



**Certified Tech  
Developer**  
The Ultimate Degree

## Estruturas de controle em Java

As estruturas de controle em Java têm a mesma sintaxe que em JavaScript. Das quais são:

Decisão: if - switch.

Repetição: while - for.

Agora, vamos relembrar brevemente a aparência de sua sintaxe.

### Estrutura de Controle

#### Exercício 1

Defina dois números inteiros e atribua um valor a cada um. Verifique se o primeiro número é divisível pelo segundo e indique o resultado obtido por meio de uma mensagem.

**NOTA:** O fato de ser divisível significa que ao dividir precisa dar um valor exato, ou seja, sem decimais.

# Classes Próprias

O núcleo do Java são as classes, mais adiante veremos o que são e como construí-las. Mas para começar a trabalhar, devemos começar a usar as próprias classes do Java.

Em Java, encontramos **tipos primitivos** como uma ferramenta de desenvolvimento e o chamamos de tipos de dados por apenas nos permitirem armazenar um valor. Por exemplo, int, float, double e char. Quando definimos uma variável com esses tipos primitivos, só podemos armazenar valores.

Além dos tipos primitivos, temos as classes próprias do Java que é um elemento que, além de armazenar um valor, nos permite realizar certas operações que já estão programadas. Chamamos esses métodos de operações. Por exemplo, String é uma classe, portanto, é inicializada em maiúsculas. Nomeamos todas as classes com a inicial maiúscula. Se definirmos a seguinte variável:

```
String nome;
```

Veremos que ao usar **nome** nos oferece os métodos disponíveis.

nome.

```
replace(CharSequence charSequ... String  
replaceAll(String s, String s... String  
replaceFirst(String s, String... String  
split(String s) String[]  
split(String s, int i) String[]  
startsWith(String s) boolean  
startsWith(String s, int i) boolean  
substring(int i) String  
substring(int i, int i1) String  
toArray() char[]  
toString() String
```



Estas são funções que já estão resolvidas e que só podemos utilizá-las com a classe a que pertence, ou seja, cada classe em Java possui seus próprios métodos.

Para entender como funcionam as próprias classes da linguagem, vamos nomear 3 classes que são úteis para nós:

- **String**

- **Integer**
- **Float**

As classes **Integer** e **Float** são equivalentes aos tipos de dados primitivos, ou seja, nos permitem armazenar valores dos tipos indicados e também fornecem certas funcionalidades. Costuma-se dizer que eles envolvem os tipos primitivos.

Tenha em mente que quando usamos essas classes não podemos usar operadores como "=", para fazer uma comparação, usamos o `.equals()`. Por exemplo:

`nome.equals("Juan")` retornará verdadeiro se a string "Juan" for salva no nome e falso caso contrário.

O **`equals`** é usado para comparar por igualdade, desde que estejamos trabalhando com classes. Se quisermos comparar se um valor é maior ou menor que outro, devemos usar **`compareTo()`**.

Outra coisa a notar é que uma String à qual não atribuímos nada terá o valor **`null`**. Isso acontece com todas as classes e, se definirmos um elemento (objeto) de uma classe, inicialmente ele terá o valor `null`.



Para organizar as classes, existem os pacotes, onde as classes podem ser agrupadas. Posteriormente, iremos usá-los para nossas classes, mas por enquanto devemos saber que as classes Java também são agrupadas em pacotes, ou como seu nome em inglês: **`package`**.

# String e Integer

## Exercício 1

Defina duas sequências de texto. Verifique se são iguais ou diferentes e indique por meio de mensagem.

## Exercício 2

Defina duas variáveis Integer e determine pelo valor delas qual é maior, qual é menor ou se são iguais. Mostre o resultado obtido.