

Softwareentwicklung 1

Prof. Dr. Hanspeter Mössenböck



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

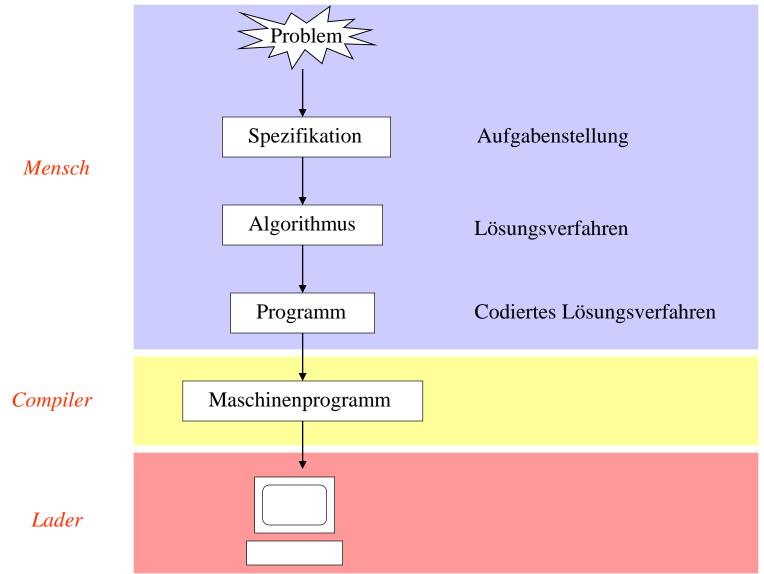
Befehle

Operationen mit den Speicherzellen

Maschinensprache		Hochsprache
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



Summiere Zahlen von 1 bis max (max, - sum)

- 1. sum 0
- 2. zahl 1
- 3. Wiederhole, solange zahl £ max

 $3.1 \quad sum \neg \quad sum + zahl$

 $3.2 \quad zahl \neg \quad zahl + 1$

Folge von Schritten

Programm = Beschreibung eines Algorithmus in einer Programmiersprache

Variablen



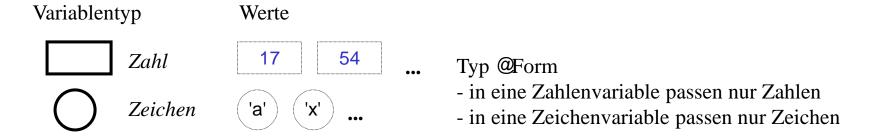
Sind benannte Behälter für Werte

X	y
99	3

Können ihren Wert ändern

$$x \rightarrow x + 1$$
 100

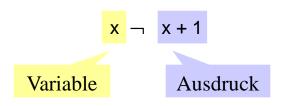
Haben einen Datentyp = Menge erlaubter Werte



Anweisungen

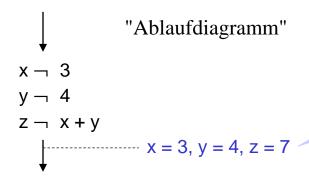


Wertzuweisung



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

Anweisungsfolge (auch Sequenz)



Assertion

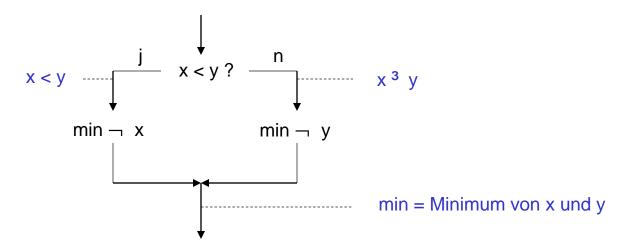
Assertion (Zusicherung)

Aussage über den Zustand des Algorithmus an einer bestimmten Stelle

Anweisungen



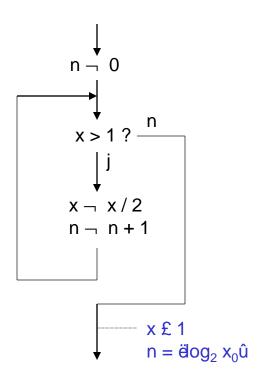
Auswahl (auch Verzweigung, Abfrage, Selektion)

