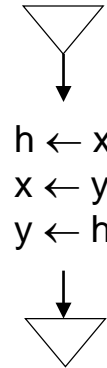


# Beispiel: Vertauschen zweier Variableninhalte

Swap ( $\downarrow\uparrow x, \downarrow\uparrow y$ )



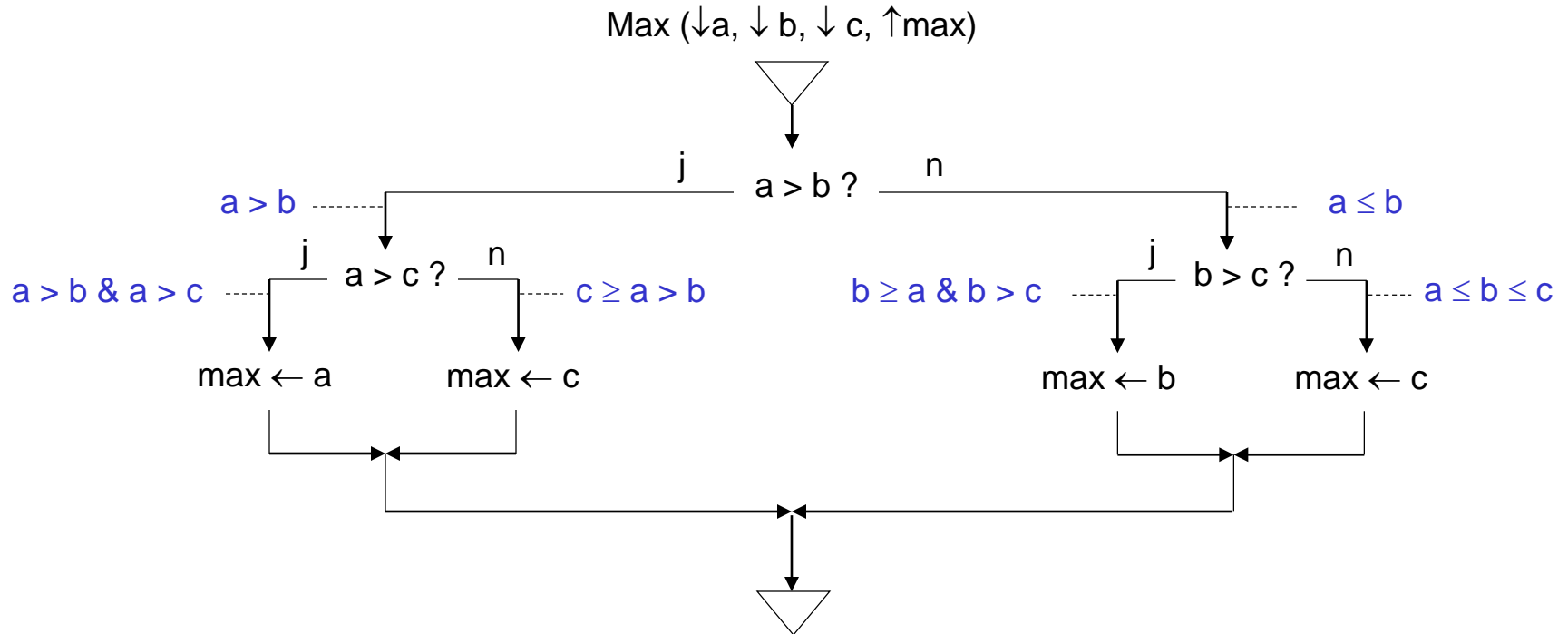
Swap ( $\downarrow\uparrow x, \downarrow\uparrow y$ )



