# Aula 09-String

## Introdução

Olá, meus estudantes, sejam bem-vindos à nossa aula de Algoritmo e Programação em Java do Instituto da Oportunidade Social. Nessa aula, você vai conhecer como trabalhar com strings.

## Strings

Strings é um conjunto de caracteres muito utilizado em diferentes linguagens de programação de diversas formas possíveis. Strings são vetores homogêneos que armazenam dados do tipo char (caracteres), ou seja, uma string pode conter letras, números e caracteres especiais e, dessa forma, possibilitar enviar bloco de dados e não somente palavras ou frases. No contexto de aplicações reais, strings são a forma de enviar informações de uma aplicação para outra. Por exemplo:

String greeting = "Hello";

## Métodos para trabalhar com strings

A linguagem de programação Java possui diversos métodos para manipular strings. O primeiro deles é o método length(), que é usado para retornar o tamanho de vetor ou string.

|  |
| --- |
| String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  System.out.println("O tamanho da string txt é: " + txt.length()); |

A saída esperada para esse trecho de código é:

Text

Description automatically generated

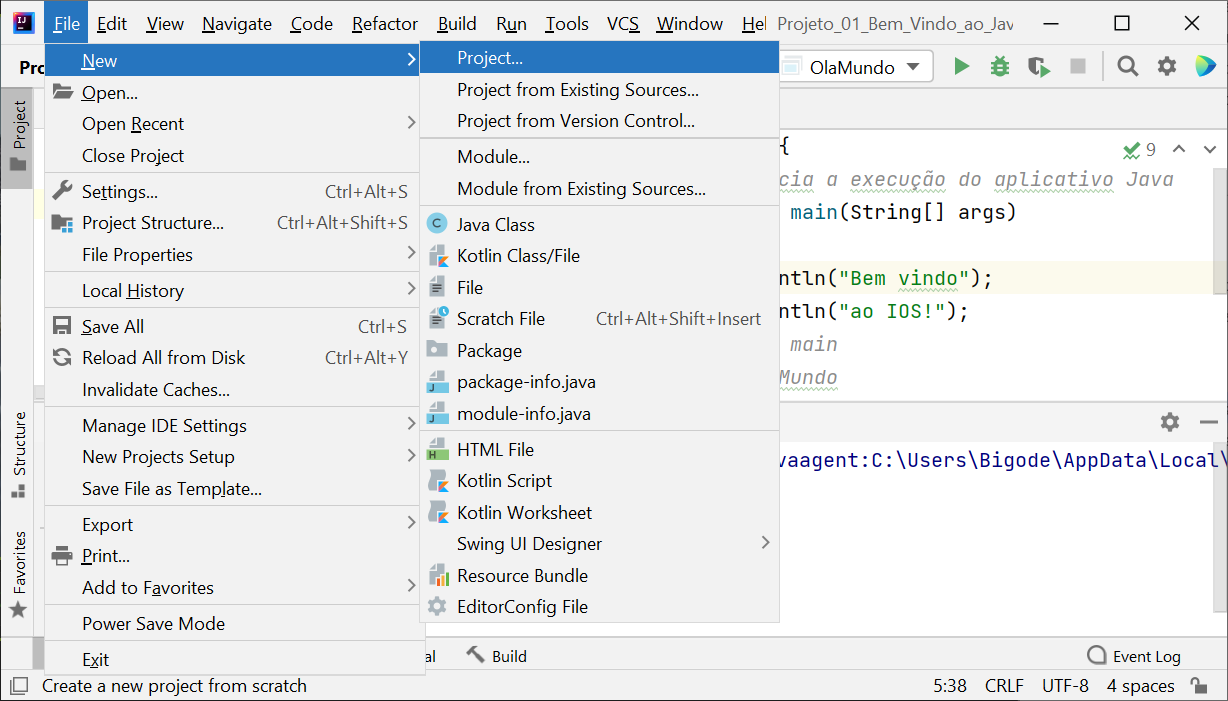
Temos também, os métodos toUpperCase() e toLowerCase(), que são usados para transformar as letras da string para maiúsculas ou para minúsculas, respectivamente.

Também, temos o método indexOf(), que retorna o índice da primeira ocorrência de um texto específico na string (incluindo espaço em branco). Se o método não encontrar o texto, ele retornará -1.

