

Para saber mais: operadores



26%

ATIVIDADES
3 DE 15FÓRUM DO
CURSO

No Java temos diversos tipos de operadores para lidar com os dados que estamos trabalhando em nossa aplicação. Vou detalhar melhor alguns deles aqui, que já mencionei nos vídeos:

Operadores de atribuição:

Os operadores de atribuição são usados para atribuir um valor a uma variável. O operador de atribuição básico é o "=" (sinal de igual). Por exemplo:

```
int valor = 5; //Atribui o valor 5 à variável valo
```

[COPIAR CÓDIGO](#)

Existem também operadores de atribuição combinados, que são uma forma abreviada de atribuição. Por exemplo, o operador "+=" adiciona um valor à variável existente. Assim:

```
int valor = 10;  
valor += 15; //Equivalente a valor = valor +
```

[COPIAR CÓDIGO](#)

Operadores aritméticos:

Os operadores aritméticos são usados para realizar operações matemáticas básicas. São eles:



26%

- "+" (adição)
- "-" (subtração)
- "*" (multiplicação)
- "/" (divisão)
- "%" (resto da divisão)

ATIVIDADES
3 DE 15

FÓRUM DO
CURSO

Por exemplo:

```
int a = 10 + 5; // Atribui o valor 15 à variável a
int b = 10 - 5; // Atribui o valor 5 à variável b
int c = 10 * 5; // Atribui o valor 50 à variável c
int d = 10 / 5; // Atribui o valor 2 à variável d
int e = 10 % 3; // Atribui o valor 1 à variável e (o
```

COPIAR CÓDIGO

Operadores relacionais:

Os operadores relacionais são usados para comparar valores. Eles retornam um valor booleano (verdadeiro ou falso). Trabalharemos melhor com eles quando estivermos na aula de condicionais, onde vamos modificar o fluxo da aplicação dada alguma condição. São



- "==" (igual a)

- "!=" (diferente de)
- ">" (maior que)
- ">=" (maior ou igual a)
- "<" (menor que)
- "<=" (menor ou igual a)



26%

Exemplo:

```
int a = 10; // Atribui o valor 10 à variável a
int b = 5;  // Atribui o valor 5 à variável b
int c = 30; // Atribui o valor 30 à variável c
```

```
boolean igual = (b == a); //Nesse caso a variável ig
boolean diferente = (b != c); //A variável diferente
```

COPIAR CÓDIGO

Operadores lógicos:

Esses operadores são usados quando queremos verificar duas ou mais condições e/ou expressões na aplicação. Eles fazem a comparação de valores booleanos e retornam também um resultado booleano.

São três operadores: AND (&&), OR (||) e NOT (!).



O operador AND (&&), que traduzindo para o português seria o E, é usado para verificar se duas condições são verdadeiras. Se ambas as

condições forem verdadeiras, o resultado será verdadeiro. Caso contrário, o resultado será falso. Aqui está um exemplo:

```
boolean a = true;
boolean b = false;
if (a && b) {
    // Este código não será executado, já que a é ver
}
```



26%

COPIAR CÓDIGO

ATIVIDADES
3 DE 15

FÓRUM DO
CURSO

O operador OR (||), que traduzindo para o português seria o OU, é usado para verificar se pelo menos uma das condições é verdadeira. Se pelo menos uma das condições for verdadeira, o resultado será verdadeiro. Caso contrário, o resultado será falso. Aqui está um exemplo:

```
boolean a = true;
boolean b = false;
if (a || b) {
    // Este código será executado, já que a é verdade
}
```

COPIAR CÓDIGO

O operador NOT (!) é usado para negar uma condição. Se a condição for verdadeira, o resultado será falso. Se a condição for falsa, o resultado será verdadeiro. Aqui está um exemplo:



```
boolean a = true;
if (!a) {
    // Este código não será executado, já que a é ver
}
```



26%

COPIAR CÓDIGO

Operadores de incremento:

ATIVIDADES
3 DE 15

FÓRUM DO
CURSO

Além dos operadores citados anteriormente, o operador de incremento é usado para aumentar o valor de uma variável em 1. Existem dois tipos de operadores de incremento: o operador de pré-incremento (++variavel) e o operador de pós-incremento (variavel++).

O operador de pré-incremento (++variavel) aumenta o valor da variável em 1 antes de usar a variável em uma expressão. Aqui está um exemplo:

```
int num = 5;
int resultado = ++num; //num é incrementado para 6 e
System.out.println(num); // imprime 6
System.out.println(resultado); // imprime 6
```

COPIAR CÓDIGO

Já o operador de pós-incremento (variavel++) aumenta o valor da variável em 1 depois de usar a variável em uma expressão. Aqui é um exemplo:



```
int num = 5;  
int resultado = num++; //num é atribuído primeiramente  
System.out.println(num); // imprime 6  
System.out.println(resultado); // imprime 5
```



26%

COPIAR CÓDIGO

ATIVIDADES
3 DE 15

FÓRUM DO
CURSO