Schreibtischttest

x	y	h
<del>3</del>	<del>2</del>	3
2	3	

# Beispiel: Maximum dreier Zahlen bestimmen

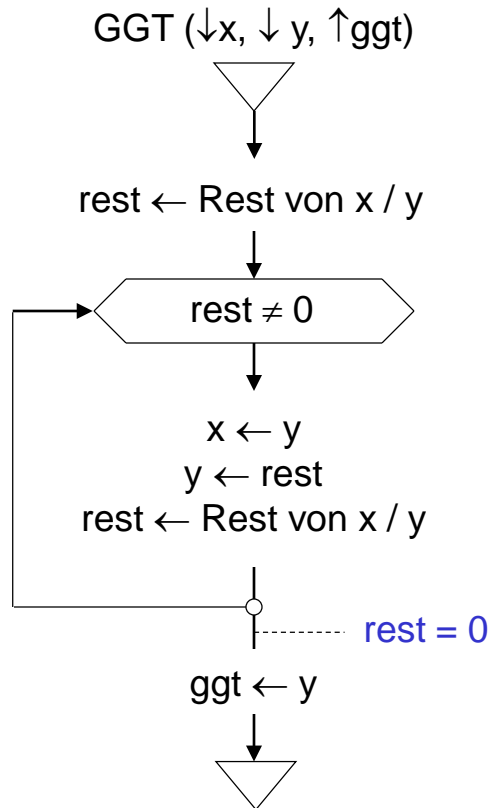


# Beispiel: Euklidischer Algorithmus

**Euklid**  
griechischer Mathematiker  
ca. 300 v. Chr.



Berechnet den größten gemeinsamen Teiler zweier Zahlen  $x$  und  $y$



## Schreibtischtest

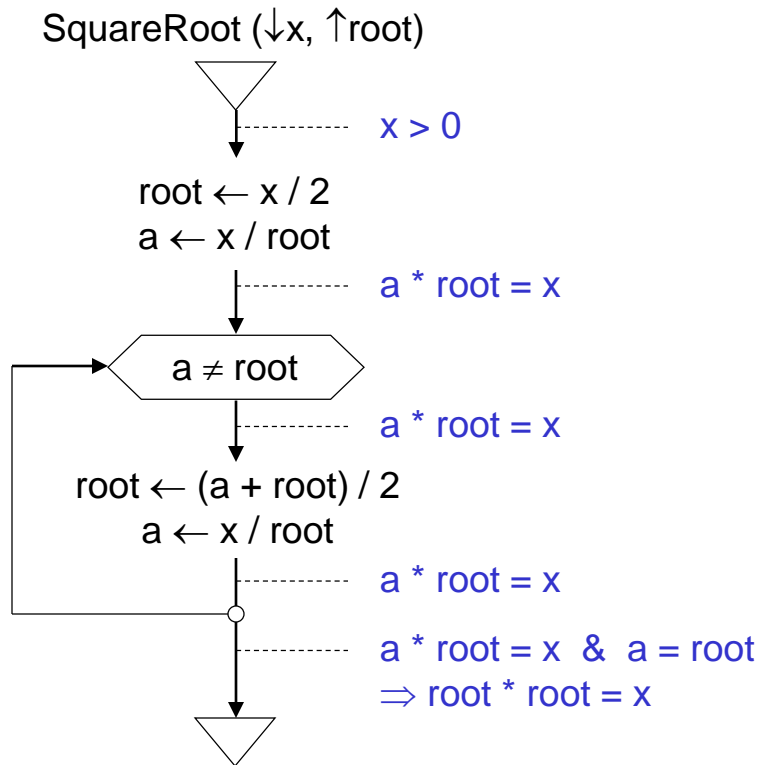
x	y	rest
28	20	8
20	8	4
8	4	0

Warum funktioniert dieser Algorithmus?  
(ggt teilt  $x$ ) & (ggt teilt  $y$ )  
 $\Rightarrow x = i \cdot \text{ggt}, y = j \cdot \text{ggt}, (x-y) = (i-j) \cdot \text{ggt}$   
 $\Rightarrow$  ggt teilt  $(x - y)$   
 $\Rightarrow$  ggt teilt  $(x - q \cdot y)$   
 $\Rightarrow$  ggt teilt rest  
 $\Rightarrow \text{GGT}(x, y) = \text{GGT}(y, \text{rest})$

# Beispiel: Quadratwurzel von $x$ berechnen

## 2. Näherung:

$$\begin{aligned} \text{root} &\leftarrow (a + \text{root}) / 2 \\ a &\leftarrow x / \text{root} \end{aligned}$$



## Schreibtischtest

x	root	a
10	<del>5</del>	<del>2</del>
	3.5	-2.85714
	3.17857	3.14607
	3.16232	3.16223
	3.16228	3.16228

Kommazahlen sind meist nicht exakt gleich  
daher besser  $|a - \text{root}| > 0.0000001$