### Vamos praticar

Vamos fazer um programa para trabalhar com os métodos mencionados anteriormente. Siga os passos para criar o projeto:

1. Crie um novo projeto no InterlliJ IDEA. Para isso, acesse o Menu **File**, a opção **New** e depois clique na opção **Project**.



1. As janelas seguintes são as mesmas descritas na criação de um projeto, portanto se tiver alguma dúvida consulte o material didático. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Projeto\_28\_strings
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: **Strings** e insira o código mostrado na Figura 1.
3. **public** **class** Strings {
4. **public** **static** **void** main(String[] args) {
5. String txt = "Ola Pessoal do IOS";
6. // Imprime o tamanho da string
7. System.out.println("O tamanho da string txt é: " + txt.length());
8. // Imprime os caracteres da string em maiúsculo
9. System.out.println(txt.toUpperCase());
10. // Imprime os caracteres da string em minúsculo
11. System.out.println(txt.toLowerCase());
12. // Imprime a posição da primeira ocorrência de um valor na string
13. System.out.println(txt.indexOf("s"));
14. System.out.println(txt.indexOf("IO"));
15. // Não encontra o texto na string
16. System.out.println(txt.indexOf("io")); // Case Sensitive
17. }
18. }

Figura 1 - Código do programa utilizando métodos para manipular strings.

1. Ao executar o código, o programa irá solicitar 10 notas e depois irá imprimir todas as notas digitadas pelo usuário.

Graphical user interface, text

Description automatically generated

Na linha 5 da Figura 1, o método **length()** retorna o tamanho da string, ou seja a quantidade de caracteres da string.

Na linha 7 da Figura 1, o método **toUpperCase()** retorna os caracteres maiúsculos da string.

Na linha 9 da Figura 1, o método **toLowerCase()** retorna os caracteres minúsculos da string.

Nas linhas 11 e 12 da Figura 1, o método **indexOf()** retorna a posição da primeira ocorrência de determinado texto na string. Na linha 14 da Figura 1, o método retorna -1, quando não encontra o padrão na string. Observe que esse método diferencia maiúsculas de minúsculas (Case sensitive).

## Mais métodos para trabalhar com strings

O método **charAt()** retorna o caractere em um índice específico da string.

O método **compareTo()** compara duas strings e retorna:

* 0 se as duas strings forem iguais
* < 0 se a primeira string é menor que a segunda string.
* > 0 se a primeira string é maior que a segunda string

Sintaxe:

Primeira\_string.compareTo(segunda\_string)

Os métodos **startWith()**e **endsWith()** verifica se a string começa ou termina com um valor textual, respectivamente. Esses métodos retornam true ou false.

### Vamos praticar II

Vamos fazer um programa para encontrar o maior valor em um vetor de 5 inteiros. Siga os passos para criar o projeto:

1. Crie um novo projeto no InterlliJ IDEA. Para isso, acesse o Menu **File**, a opção **New** e depois clique na opção **Project**.

Graphical user interface, text, application, email

Description automatically generated

1. As janelas seguintes são as mesmas descritas na criação de um projeto, portanto se tiver alguma dúvida consulte o material didático. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Projeto\_29\_strings
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: **Strings** e insira o código mostrado na Figura 2.
3. **public** **class** Strings {
4. **public** **static** **void** main(String[] args) {
5. String txt = "Ola Pessoal do IOS";
6. String myStr1 = "Hello";
7. String myStr2 = "Hello";
9. **char** result = txt.charAt(0); // Caractere na posição zero
10. System.out.println("Char na posição zero: " + result);
11. result = txt.charAt(10);     // Caractere na posição dez
12. System.out.println("Char na posição dez: " + result);
14. // Compara strings
15. System.out.println("Compara myStr1 com myStr2: " + myStr1.compareTo(myStr2)); // Returna 0
16. System.out.println("Compara txt com myStr2: " + txt.compareTo(myStr2)); // Returna um valor > 0
17. System.out.println("Compara myStr1 com txt: " + myStr1.compareTo(txt)); // Returna um valor < 0
19. // Verifica o ínico e o final de um string
20. System.out.println("Verifica início da string txt: " + txt.startsWith("Ola")); // Retorna true
21. System.out.println("Verifica início da string txt: " + txt.startsWith("O")); // Retorna true
22. System.out.println("Verifica início da string txt: " + txt.startsWith("o")); // Retorna false
24. System.out.println("Verifica final da string txt: " + txt.endsWith("IOS")); // Retorna true
25. System.out.println("Verifica final da string txt: " + txt.endsWith("S")); // Retorna true
26. System.out.println("Verifica final da string txt: " + txt.endsWith("s")); // Retorna false
27. }
28. }

Figura 2 - Código do programa utilizando métodos com strings.

