

INTRODUÇÃO A LÓGICA

Curso Superior Tecnológico em

Sistemas para Internet

**Semana4 – Operadores de Atribuição,
Aritméticos, Relacionais e Fluxo Condicional**

Instituto Federal do Tocantins

Prof. Helder – helder@ifto.edu.br

Operador de atribuição

O operador de atribuição na linguagem java é representada pelo sinal de igualdade, sendo usado para:

- > atribuir literais à variáveis – ex: int x = 2;
- > variáveis à variáveis – ex. int x =2; int y = x;

As atribuições devem respeitar questões de compatibilidade, ou seja, deve-se atribuir a uma variável um tipo que pode ser suportado por ela. Em alguns casos é possível fazer cast durante o processo de atribuição.

Ex: int x = (int) 3.2; //sendo que 3.2 é um tipo double.

Operadores aritméticos

 +	adição	$x=x + 3$
 -	subtração	$x=x - 3$
 /	divisão	$x=y / 4$
 %	resto da divisão	$x=y \% 5$
 ++	incremento de 1	$x++$ $++x$
 --	decremento de 1	$x--$ $-x$
	* multiplicação	$x = x * 3$

`int x = 6 / 3; // o valor de x passa a ser 2, uma vez que x recebe o quociente da divisão de 6 por 3`

`Int x = 6 % 3; // o valor de x passa a ser 0, uma vez que x passa a receber o resto da divisão de 6 por 3`

`int x =2; x++; //o resultado da execução dessas duas instruções deixa o valor de x passa a ser 3;`

`int x =2; x--; //o resultado da execução dessas duas instruções deixa o valor de x passa a ser 1;`

Os operadores de uma operação aritmética podem ser: apenas literal, apenas variáveis ou literal e variáveis juntas.

Operadores aritméticos

■ $+=$	atribuição aditiva	$x+=3$
	tem o mesmo efeito de	$x=x+3$
■ $-=$	atribuição subtrativa	$x-=3$
	tem o mesmo efeito de	$x=x-3$
■ $*=$	atribuição multiplicativa	$x*=3$
■ $/=$	atribuição divisiva	$x/=3$
■ $\% =$	atribuição de resto	$x\%=3$