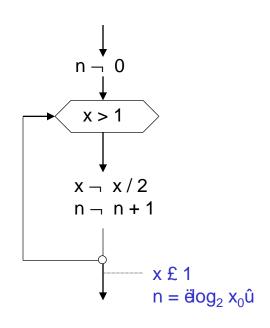


Anweisungen



Wiederholung (auch Schleife, Iteration)

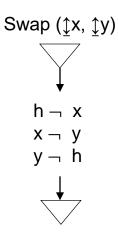




Alternative Darstellung

Beispiel: Vertauschen zweier Variableninhalte



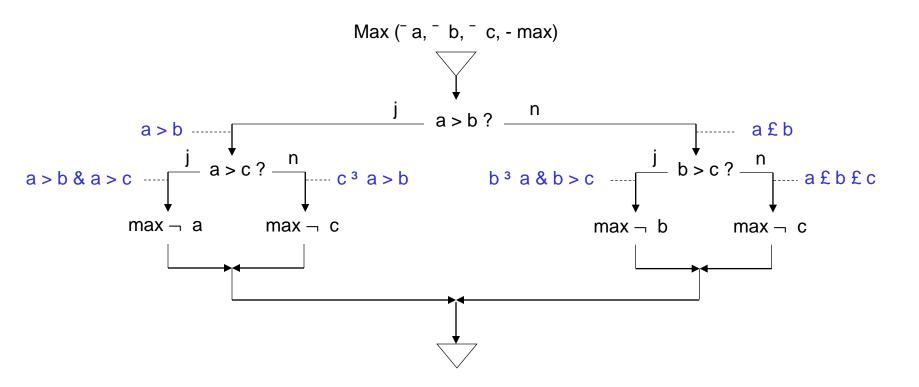


Schreibtischtest

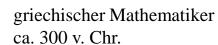
X	y	h
3	2	3
2	3	

Beispiel: Maximum dreier Zahlen bestimmen





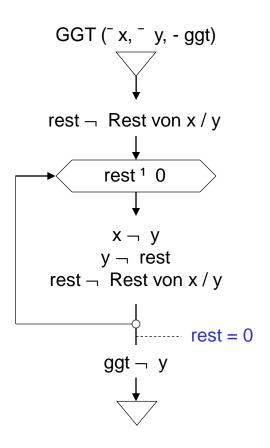
Euklid





Beispiel: Euklidscher Algorithmus

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)

$$\triangleright x = i*ggt, y = j*ggt, (x-y) = (i-j)*ggt$$

- Þ ggt teilt (x y)
- \triangleright ggt teilt (x q*y)
- b ggt teilt rest

Beispiel: Quadratwurzel von x berechnen



2. Näherung:

$$root \neg (a + root) / 2$$

 $a \neg x / root$



SquareRoot (x, - root) root - x/2a - x / root----- a * root = x a 1 root a * root = xroot - (a + root) / 2a - x / root----- a * root = x ----- a * root = x & a = root \triangleright root * root = x

Schreibtischtest

X	root	a
10	5	2
	3.5	2.85714
	3.17857	3.14607
	3.16232	3.16223
	3.16228	3.16228

Kommazahlen sind meist nicht exakt gleich

daher besser $\langle | a\text{-root} | > 0.0000001$



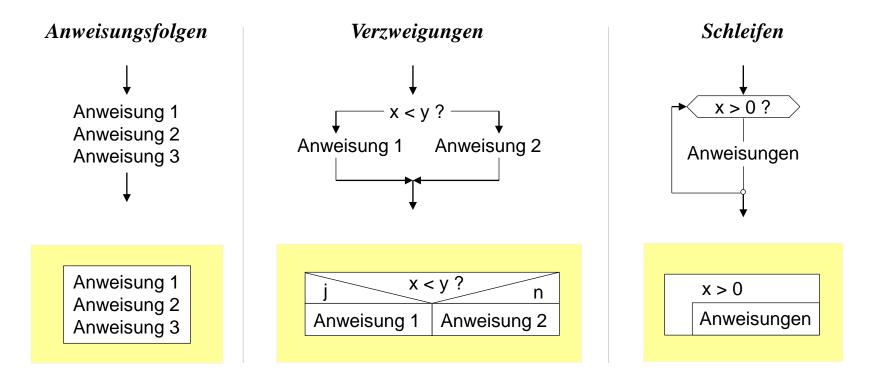
1. Einführung

- 1.1 Grundbegriffe
- 1.2 Algorithmen und Ablaufdiagramme
- 1.3 Struktogramme
- 1.4 Grammatiken

