



**CURSO**  
**TECNICO EM INFORMATICA**  
**DISCIPLINA**  
**LINGUAGEM DE PROGRAMAÇÃO-II**



**PROFESSOR: EDILSON MELO**



# OPERADORES NO JAVA

## OPERADORES DE ATRIBUIÇÃO EM JAVA

- O operador de atribuição é utilizado para definir o valor inicial ou sobrescrever o valor de uma variável. Em seu uso, o operando à esquerda representa a variável para a qual desejamos atribuir o valor informado à direita.

Exemplo de uso:

```
int lado = 2;  
float pi = 3.1426F;  
String texto = "DevMedia";  
lado = 3;
```

- Nesse exemplo iniciamos as variáveis lado, pi e texto, sobrescrevendo a variável lado em seguida.

# OPERADORES NO JAVA

## OPERADORES ARITMÉTICOS :

- Os operadores aritméticos realizam as operações fundamentais da matemática entre duas variáveis e retornam o resultado.
- Caso seja necessário escrever operações maiores ou mais complexas, podemos combinar esses operadores e criar expressões, o que nos permite executar todo tipo de cálculo de forma programática.

```
int area = 2 * 2;
```

- Esse código demonstra como calcular a área de um quadrado de lado igual a 2.

# OPERADORES NO JAVA

## OPERADORES ARITMÉTICOS :

- Também podemos utilizar os operadores aritméticos em conjunto com o operador de atribuição, realizando, em uma mesma instrução, as ações de calcular o valor e atribuí-lo à variável.

Exemplo de uso:

```
int area = 2;  
area *= 2;
```

- *A segunda linha desse código é equivalente a `area = area * 2.`*

# OPERADORES NO JAVA

## OPERADORES ARITMÉTICOS :

### ➤ OPÇÕES DE OPERADORES ARITMÉTICOS

✓ A tabela abaixo apresenta os operadores aritméticos da linguagem Java:

|          |   |
|----------|---|
| <b>+</b> | <b>operador de adição</b>                       |
| <b>-</b> | <b>operador subtração</b>                       |
| <b>*</b> | <b>operador de multiplicação</b>                |
| <b>/</b> | <b>operador de divisão</b>                      |
| <b>%</b> | <b>operador de módulo (ou resto da divisão)</b> |

# OPERADORES NO JAVA

## OPERADORES DE INCREMENTO E DECREMENTO:

- Os **operadores de incremento** e decremento também são bastante utilizados. Basicamente temos dois deles:++ e --, os quais podem ser declarados antes ou depois da variável e incrementam ou decrementam em 1 o valor da variável.

Exemplo de uso:

```
int numero = 5;  
numero++;  
numero--;  
//numero continuará sendo 5.
```



# OPERADORES NO JAVA

## OPERADORES DE INCREMENTO E DECREMENTO:

- Quando declaramos esse operador antes da variável, o incremento é realizado antes do valor da variável ser lido para o processamento ao qual a instrução pertence.
- Quando declarado depois, ocorre o contrário: lê-se o valor da variável para processamento e só então o valor da variável é incrementado. Com base nisso, suponha que temos o código abaixo:

Exemplo de uso:



# OPERADORES NO JAVA

```
int desafioUm = 5;  
System.out.println(desafioUm += ++desafioUm );  
  
int desafioDois = 5;  
System.out.println(desafioDois += desafioDois++);
```

- Quais valores serão impressos no console?
- 10 e 10, 10 e 11, 11 e 10 ou 11 e 11?
- A resposta é 11 e 10.
- No primeiro println(), desafioUm é incrementado antes de seu valor ser lido para compor a instrução de soma. Sendo assim, temos  $\text{desafioUm} = 5 + 6$ .
- Já no segundo println(), primeiro o valor é lido, resultando em  $\text{desafioDois} = 5 + 5$ . Somente após a leitura desafioDois é incrementado, e depois, recebe o valor da soma, tendo seu valor sobrescrito com o número 10.



# OPERADORES NO JAVA

## OPERADORES DE IGUALDADE:

- Os operadores de igualdade verificam se o valor ou o resultado da expressão lógica à esquerda é igual (“==”) ou diferente (“!=”) ao da direita, retornando um valor booleano.