# 1. Einführung

1.1 Grundbegriffe

1.2 Algorithmen und Ablaufdiagramme

1.3 **Struktogramme**

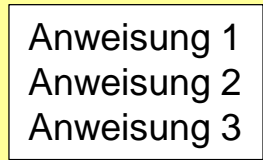
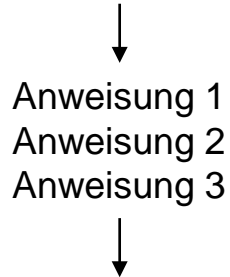
1.4 Grammatiken



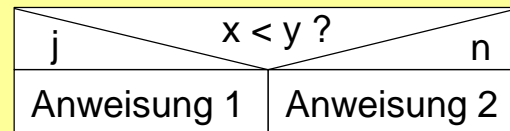
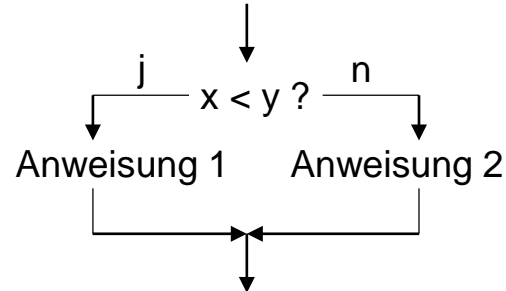
# Struktogramme (Nassi-Shneiderman-Diagramme)

Alternative Darstellungsform von Algorithmen  
 Block-artige Bausteine mit 1 Eingang und 1 Ausgang

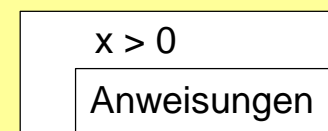
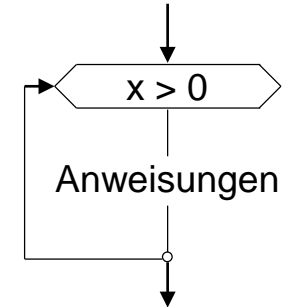
## Anweisungsfolgen



## Verzweigungen



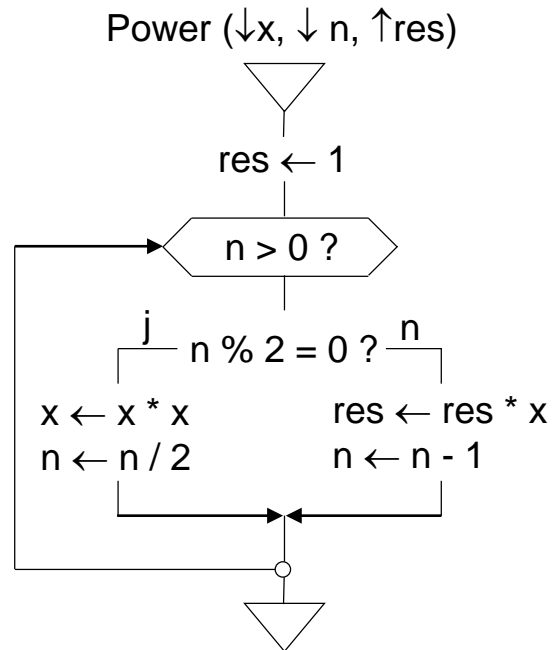
## Schleifen



- Lassen sich auch schachteln
- Aufwändig zu zeichnen
- Heute kaum noch verwendet

# Struktogramme -- Beispiel

Berechnung der Exponentialfunktion  $x^n$



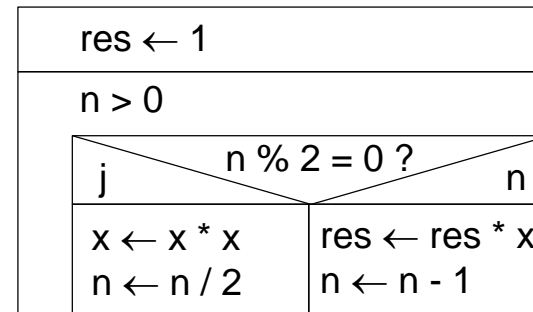
$n \% 2$  ... Rest der Division  $n / 2$

Warum funktioniert der Algorithmus?

$$x^{2n} = (x * x)^n$$

$$x^{n+1} = x * x^n$$

Power ( $\downarrow x, \downarrow n, \uparrow res$ )



Schreibtischtest für  $2^5$

x	n	res
2	5	1
	4	2
4	2	
16	1	
	0	32

# 1. Einführung

1.1 Grundbegriffe

1.2 Algorithmen und Ablaufdiagramme

1.3 Struktogramme

1.4 **Grammatiken**

## Syntax

Regeln, nach denen Sätze gebaut werden dürfen

z.B.: Zuweisung = Variable "←" Ausdruck.

