

Java Operators

© Copyright 1998-2003 by Markus Falkhausen, all rights reserved.
Most recent version: www.falkhausen.de

Priority	Operator	Name	Associativity	Example	Result*	Incidents per 100 Lines
1	++	Increment	r	$x++$ $++x$	3.5 (+ effect) 4.5 (+ effect)	.77
	--	Decrement	r	$x--$ $--x$	3.5 (+ effect) 2.5 (+ effect)	.09
	+	Unary plus	r	$+x$	3.5	?
	-	Unary minus	r	$-x$	-3.5	?
	!	Logical complement	r	<code>isOpen</code>	false	.28
	~	Bitwise complement	r	$\sim i$	-5	.01
	(type)	Cast	r	<code>i = (int) x</code>	3	?
2	*	Multiplication	l	$x * 2$	7.	1.24
	/	Division	l	$x / 2$	1.75	.22
	%	Remainder	l	$x \% 2$	1.5	.04
3	+	Binary plus	l	$x + 2$ " " + x + i	5.5 " 3.54"	(2.70)
	-	Binary minus	l	$x - i$	-0.5	(1.46)
4	<<	Shift left	l	$i << 2$	16	.08
	>>	Shift right	l	$-i >> 2$	-1	.04
	>>>	Shift right ignore sign	l	$-i >>> 2$	1073741823	.02
5	>	greater than	l	$i > x$	true	.36
	<	lesser than	l	$i < x$	false	.86
	>=	greater equal	l	$i >= x$	true	.14
	<=	lesser equal	l	$i <= x$	false	.24
	instanceof	Type check	l	<code>s instanceof String</code>	true	.25

*Results are given for the declarations: int i=4, int j=2, double x = 3.5, String s="", boolean isOpen=true.

Priority	Operator	Name	Associativity	Example	Result*	Incidents per 100 Lines
6	==	Equals	l	$i == j$ $s == ""$	false true	1.28
	!=	Not equal	l	$i != j$ $s != null$	true true	1.17
7	&	Bitwise and	l	$i \& j$	0	.18
8	^	Exclusive or	l	$i \wedge 5$	1	.01
9		Bitwise or	l	$i j$	6	.10
10	&&	Logical and	l	<code>isOpen && false</code>	false	.58
11		Logical or	l	<code>isOpen false</code>	true	.33
12	?:	Conditional	r	<code>i<0 ? -1 : 1</code>	1	.20
13	=	Assignment	r	$j = i$ <code>o = s;</code>	4 (+ effect) " " (+ effect)	9.68
	+=	Plus assignment	r	$j += x$	5 (+ effect)	.27
	-=	Minus assignment	r	$j -= x$	-1 (+ effect)	.09
	*=	Multiplication assign.	r	$j *= x$	7 (+ effect)	.02
	/=	Division assign.	r	$j /= x$	0 (+ effect)	0
	&=	Bitwise and assign.	r	$j \&= i$	0 (+ effect)	.01
	=	Bitwise or assign.	r	$j = i$	6 (+ effect)	.03
	^=	Exclusive or assign.	r	$j \wedge= i$	6 (+ effect)	0
	%=	Remainder assign.	r	$j \% = i$	1 (+ effect)	0
	<<=	Shift left assign.	r	$j <<= i$	32 (+ effect)	0
	>>=	Shift right assign.	r	$j >>= i$	0 (+ effect)	0
	>>>=	Shift right i.s. assign.	r	$j >>>= i$	0 (+ effect)	0