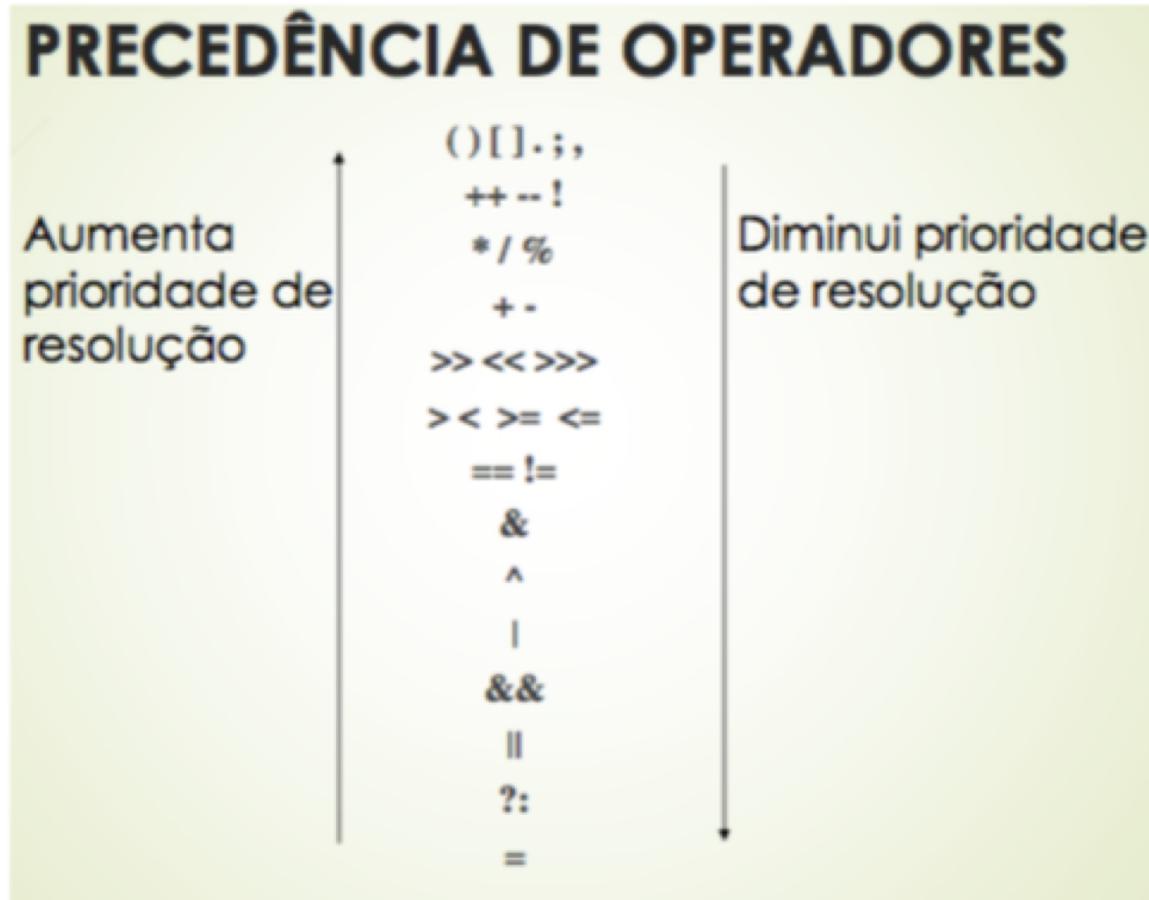
Esta é a forma simplificada dos operadores mostrados no slide anterior.

Operadores relacionais

► ==	igual	$x==y$
► !=	diferente	$x!=y$
► >	maior	$x>y$
► <	menor	$x<y$
► >=	maior ou igual	$x>=y$
► <=	menor ou igual	$x<=y$

A execução de operadores relacionais sempre resulta na produção de um valor primitivo do tipo **boolean**;

Precedência de operadores



Observe a instrução abaixo:

$$X = 6/3 + 4 \% 2;$$

Baseadas nas procedências dos operadores podemos decompor a instrução:

1. = atribuição
2. / divisão
3. + soma
4. % resto da divisão

Ordem de execução: 2 -> 4 -> 3 -> 1

Operadores - Resumo

Basicamente os operadores, sejam aritméticos ou relacionais são usados quase sempre apenas entre tipos primitivos e com raras aplicabilidade para tipos não primitivos (objetos).

Fluxo Condisional (if-else)

Existe um “método do compilador” utilizado para fazer perguntas, obtendo duas possíveis repostas (sim ou não) ou seja a resposta às perguntas sempre será um valor booleano, mais precisamente o tipo primitive **boolean**.

A estrutura do método:

```
if (dado){  
    ....  
}else{  
    ....  
}
```

Dentro do bloco do if pode ser colocado qualquer tipo de instrução, bem como dentro do bloco do else.

O bloco do else é opcional ele pode existir ou não, isso depende da lógica do programador.

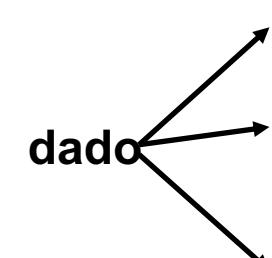
Durante a execução do programa só se executa o bloco if ou o bloco else, jamais os dois.

Executa-se o bloco if se o dado for verdadeiro e bloco else se o dado for falso.

Fluxo Condisional (if-else)

```
if (dados){  
    ....  
} else{  
    ....  
}
```

Dado pode ser:

- 
- Diretamente um valor Boolean (true ou false)
 - Um método que retorne um valor boolean
 - Uma expressão matemática usando operadores relacionais que ao ser executada resulte um valor boolean