Struktugramme (Nassi-Shneiderman-Diagramme)



Alternative Darstellungsform von Algorithmen Block-artige Bausteine mit <u>1 Eingang</u> und <u>1 Ausgang</u>

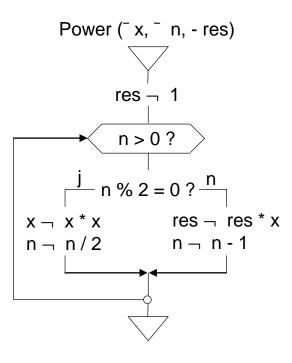


- Lassen sich auch schachteln
- Aufwändig zu zeichnen
- Heute kaum noch verwendet

Struktugramme -- Beispiel

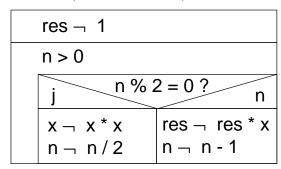


Berechnung der Exponentialfunktion xⁿ



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

Power (x, n, - res)



Schreibtischtest für 2⁵

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

Beschreibung von Programmiersprachen



Syntax

Regeln, nach denen Sätze gebaut werden dürfen

z.B.: Zuweisung = Variable " \neg " Ausdruck.

Semantik

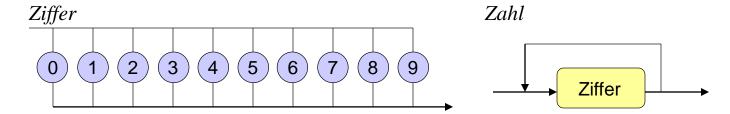
Bedeutung der Sätze

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

Grammatik

Menge von Syntaxregeln

z.B. Grammatik der ganzen Zahlen



EBNF (Erweiterte Backus-Naur-Form)

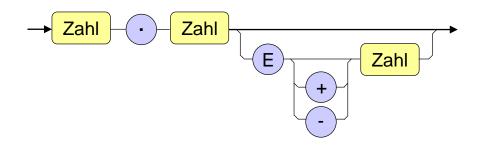


Metazeichen	Bedeutung	Beispiel	beschreibt
=	trennt Regelseiten	A = x y z .	
•	schließt Regel ab		
	trennt Alternativen	x y	x, y
()	klammert Alternativen	(x y) z	XZ, YZ
[]	wahlweises Vorkommen	[x] y	xy, y
{ }	0n-maliges Vorkommen	{x} y	y, xy, xxy, xxxy,

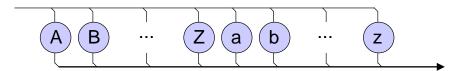
EBNF -- Beispiele



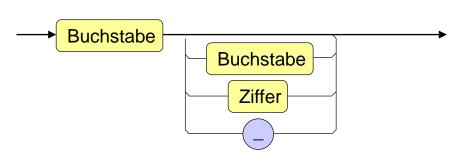
Grammatik von Gleitkommazahlen



Grammatik von Namen



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



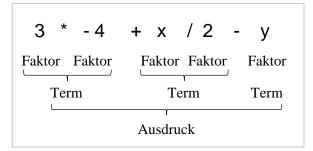
EBNF -- Beispiele

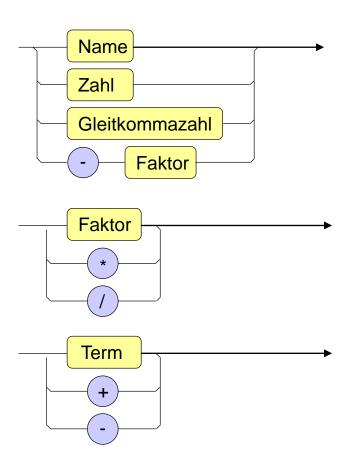


Grammatik von arithmetischen Ausdrücken

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

Analyse eines Ausdrucks



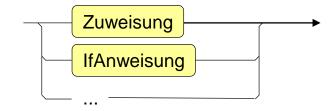


EBNF -- Beispiele



Grammatik von Anweisungen (vereinfacht)

Anweisung = Zuweisung | IfAnweisung |



Zuweisung = Name "=" Ausdruck ";".



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

