

Java Orientado a Objetos

Operadores



|| (ou lógico) e | (ou binário)

Condição 1	Operador	Condição2	Resultado
true		true	true
false		true	true
true		false	true
false		false	false



A diferença básica entre os operadores || e |, é que, semelhante ao operador &&, o || também suporta a avaliação parcial.

\wedge (ou exclusivo binário)



Condição 1	Operador	Condição2	Resultado
true	\wedge	true	false
false	\wedge	true	true
true	\wedge	false	true
false	\wedge	false	false

! (Negação)



Condição	Operador	Resultado
true	!	false
false	!	true

Operadores de Incremento e Decremento

int varInt = 1;

varInt



varInt



varInt



varInt

Precedência de Operadores



1º Calcula o parêntese em maior nível

3º Calcula as operações de adição e subtração

$$((10\%4)*5)+(4/2)+88-10$$

$$(2*5)+(4/2)+88-10$$

$$10+2+88-10$$

$$90$$

2º Calcula os valores dos parênteses restantes

Operador Condicional (?:)

Expressão Booleana
retorna **true** ou **false**

exp1

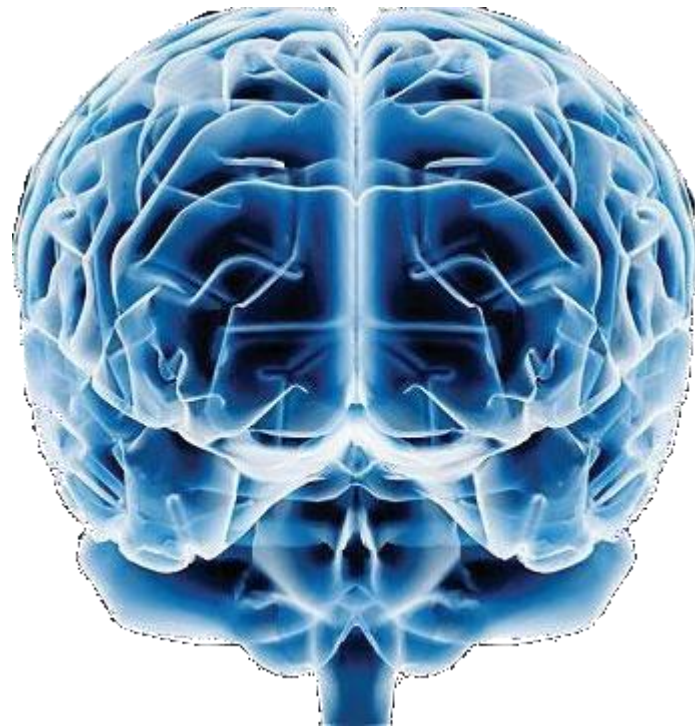
?

exp2 : exp3

Se o valor de **exp1** for **verdadeiro**, então o
resultado será **exp2**, caso contrário, **exp3**

Java Orientado a Objetos

Estruturas de Controle



Operadores Aritméticos



*	var1 * var2	Multiplicação
+	var1 + var2	Adição
-	var1 - var2	Subtração
%	var1 % var2	Resto da divisão
/	var1 / var2	Divisão

Operadores Relacionais

!=

|
==
<=
|
|
|
<

	var1 var2	Maior que
=	var1 = var2	Maior ou igual
<	var1 < var2	Menor que
<=	var1 <= var2	Menor ou igual
=	var1 == var2	Igual
!=	var1 != var2	Diferente

Operadores Lógicos

&
&

&

!

||

&&	var1 &&var2	'E' lógico (AND)
&	var1 & var2	'E' binário
	var1 var2	'OU' lógico (OR)
	var1 var2	'OU' binário
^	var1 ^ var2	OU' exclusivo binário
!	var1 ! var2	Negação (NOT)

&& (e lógico) e & (e binário)

Condição 1	Operador	Condição 2	Resultad o
true	&&	true	true
false	&&	true	false
true	&&	false	false
false	&&	false	false



A diferença básica do operador && para & é que o && suporta uma avaliação de curto- circuito (ou avaliação parcial), enquanto que o & não.