



**INSTITUTO
FEDERAL**

Sudeste de Minas Gerais

Campus
Barbacena

Curso Superior de Tecnologia em Sistemas para Internet
Tecnologia de Orientação a Objetos

Introdução aos aplicativos Java

Prof. Márlon Oliveira da Silva

marlon.silva@ifsudestemg.edu.br

10/03/2023

Sequências de Escape

Sequência de escape	Descrição
<code>\n</code>	Nova linha. Posiciona o cursor de tela no início da próxima linha.
<code>\t</code>	Tabulação horizontal. Move o cursor de tela para a próxima parada de tabulação.
<code>\r</code>	Retorno de carro. Posiciona o cursor da tela no início da linha atual — não avança para a próxima linha. Qualquer saída de caracteres depois do retorno de carro sobrescreve a saída de caracteres anteriormente gerados na linha atual.
<code>\\</code>	Barras invertidas. Utilizada para imprimir um caractere de barra invertida.
<code>\"</code>	Aspas duplas. Utilizada para imprimir um caractere de aspas duplas. Por exemplo, <pre>System.out.println("\"in quotes\"");</pre> exibe <code>"in quotes"</code>

Tipos Primitivos de Dados

Tipo	Tamanho em bits	Valores	Padrão
boolean		true ou false	
[Nota: a representação de um boolean é específica à Java Virtual Machine em cada plataforma.]			
char	16	'\u0000' a '\uFFFF' (0 a 65535)	(conjunto de caracteres Unicode ISO)
byte	8	-128 a $+127$ (-2^7 a $2^7 - 1$)	
short	16	-32.768 a $+32.767$ (-2^{15} a $2^{15} - 1$)	
int	32	$-2.147.483.648$ a $+2.147.483.647$ (-2^{31} a $2^{31} - 1$)	
long	64	$-9.223.372.036.854.775.808$ a $+9.223.372.036.854.775.807$ (-2^{63} a $2^{63} - 1$)	
float	32	<i>Intervalo negativo:</i> $-3,4028234663852886E+38$ a $-1,40129846432481707e-45$ <i>Intervalo positivo:</i> $1,40129846432481707e-45$ a $3,4028234663852886E+38$	(ponto flutuante IEEE 754)
double	64	<i>Intervalo negativo:</i> $-1,7976931348623157E+308$ a $-4,94065645841246544e-324$ <i>Intervalo positivo:</i> $4,94065645841246544e-324$ a $1,7976931348623157E+308$	(ponto flutuante IEEE 754)

Operadores Aritméticos

Operação Java	Operador	Expressão algébrica	Expressão Java
Adição	+	$f + 7$	<code>f + 7</code>
Subtração	-	$p - c$	<code>p - c</code>
Multiplicação	*	bm	<code>b * m</code>
Divisão	/	x/y ou $\frac{x}{y}$ ou $x \div y$	<code>x / y</code>
Resto	%	$r \bmod s$	<code>r % s</code>

Operadores de Incremento e Decremento

Operador	Nome do operador	Expressão de exemplo	Explicação
++	pré-incremento	++a	Incrementa a por 1, então utiliza o novo valor de a na expressão em que a reside.
++	pós-decremento	a++	Utiliza o valor atual de a na expressão em que a reside, então incrementa a por 1.
--	pré-incremento	--b	Decrementa b por 1, então utiliza o novo valor de b na expressão em que b reside.
--	pós-decremento	b--	Utiliza o valor atual de b na expressão em que b reside, então decrementa b por 1.

Operadores de Atribuição

Operador de atribuição	Expressão de exemplo	Explicação	Atribuições
Suponha: <code>int c = 3, d = 5, e = 4, f = 6, g = 12;</code>			
<code>+=</code>	<code>c += 7</code>	<code>c = c + 7</code>	10 a c
<code>-=</code>	<code>d -= 4</code>	<code>d = d - 4</code>	1 a d
<code>*=</code>	<code>e *= 5</code>	<code>e = e * 5</code>	20 a e
<code>/=</code>	<code>f /= 3</code>	<code>f = f / 3</code>	2 a f
<code>%=</code>	<code>g %= 9</code>	<code>g = g % 9</code>	3 a g

