



Técnicas de Programação

Aula 13 – Funções para manipulação de strings



Objetivos de Aprendizagem

- Descrever as funções existentes para manipulação de *strings*.
- Criar aplicações com a utilização de funções que manipulam *strings*.



String

- A linguagem Java não tem um tipo *string* nativo, como outras linguagens.
- Para resolver isso, a biblioteca padrão do Java contém uma classe chamada de *String*.
- Cada *string* entre aspas é uma instancia da classe *String*.

Concatenação

- Como a maior parte das linguagens de programação, a linguagem Java permite que se use o sinal + para unir (concatenar) duas *strings*.


```
String nome = "Max";  
String sobrenome = "Santana";  
String nomeCompleto = nome + sobrenome;  
//nomeCompleto = MaxSantana  
System.out.println("O nome completo é: " +  
nomeCompleto);
```



Substring

- Em Java pode-se extrair uma **substring** de uma *string*, com o método **substring** da classe **String**.
- A linguagem Java conta os caracteres das **strings** de uma maneira peculiar: o primeiro caractere de uma *string* fica na **posição 0**.

```
String saudacao = "Bem-vindo";  
String s = saudacao.substring( 0, 3 ); // s = Bem
```



Tamanho da string

- Para descobrir o tamanho de uma *string*, em Java, basta usar o método ***length*** da classe ***String***.

```
String saudacao = "Olá";  
int n = saudacao.length(); // n = 3
```



Igualdade de string

- A linguagem Java usa o método ***equal***, da classe ***String***, para testar se duas ***strings*** são iguais ou não.
- Para testar se duas strings são idênticas, sem levar em conta diferenciação entre maiúsculas e minúsculas, usa-se o método ***equalsIgnoreCase***.
- Não use o operador **`==`** para testar se duas ***strings*** são iguais! Ele determina somente se as ***strings*** estão ou não armazenadas em determinado local.



Igualdade de string

```
public static void main(String[] args) {  
    String nome0 = "Max Santana";  
    String nome1 = "Max santana";  
    if (nome0.equals(nome1)) // Nomes diferentes  
        System.out.println("Nomes iguais.");  
    else  
        System.out.println("Nomes diferentes.");  
    if (nome0.equalsIgnoreCase(nome1)) // Nomes iguais  
        System.out.println("Nomes iguais.");  
    else  
        System.out.println("Nomes diferentes.");  
    if (nome0 == nome1) // Nomes diferentes  
        System.out.println("Nomes iguais.");  
    else  
        System.out.println("Nomes diferentes.");  
}
```


Métodos da classe String

Métodos	Descrição
<code>char charAt(int índice)</code>	Retorna o caractere da posição especificada.
<code>int compareTo(String outra)</code>	Retorna um valor negativo se a string vier antes de <i>outra</i> na ordem do dicionário, um valor positivo se a string vier depois de <i>outra</i> na ordem do dicionário, ou 0 se as strings forem iguais.
<code>boolean endsWith(String sufixo)</code>	Retorna <i>true</i> se a string termina com <i>sufixo</i> .
<code>boolean equals(Object outra)</code>	Retorna <i>true</i> se a string for igual a <i>outra</i> .
<code>boolean equalsIgnoreCase(String outra)</code>	Retorna <i>true</i> se a string for igual a <i>outra</i> , não diferenciando maiúsculas de minúsculas.
<code>int indexOf(String str)</code>	Retorna o início da primeira substring igual a <i>str</i> , começando no índice 0 ou <i>indiceInicio</i> .
<code>int indexOf(String str, int indiceInicio)</code>	
<code>int lastIndexOf(String str)</code>	Retorna o início da última substring igual a <i>str</i> , começando no índice 0 ou <i>indiceInicio</i> .

Métodos da classe String

Métodos	Descrição
<code>int lastIndexOf(string str, int indiceInicio)</code>	Retorna o início da última substring igual a <i>str</i> , começando no índice 0 ou <i>indiceInicio</i>
<code>int length()</code>	Retorna o comprimento da string.
<code>String replace(char charAnt, char charNovo)</code>	Retorna uma nova string que é obtida substituindo-se todos os caracteres de <i>charAnt</i> na string por <i>charNovo</i> .
<code>boolean startsWith(String prefixo)</code>	Retorna <i>true</i> se a string começar com <i>prefixo</i> .
<code>String substring(int indiceInicio)</code>	Retorna uma nova string com todos os caracteres a partir de <i>indiceInicio</i> até o final da string ou até <i>indiceFim</i> .
<code>String substring(int indiceInicio, indiceFim)</code>	

Métodos da classe String

Métodos	Descrição
<code>String toLowerCase()</code>	Retorna uma nova string contendo todos os caracteres da string original, com as maiúsculas convertidas para minúsculas.
<code>String toUpperCase()</code>	Retorna uma nova string contendo todos os caracteres da string original, com as minúsculas convertidas para maiúsculas.
<code>String trim()</code>	Retorna uma nova string eliminando todos os espaços iniciais e finais da string original.



Exercício

- Faça um programa que solicita o usuário digitar uma mensagem (string).
- Em seguida o programa converte todos os caracteres da string para maiúsculo e depois imprime os resultados.



Exercício

- Faça um programa que solicita o usuário digitar o nome e sobrenome
- Em seguida o programa solicita o usuário digitar rua, numero, bairro, cidade (capturando todos os dados como string).
- Finalmente o programa concatena o nome e sobrenome e mostra na tela.
- Depois o programa concatena os dados do endereço e imprime o endereço de uma só vez.