

Infix-Notation

Arithmetische Ausdrücke werden sowohl in der Mathematik als auch in Programmiersprachen üblicherweise in der so genannten Infix-Notation beschrieben. Hierbei steht der Rechenoperator zwischen den Operanden. Zusätzlich legen die Klammern die Reihenfolge der Auswertung fest („von innen nach außen“).

Infix-Term: $3 * (4 + 5)$

Der Operator „*“ steht zwischen seinen beiden Operanden 3 und $(4 + 5)$. Ohne die Klammern würde der Term von links nach rechts ausgewertet (Prioritäten werden beachtet).

Postfix-Notation

Bei der Postfix-Notation wird ein Term immer in der Reihenfolge Operand1, Operand2, Operator dargestellt, also erst die beiden Operanden, danach der Operator. Diese Schreibweise hat gegenüber der Infix-Schreibweise den Vorteil, dass die Rechenreihenfolge durch die Position des Operators eindeutig festgelegt ist. Es werden daher keine Klammern für die Festlegung der Rechenreihenfolge benötigt. Postfix-Terme sind besonders leicht auszuwerten. Deshalb wandelt der Compiler mancher Programmiersprachen einen Infix-Term vor seiner Auswertung in einen Postfix-Term um.

Postfix-Term: $3\ 4\ 5\ +\ *$

Der Term wird von links nach rechts analysiert, bis der erste Operator auftritt. Dieser wird mit den beiden Operanden davor ausgeführt, und das Ergebnis der Rechnung tritt an die Stelle des ausgewerteten Ausdrucks.

Auswertung:	$3\ 4\ 5\ +\ *$	Nächsten Operator suchen
	$3\ [4\ 5\ +]\ *$	Auswertung
	$3\ 9\ *$	
	$3\ 9\ *$	Nächsten Operator suchen
	$[3\ 9\ *]$	Auswertung
	27	
	27 _	Nächsten Operator suchen: fertig

Aufgabe 1

Wandeln Sie die Infix-Terme in Postfix-Terme um (bzw. umgekehrt):

$(3 + 5) * 8 + 11$ | $3 (3 * 8 + 1) - 9 * 5$ | $2\ 5\ 7\ 1\ 4\ * + - *$ | $9\ 3\ 2\ 1\ 7\ + + * +$ | $8\ 2\ 4\ 1\ 6\ 7\ * - * + *$

Umwandlung der Notationen ineinander

Die Umwandlung eines Infix-Terms in einen Postfix-Term kann mit Hilfe von drei ADT Stack (**StackInfix**, **StackHelp**, **StackPostfix**) erfolgen. Diese enthalten die jeweiligen arithmetischen Ausdrücke, entweder Operanden (eine Zahl), einen Operator (+, -, *, :) oder eine Klammer.

Die Umwandlung eines Infix-Terms (aus einem **StackInfix**) in einen Postfix-Term (in einem **StackPostfix**) erfolgt nach folgenden Regeln:

- Operanden werden direkt vom **StackInfix** in den **StackPostfix** verschoben.
- Operatoren und Klammern werden mit dem ToS (Top of Stack) des **StackHelp** verglichen und unter Beachtung der folgenden Regeln verschoben:

- Eine öffnende Klammer „(“ wird gleich in den **StackHelp** verschoben.
- Eine schließende Klammer „)” veranlasst das Verschieben aller Elemente vom **StackHelp** in den **StackPostfix** so lange, bis der ToS des **StackHelp** ein öffnende Klammer ist. Beide Klammern werden anschließend gelöscht.
- Ein Operator wird vom **StackInfix** in den **StackHelp** verschoben. Falls jedoch der ToS des **StackHelp** eine höhere oder gleich große Priorität als der Operator im ToS des **StackInfix** hat, wird dieser vorher in den **StackPostfix** verschoben.
- Das Ende der Eingabe veranlasst das Verschieben aller Elemente vom **StackHelp** in den **StackPostfix**.

Bsp1 $3 * (4 + 5)$

Stack Infix	Stack Help	Stack Postfix
3 *	(
* (4 + 5)	(*	3
(4 + 5)	(*	3
4 + 5)	(*	3
+ 5)	(*	4 3
5)	+ (*	4 3
)	+ (*	5 4 3
	(*	+ 5 4 3
	*	+ 5 4 3
		* + 5 4 3

↳ Term: 3 4 5 + *

Aufgabe 2

Wandeln Sie die in Aufgabe 1 gegebenen Infix-Terme mit Hilfe des gegebenen Algorithmus in Postfix-Terme um.

Aufgabe 3

Gesucht ist Java-Programm (Konsole oder GUI), welches einen gegebenen mathematischen Term (in Infix-Notation) berechnet ... und das können wir im Moment nur mit Hilfe der Umwandlung in die Postfix-Notation.