1. Ao executar o programa, ele irá solicitar cinco valores inteiros, depois irá procurar o maior valor digitado e imprimi-lo na tela.

Text

Description automatically generated

Na linha 14 da Figura 2, a instrução retorna um valor maior que zero arbitrário e aleatório. A única coisa que podemos afirmar com certeza é que é maior que zero.

Na linha 15 da Figura 2, a instrução retorna um valor menor que zero arbitrário e aleatório. A única coisa que podemos afirmar com certeza é que é menor que zero.

## Mais alguns métodos para trabalhar com strings

O método **isEmpty()** verifica se a string está vazia e retorna true se a string estiver vazia e false caso contrário.

O método **lastIndexOf**(), que retorna o índice da última ocorrência de um texto específico na string (incluindo espaço em branco). Se o método não encontrar o texto, ele retornará -1.

O método **replace()** retorna uma nova string substituindo um valor desejado pelo outro.

O método **trim()** retira os espaços em branco no início e final de uma string.

### Vamos praticar III

Vamos fazer um programa para somar os valores de um vetor. Siga os passos para criar o projeto:

1. Crie um novo projeto no InterlliJ IDEA. Para isso, acesse o Menu **File**, a opção **New** e depois clique na opção **Project**.

Graphical user interface, text, application, email

Description automatically generated

1. As janelas seguintes são as mesmas descritas na criação de um projeto, portanto se tiver alguma dúvida consulte o material didático. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Projeto\_30\_strings
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: **Strings** e insira o código mostrado na Figura 3.
3. **public** **class** Strings {
4. **public** **static** **void** main(String[] args) {
5. String txt1 = "Ola aa Pessoal aa do aa IOS";
6. String txt2 = "";
7. String txt3 = "       Ola mundo!        ";
9. // Verifica se a string está vazia
10. System.out.println("String txt1 está vazia? " + txt1.isEmpty());
11. System.out.println("String txt2 está vazia? " + txt2.isEmpty());
13. System.out.println("Última posiçõa do aa: " + txt1.lastIndexOf("aa"));
15. // Substituir algo na string
16. String txt4 = txt1.replace('a', 'x');
17. System.out.println(txt4);
19. // Retirar espaços no início e no final
20. System.out.println("String original: " + txt3);
21. System.out.println("String sem espaços: " + txt3.trim());
22. }
23. }

Figura 3 - Código do programa utilizando método para manipular strings.

1. Ao executar o programa, ele irá imprimir a soma de todos os elementos do vetor.

Text

Description automatically generated

## Desafio

1. Faça um programa que receba uma string digitada pelo usuário e retorne:
   * A quantidade de caracteres dessa string.
   * O número de vezes a cada letra aparece, por exemplo:
     + Suponha que o usuário digitou a seguinte string:

Ola pessoal, tudo bem?

* + - Seu programa deverá retornar:

Quantidade de caracteres: 22

Ocorrências:

A – 2

B – 1

C – 0

D – 1

E – 2

F – 0

G – 0

H – 0

I – 0

J – 0

K – 0

L – 2

M – 1

N – 0

O – 3

P – 1

Q – 0

R – 0

S – 2

T – 1

U – 1

V – 0

X – 0

Z – 0

1. Faça o programa contar as ocorrências das letras diferenciando maiúsculas das minúsculas. Por exemplo:
   * + Suponha que o usuário digitou a seguinte string:

Ola pessoal, tudo bem?

* + - Seu programa deverá retornar:

Quantidade de caracteres: 23

Ocorrências:

A – 0 a – 2

B – 0 b – 1

C – 0 c – 0

D – 0 d – 1

E – 0 e – 2

F – 0 f – 0

G – 0 g – 0

H – 0 h – 0

I – 0 i – 0

J – 0 j – 0

K – 0 k – 0

L – 0 l – 2

M – 0 m – 1

N – 0 n – 0

O – 1 o – 2

P – 0 p – 1

Q – 0 q – 0

R – 0 r – 0

S – 0 s – 2

T – 0 t – 1

U – 0 u – 1

V – 0 v – 0

X – 0 x – 0

Z – 0 z – 0