POR CRISTIANE PAGINE

JAVA

TIPOS DE DADOS, OPERADORES E STRINGS

2025

1. Tipos de Dados em Java

Java possui diversos tipos de dados primitivos que representam diferentes categorias de valores. Aqui estão alguns dos principais tipos de dados:

- **int:** Número inteiro de 32 bits, usado para representar valores inteiros (ex: 42).
- **short:** Número inteiro de 16 bits, geralmente usado para economizar memória.
- **long:** Número inteiro de 64 bits, utilizado para valores inteiros muito grandes.
- **float:** Número de ponto flutuante de 32 bits, útil para valores decimais (ex: 3.14).
- **char:** Representa um único caractere de 16 bits, como 'a' ou '5'.

2. Operadores Aritméticos

Abaixo podemos ver os principais operadores aritméticos usados em Java.

Operador	Significado
+	Adição (também mais unário)
–	Subtração (também menos unário)
*	Multiplicação
/	Divisão
%	Módulo
++	Incremento
– –	Decremento

Operador de Adição (+)
Descrição: Realiza a soma de dois valores.
No exemplo abaixo o resultado será 8:

```
int a = 5;
int b = 3;
int soma = a + b;
```

Operador de Subtração (-)

Descrição: Subtrai o valor à direita do valor à esquerda.
No exemplo abaixo o resultado será 6:

```
int a = 10;
int b = 4;
int subtracao = a - b;
```

Operador de Multiplicação (*)

Descrição: Multiplica dois valores.
No exemplo abaixo o resultado será 35:

```
int a = 7;
int b = 5;
int multiplicacao = a * b;
```

Operador de Divisão (/)

Descrição: Divide o valor à esquerda pelo valor à direita. No exemplo abaixo o resultado será 5:

```
int a = 20;
int b = 4;
int divisao = a / b;
```

Operador de Módulo (%)

Descrição: Retorna o resto da divisão do valor à esquerda pelo valor à direita.

No exemplo abaixo o resultado será 3 :

```
int a = 15;
int b = 4;
int modulo = a % b;
```

Operador de Incremento (++)

Descrição: Aumenta o valor da variável em 1.

No exemplo abaixo o resultado será 11:

```
int a = 10;
a++;
```

Operador de Decremento (--)

Descrição: Diminui o valor da variável em 1.

No exemplo abaixo o Resultado será 9:

```
nt a = 10;
a--;
```

3. Operadores Relacionais

Abaixo podemos ver os principais operadores relacionais usados em Java.

Operador	Significado
= =	Igual a
!=	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Igualdade (==)
Descrição: Verifica se dois valores são iguais.
No exemplo abaixo o resultado será true:

```
int a = 5;
int b = 5;
boolean resultado = (a == b);
```

Diferença (!=)
Descrição: Verifica se dois valores são diferentes.
No exemplo abaixo o resultado será true:

```
int a = 5;
int b = 3;
boolean resultado = (a != b);
```

Maior Que (>)
Descrição: Verifica se o valor à esquerda é maior que o valor à direita.
No exemplo abaixo o resultado será true pois 10 é maior que 5:

```
int a = 10;
int b = 5;
boolean resultado = (a > b);
```

Menor Que (<)
Descrição: Verifica se o valor à esquerda é menor que o valor à direita.
No exemplo abaixo o resultado será true pois 3 é menor que 8:

```
int a = 3;
int b = 8;
boolean resultado = (a < b);
```

Maior ou Igual (>=)

Descrição: Verifica se o valor à esquerda é maior ou igual ao valor à direita.

No exemplo abaixo o resultado será true pois 10 é maior ou igual 10:

```
int a = 10;
int b = 10;
boolean resultado = (a >= b);
```

Menor ou Igual (<=)

Descrição: Verifica se o valor à esquerda é menor ou igual ao valor à direita.

No exemplo abaixo o Resultado será true pois 5 é menor ou igual a 10::

```
int a = 10;
int b = 10;
boolean resultado = (a >= b);
```

4. Operadores Lógicos

Abaixo podemos ver os principais operadores lógicos usados em Java.

Operador	Significado
&	AND
|	OR
^	XOR (exclusive OR)
||	OR de curto-circuito
&&	AND de curto-circuito
!	NOT

E lógico (&)
Descrição: Retorna true se ambos os valores forem verdadeiros.
No exemplo abaixo o resultado é falso pois ambos os valores não são iguais:

```
boolean a = true;
boolean b = false;
boolean resultado = a & b;
```

OU lógico (|)

Descrição: Retorna true se pelo menos um dos valores for verdadeiros.

No exemplo abaixo o resultado é true pois um dos valores é verdadeiro:

```
boolean a = true;
boolean b = false;
boolean resultado = a | b;
```

OU exclusivo (XOR) (^)

Descrição: Retorna true se um dos valores for true e o outro false.

No exemplo abaixo o resultado é true pois um dos valores é verdadeiro e o outro é falso:

```
boolean a = true;
boolean b = false;
boolean resultado = a ^ b;
```

E de comparação (&&)

Descrição: Retorna true se ambos os valores forem true, mas para de avaliar quando o primeiro false é encontrado.

No exemplo abaixo o resultado é false pois o primeiro é verdadeiro e o outro é falso:

```
boolean a = true;
boolean b = false;
boolean resultado = a && b;
```

NOT (!)
Descrição: Inverte o valor lógico.
No exemplo abaixo o resultado será falso:

```
boolean a = true;
boolean resultado = !a;
```

OU de comparação(||)
Descrição: Retorna true se pelo menos um dos valores for true, mas para de avaliar quando o primeiro true é encontrado.
No exemplo abaixo o resultado será true pois a é true:

```
boolean a = true;
boolean b = false;
boolean resultado = a || b;
```

5.Strings

Declaração de Strings que iremos usar nos exemplos abaixo:

```
String s = "programa";
String t = "em java";
```

Comprimento de uma String

Descrição: O método .length() retorna o número de caracteres da String.

No exemplo abaixo o resultado será 7 (t contém "em java") :

```
int n = t.length();
```

Substrings

Descrição: O método .substring(início, fim) extrai uma parte da String, de acordo com o índice de início e fim.

No exemplo abaixo o resultado será “pr” (primeiros dois

```
s = s.substring(0, 2);
```

Comparação de Strings

Descrição: Para comparar o conteúdo de duas Strings, usamos o método .equals().

No exemplo abaixo o resultado será true.

Dica:

Use .equals() para comparar o conteúdo das Strings.

Evite usar == para comparação de conteúdo, pois ele compara referências de memória.

```
String s = "casa";
String t = "casa";
boolean resultado = t.equals(s);
```