## Semantik

Bedeutung der Sätze

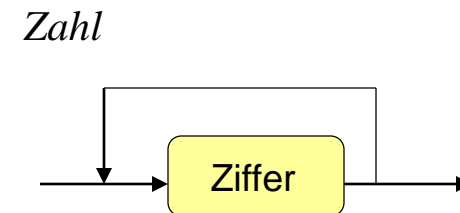
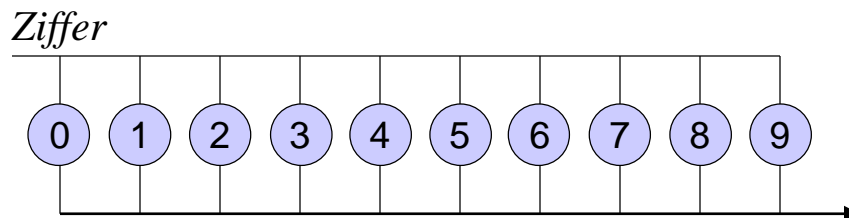
z.B.: *werte Ausdruck aus und weise ihn der Variablen zu*

## Grammatik

Menge von Syntaxregeln

z.B. Grammatik der ganzen Zahlen

Ziffer = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9".  
Zahl = Ziffer {Ziffer}.



# EBNF (Erweiterte Backus-Naur-Form)

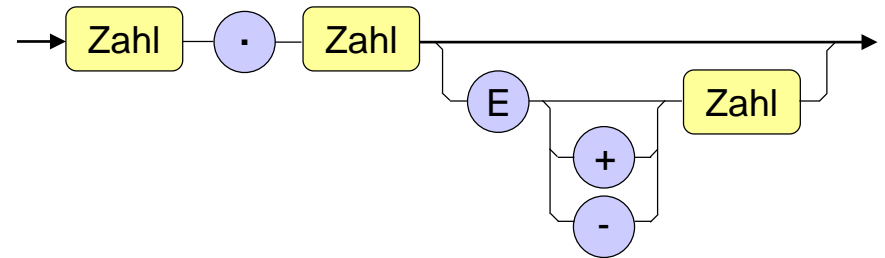


Metazeichen	Bedeutung	Beispiel	beschreibt
=	trennt Regelseiten	$A = x y z .$	
.	schließt Regel ab		
	trennt Alternativen	$x   y$	$x, y$
( )	klammert Alternativen	$(x   y) z$	$xz, yz$
[ ]	wahlweises Vorkommen	$[x] y$	$xy, y$
{ }	0..n-maliges Vorkommen	$\{x\} y$	$y, xy, xxy, xxxy, \dots$

# EBNF -- Beispiele

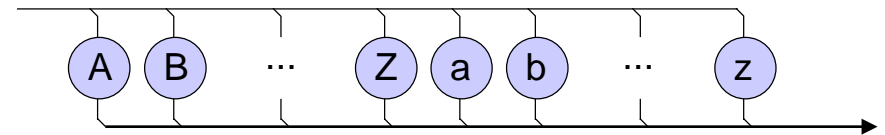
Grammatik von Gleitkommazahlen (z.B. 31.4E-1 d.h.  $31.4 * 10^{-1}$ )

Zahl = Ziffer {Ziffer}.  
 Gleitkommazahl = Zahl "." Zahl  
 ["E" ["+" | "-"] Zahl].

