

Merkblatt zur Programmierung in C

Operatoren, deren Vorrang und Assoziativität

Rang	Symbol	Beispiel	Beschreibung	Assoziativität
1	++	x++	Erhöhung nach Auswertung (Post-Inkrement)	➔
	--	x--	Erniedrigung nach Auswertung (Post-Dekrement)	➔
	()	sin(x)	Funktionsaufruf	➔
	[]	a[0]	Arrayelement (Indexoperator)	➔
	.	q.x	Elementauswahl einer Struktur oder Union	➔
	->	p->x	Elementauswahl durch Zeiger	➔
2	++	++x	Erhöhung vor Auswertung (Pre-Inkrement)	➡
	--	--x	Erniedrigung vor Auswertung (Pre-Dekrement)	➡
	!	!x	logisches NOT	➡
	~	~x	bitweises NOT	➡
	-	-x	unäres Minus	➡
	+	+x	unäres Plus	➡
	&	&x	Adressoperator	➡
	*	*p	Dereferenzierung einer Adresse	➡
	sizeof (type)	sizeof(x) (int) x	Größe in Bytes Typumwandlung (cast)	➡
3	*	x * y	Multiplikation	➔
	/	x / y	Division	➔
	%	x % y	Modulo-Operator (Divisionsrest), nur für Datentyp int	➔
4	+	x + y	Addition (binäres Plus)	➔
	-	x - y	Subtraktion (binäres Minus)	➔
5	<<	n << i	bitweises Linksschieben, nur für Datentyp int	➔
	>>	n >> i	bitweises Rechtsschieben, nur für Datentyp int	➔
6	<	x < y	kleiner als	➔
	<=	x <= y	kleiner oder gleich	➔
	>	x > y	größer als	➔
	>=	x >= y	größer oder gleich	➔
7	==	x == y	gleich	➔
	!=	x != y	ungleich	➔
8	&	x & y	bitweises AND	➔
9	^	x ^ y	bitweises EXCLUSIVE OR	➔
10		x y	bitweises OR	➔
11	&&	x && y	logisches AND	➔
12		x y	logisches OR	➔
13	? :	x ? y : z	Bedingungsoperator	➡
14	=	x = y	Zuweisung	➡
	*=	x *= y	zusammengesetzte Zuweisung	➡
	/=			➡
	%=			➡
	+=			➡
	-=			➡
	<<=			➡
	>>=			➡
	&=			➡
	^=			➡
	=			➡
15	,	int i, j;	Komma-Operator	➔

Die Auswertungsreihenfolge von Ausdrücken wird durch den Vorrang der Operatoren bestimmt. Die Assoziativität gibt an, ob eine Folge von Operatoren gleichen Vorrangs von links oder von rechts abgearbeitet wird:

Rang 1 höchster Rang

Rang 15 niedrigster Rang

➔ Assoziativität von links nach rechts, d.h. erst Auswertung der linken Seite. Z.B. **i*j/k** ist äquivalent zu **(i*j)/k**

➡ Assoziativität von rechts nach links, d.h. erst Auswertung der rechten Seite. Z.B. **-*p** ist äquivalent zu **-(*p)**