

Operadores de *Bits* do Java

Quinta-feira, 21 de abril de 2022.

Os operadores de *bits* permitem manipular os *bits* dos tipos de dados integrais. Os tipos integrais representam números inteiros, como *byte*, *char*, *short*, *int* e *long*.

Operador	Nome	Descrição
&	E sobre bits	Define cada bit no resultado como 1 se o bit correspondente em ambos os operando for 1, assim: $0 \& 0 = 0$ $0 \& 1 = 0$ $1 \& 0 = 0$ $1 \& 1 = 1$
 	OU inclusivo sobre bits	Define cada bit no resultado como 1 se o bit correspondente em um ou ambos os operando for 1, assim: $0 0 = 0$ $0 1 = 1$ $1 0 = 1$ $1 1 = 1$
^	OU exclusivo sobre bits	Define cada bit no resultado como 1 se o bit correspondente em um operando for 0 e no outro 1, assim: $0 \wedge 0 = 0$ $0 \wedge 1 = 1$ $1 \wedge 0 = 1$ $1 \wedge 1 = 0$
~	Complemento de bits ou Complemento de um	Define todos os bits 0 no operando como 1 e todos os bits 1 como 0. É um inversor de <i>bits</i> , assim: $\sim 0 = 1$ $\sim 1 = 0$
<<	Deslocamento de bits para esquerda	Desloca os bits do operando esquerdo para a esquerda de acordo com o número de bits definido no operando direito. Os bits vagos à direita são substituídos por zero. operando << número de bits
>>	Deslocamento de bits para direita com sinal	Desloca os bits do operando esquerdo para a direita de acordo com o número de bits definido no operando direito. Se o operando esquerdo for negativo, uns são inseridos a partir da esquerda; caso contrário, zeros são inseridos a partir da esquerda. operando >> número de bits
>>>	Deslocamento de bits para direita sem sinal	Desloca os bits do operando esquerdo para a direita de acordo com o número de bits definido no operando direito; zeros são inseridos a partir da esquerda. operando >>> número de bits

Como o comando desloca os bits para além do fim do comprimento dos bits, zeros são introduzidos no outro lado. Lembre-se de que um deslocamento não é uma rotação; os bits deslocados para além do final de um lado não voltam pelo outro lado. Os bits deslocados para fora se perdem, e zeros são introduzidos para substituí-los.

Um deslocamento para a esquerda multiplicará um número por dois e um deslocamento para a direita dividirá por dois.

> Operadores Lógicos Bit a Bit

& (E bit a bit): $6 \& 5 = 4$

| (OU bit a bit): $6 | 5 = 7$

~ (NÃO bit a bit): $\sim 6 = -7$

^ (OU Exclusivo bit a bit): $6 \wedge 5 = 3$

<< (deslocamento bit a bit à esquerda): $16 \ll 1 = 32$

<< (deslocamento bit a bit à esquerda): $16 \ll 2 = 64$

>> (deslocamento bit a bit à direita sem sinal): $16 \gg 1 = 8$

>> (deslocamento bit a bit à direita sem sinal): $16 \gg 2 = 4$

As tabelas abaixo apresentam como cada resultado da tela acima foi obtido.

$$6 \& 5 = 4$$

Valor significativo do bit	128	64	32	16	8	4	2	1
Comprimento do número com 8 bits (1 byte)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
6 em binário	0	0	0	0	0	1	1	0
5 em binário	0	0	0	0	0	1	0	1
Resultado do & = 4 em binário	0	0	0	0	0	1	0	0

$$6 | 5 = 7$$

Valor significativo do bit	128	64	32	16	8	4	2	1
Comprimento do número com 8 bits (1 byte)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
6 em binário	0	0	0	0	0	1	1	0
5 em binário	0	0	0	0	0	1	0	1
Resultado do = 7 em binário	0	0	0	0	0	1	1	1

$$\sim 6 = -7$$

Valor significativo do bit	128	64	32	16	8	4	2	1
Comprimento do número com 8 bits (1 byte)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
6 em binário	0	0	0	0	0	1	1	0
Resultado do ~ = -6 em binário	0	0	0	0	0	1	1	1

O complemento de dois é usado pelo computador para representar um número negativo, como o inverso de +6 é -6, então o complemento de dois é aplicado para representar o número -6 resultante

do complemento de um sobre +6. O complemento de dois é obtido aplicando no número o complemento de um e somando mais 1. Como o resultado do complemento de um aplicado sobre o número 6 em binário é:

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Complemento de um sobre o número 6 = X	1	1	1	1	1	0	0	1

Deve-se aplicar o complemento de dois sobre esse número X para obter sua representação negativa, então se aplica o complemento de um sobre ele e soma 1. Logo, temos que:

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Complemento de um sobre X = ~X	0	0	0	0	0	1	1	0
1 em binário	0	0	0	0	0	0	0	1
Resultado da soma binária de 1 com ~X	0	0	0	0	0	1	1	1

$$6 \wedge 5 = 3$$

Valor significativo do bit	128	64	32	16	8	4	2	1
Comprimento do número com 8 bits (1 byte)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
6 em binário	0	0	0	0	0	1	1	0
5 em binário	0	0	0	0	0	1	0	1
Resultado do \wedge = 3 em binário	0	0	0	0	0	0	1	1

$$16 \ll 1 = 32$$

Valor significativo do bit	128	64	32	16	8	4	2	1
Comprimento do número com 8 bits (1 byte)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
16 em binário	0	0	0	1	0	0	0	0
Resultado de $16 \ll 1 = 32$ em binário	0	0	1	0	0	0	0	0

O dígito 1 do *bit* 4 no número 16 foi deslocado uma posição para a esquerda, portanto no resultado ele ocupa a posição do *bit* 5, assim seu valor significativo em decimal mudou de 16 para 32.

$$16 \ll 2 = 64$$

Valor significativo do bit	128	64	32	16	8	4	2	1
Comprimento do número com 8 bits (1 byte)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
16 em binário	0	0	0	1	0	0	0	0
Resultado de $16 \ll 2 = 64$ em binário	0	1	0	0	0	0	0	0

O dígito 1 do *bit* 4 no número 16 foi deslocado duas posições para a esquerda, portanto no resultado ele ocupa a posição do *bit* 6, assim seu valor significativo em decimal mudou de 16 para 64.

$$16 \gg 1 = 8$$

Valor significativo do bit	128	64	32	16	8	4	2	1
Comprimento do número com 8 bits (1 byte)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
16 em binário	0	0	0	1	0	0	0	0
Resultado de $16 \gg 1 = 8$ em binário	0	0	0	0	1	0	0	0

O dígito 1 do *bit* 4 no número 16 foi deslocado uma posição para a direita, portanto no resultado ele ocupa a posição do *bit* 3, assim seu valor significativo em decimal mudou de 16 para 8.

$$16 \gg 2 = 4$$

Valor significativo do bit	128	64	32	16	8	4	2	1
Comprimento do número com 8 bits (1 byte)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
16 em binário	0	0	0	1	0	0	0	0
Resultado de $16 \gg 2 = 4$ em binário	0	0	0	0	0	1	0	0

O dígito 1 do *bit* 4 no número 16 foi deslocado duas posições para a direita, portanto no resultado ele ocupa a posição do *bit* 2, assim seu valor significativo em decimal mudou de 16 para 4.

Prof. Márlon Oliveira da Silva
marlon.silva@ifsudestemg.edu.br