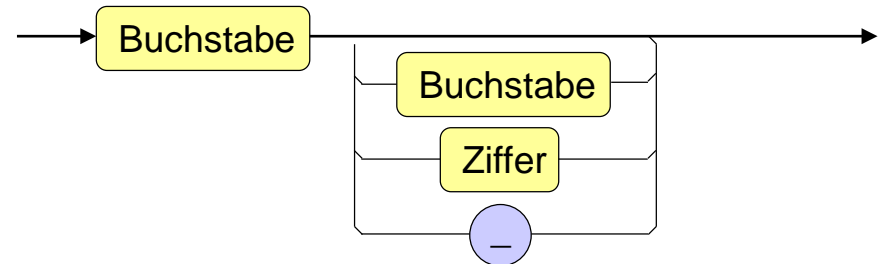


Grammatik von Namen (z.B. sum\_of\_values2)

Buchstabe = "A" | "B" | ... | "Z"  
 | "a" | "b" | ... | "z".



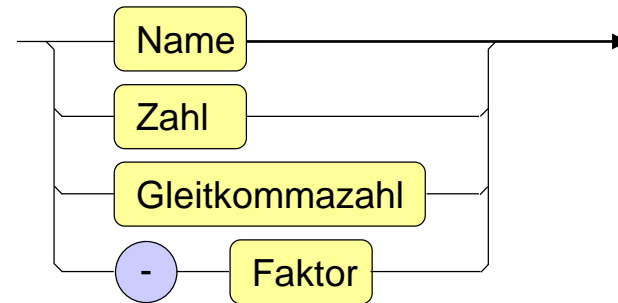
Name = Buchstabe {Buchstabe | Ziffer | "\_"}



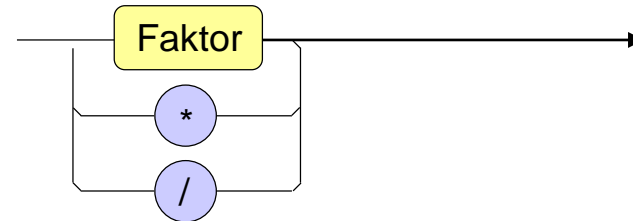
# EBNF -- Beispiele

Grammatik von arithmetischen Ausdrücken (z.B.  $3 * -4 + x / 2 - y$ )

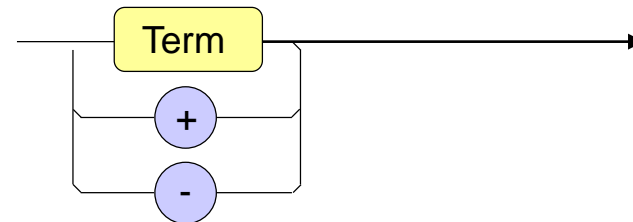
Faktor = Name  
 | Zahl  
 | Gleitkommazahl  
 | "-" Faktor.



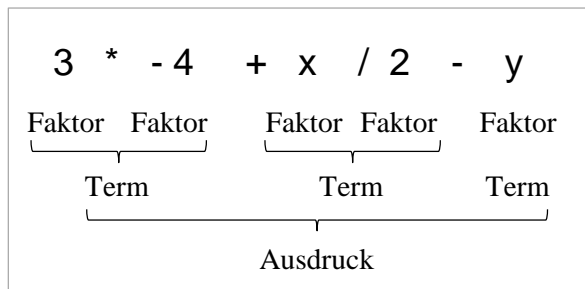
Term = Faktor { ("\*" | "/" ) Faktor }.



Ausdruck = Term { ("+" | "-" ) Term }.



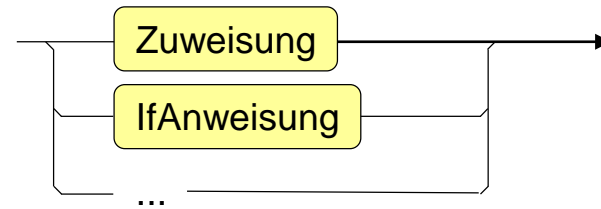
## Analyse eines Ausdrucks



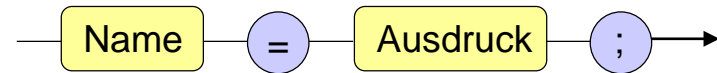
# EBNF -- Beispiele

Grammatik von Anweisungen (vereinfacht)

Anweisung = Zuweisung  
 | IfAnweisung  
 | ... .



Zuweisung = Name "=" Ausdruck ";" .



IfAnweisung = "if" "(" Ausdruck ")" Anweisung  
 [ "else" Anweisung ] .