Operadores Relacionais

Operador de igualdade ou relacional algébrico padrão	Operador de igualdade ou relacional Java	Exemplo de condição em Java	Significado da condição em Java
<i>Operadores de igualdade</i>			
=	==	x == y	x é igual a y
≠	!=	x != y	x é diferente de y
<i>Operadores relacionais</i>			
>	>	x > y	x é maior que y
<	<	x < y	x é menor que y
≥	>=	x >= y	x é maior que ou igual a y
≤	<=	x <= y	x é menor que ou igual a y

Operadores Lógicos

E Lógico (&&)

expressão1	expressão2	expressão1 && expressão2
false	false	false
false	true	false
true	false	false
true	true	true

Operadores Lógicos

OU Lógico (||)

expressão1	expressão2	expressão1 expressão2
false	false	false
false	true	true
true	false	true
true	true	true

Operadores Lógicos

OU Exclusivo Lógico (^)

expressão1	expressão2	expressão1 ^ expressão2
false	false	false
false	true	true
true	false	true
true	true	false

NÃO Lógico ou Negação Lógica (!)

expressão	! expressão
false	true
true	false

Operadores *Bit a Bit*

Operador	Nome	Descrição
&	E sobre bits	Os bits no resultado são configurados como 1 se os bits correspondentes nos dois operandos forem ambos 1.
	OU inclusivo sobre bits	Os bits do resultado são configurados como 1 se pelo menos um dos bits correspondentes nos dois operandos for 1.
^	OU exclusivo sobre bits	Os bits do resultado são configurados como 1 se exatamente um dos bits correspondentes nos dois operandos for 1.
<<	deslocamento de bits para a esquerda	Desloca os bits do operando esquerdo à direita pelo número de bits especificado pelo operando direito; preenche a partir da direita com 0s.
>>	deslocamento para a direita com sinal	Desloca os bits do operando esquerdo à direita pelo número de bits especificado pelo operando direito. Se o operando esquerdo for negativo, 1s são inseridos a partir da esquerda; caso contrário, 0s são inseridos a partir da esquerda.
>>>	deslocamento para a direita sem sinal	Desloca os bits do operando esquerdo à direita pelo número de bits especificado pelo segundo operando; 0s são preenchidos a partir da esquerda.
~	complemento de bits	Todos os bits 0 são configurados como 1 e todos os bits 1 são configurados como 0.

Precedência e Associatividade dos Operadores

Operador	Descrição	Associatividade
++ --	incremento pós-fixado unário decremento pós-fixado unário	da direita para a esquerda
++ -- + - ! ~ (tipo)	pré-incremento unário pré-decremento unário mais unário menos unário negação unária lógica complemento unário sobre bits coerção unária	da direita para a esquerda
* / %	multiplicação divisão módulo	da esquerda para a direita
+ -	adição ou concatenação de strings subtração	da esquerda para a direita
<< >> >>>	deslocamento de bits para a esquerda deslocamento para a direita com sinal deslocamento para a direita sem sinal	da esquerda para a direita
< <= > >= instanceof	menor que menor que ou igual a o maior que maior que ou igual a comparação de tipo	da esquerda para a direita
== !=	é igual a não é igual a	da esquerda para a direita
& ^	E sobre bits E lógico booleano OU exclusivo sobre bits OU lógico booleano exclusivo	da esquerda para a direita

Precedência e Associatividade dos Operadores

Operador	Descrição	Associatividade
	OU inclusivo sobre bits OU inclusivo lógico booleano	da esquerda para a direita
&&	E condicional	da esquerda para a direita
	OU condicional	da esquerda para a direita
?:	condicional	da direita para a esquerda
=	atribuição	da direita para a esquerda
+=	atribuição de adição	
-=	atribuição de subtração	
*=	atribuição de multiplicação	
/=	atribuição de divisão	
%=	atribuição de resto	
&=	atribuição E sobre bits	
^=	atribuição de OU exclusivo sobre bits	
=	atribuição de OU inclusivo sobre bits	
<<=	atribuição de deslocamento para a esquerda sobre bits	
>>=	atribuição de bits com deslocamento para a direita com sinal	
>>>=	atribuição de bits com deslocamento para a direita com sinal	

Referências

DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**. 8ª ed. São Paulo: Pearson Prentice Hall, 2010.