Exemplo de uso:

```
int valorA = 1;
int valorB = 2;

if(valorA == valorB) {
    System.out.println("Valores iguais");
} else {
    System.out.println("Valores diferentes");
}
```

# OPERADORES NO JAVA

- Esse código verifica se duas variáveis contêm o mesmo valor e imprime o resultado. Uma vez que as variáveis `valorA` e `valorB` possuem valores diferentes, o trecho de código presente no `else` será executado.

|                 |  |
|-----------------|--|
| <code>==</code> | Utilizado quando desejamos verificar se uma variável é igual a outra.      |
| <code>!=</code> | Utilizado quando desejamos verificar se uma variável é diferente de outra. |

- *Os operadores de igualdade normalmente são utilizados para comparar tipos primitivos (`byte`, `short`, `int`, `long`, `float`, `double`, `boolean` e `char`). No entanto, também podemos utilizá-los para saber se duas instâncias estão apontando para o mesmo objeto.*

# OPERADORES NO JAVA

## OPERADORES RELACIONAIS:

- os operadores relacionais, assim como os de igualdade, avaliam dois operandos. Neste caso, mais precisamente, definem se o operando à esquerda é menor, menor ou igual, maior ou maior ou igual ao da direita, retornando um valor booleano.
- O código a seguir realiza uma série de comparações entre duas variáveis para determinar o que será impresso no console. Uma vez que o valor da variável valor A é menor que valor B serão impressas as mensagens “menor” e “menor ou igual”.

# OPERADORES NO JAVA

```
int valorA = 1;
int valorB = 2;

if (valorA > valorB) {
    System.out.println("maior");
}

if (valorA >= valorB) {
    System.out.println("maior ou igual");
}

if (valorA < valorB) {
    System.out.println("menor");
}

if (valorA <= valorB) {
    System.out.println("menor ou igual");
}
```

# OPERADORES NO JAVA

➤ A tabela abaixo apresenta os operadores relacionais do Java:

|    |  |
|----|--|
| >  | Utilizado quando desejamos verificar se uma variável é maior que outra.        |
| >= | Utilizado quando desejamos verificar se uma variável é maior ou igual a outra  |
| <  | Utilizado quando desejamos verificar se uma variável é menor que outra.        |
| <= | Utilizado quando desejamos verificar se uma variável é menor ou igual a outra. |

# OPERADORES NO JAVA

## OPERADORES LÓGICOS:

- Os operadores lógicos representam o recurso que nos permite criar expressões lógicas maiores a partir da junção de duas ou mais expressões. Para isso, aplicamos as operações lógicas E (representado por “&&”) e OU (representado por “||”).
- Exemplo de uso:

```
if((1 == (2 - 1)) && (2 == (1 + 1))){  
    System.out.println("Ambas as expressões são verdadeiras");  
}
```

Uma vez que utilizamos o operador lógico &&, o System.out.println somente será executado se as duas condições declaradas no if forem verdadeiras.



# OPERADORES NO JAVA

A tabela abaixo apresenta os operadores lógicos do Java:

|    |   |
|----|---|
| && | Utilizado quando desejamos que as duas expressões sejam verdadeiras.          |
|    | Utilizado quando precisamos que pelo menos um das expressões seja verdadeira. |



# OPERADORES NO JAVA

## PRECEDÊNCIA DE OPERADORES:

- Uma vez que os operadores aritméticos buscam reproduzir as operações matemáticas fundamentais, é natural que eles mantenham as suas regras de precedência, que podem ser manipuladas pelo programador com o uso de parênteses.
- Por exemplo, a expressão  $1 + 1 * 2$ , quando analisada pelo compilador, vai retornar o valor 3, porque a multiplicação será resolvida antes da adição. Usando parênteses, a expressão  $(1 + 1) * 2$  retornará o valor 4, pois a adição, por estar dentro dos parênteses, será resolvida primeiro.

# OPERADORES NO JAVA

## Exemplo de uso:

```
if ((1 != (2 - 1)) || (2 == (1+1))) {  
    System.out.println("iguais");  
}
```

