

Manipulação de texto

Prof. Me. Hélio Esperidião

Métodos de Manipulação de Strings.

São métodos capazes de gerar operações com strings fazendo com que os programas fiquem mais avançados e flexíveis.

Sintaxe geral dos métodos.

String <variável> = <String>.<método>(<parametros>);

Ou

int/boolean <variável> = <String>.<método>(<parametros>);

`equals()` - Retorna verdadeiro ou falso mediante duas strings sendo comparadas.

Exemplo :

```
String a = "Abacaxi";  
boolean resul = a.equals("Morando");
```

O valor booleano retornado para result será falso. (false) as strings não são iguais.

`equalsIgnoreCase()`:
compara se duas string são iguais independente de serem
maiúsculas ou minúsculas.

Exemplo :

```
String nome = "maria";
```

```
String nome2 = "MARIA";
```

```
if (nome.equalsIgnoreCase(nome2) ) System.out.println("Iguais");  
else System.out.println("Diferenetes");
```

`startsWith()`: Retorna verdadeiro se a cadeia ou um caractere estiver no início da string.

Exemplo :

```
String S = "Tchak estava na cozinha ....";  
boolean bld = S.startsWith("Tchak");  
System.out.println(bld);
```

`endsWith()`: Retorna verdadeiro se a cadeia ou um caractere estiver no fim da string.

Exemplo :

```
String nome = "maria";  
boolean bl2 = nome.endsWith("a");  
System.out.println(bl2);
```

isDigit() - Retorna verdadeiro se caracter passado como argumento for um número.

Exemplo :

```
boolean d = Character.isDigit('1');
```

Retorna true para a variável d.

Obs : Um char em Java deve ser declarado com apóstrofes (') e não aspas (").

`charAt()` - Retorna com um caracter da string mediante a informação da posição desse elemento.

Exemplo :

```
String palavra = "Morango";  
char y = palavra.charAt(2);
```

Retorna o caracter **r** que está na posição 2 da string, considerando a partir do zero.

replace() - Substitui na string um determinado caracter, mediante a informação do mesmo.

Exemplo :

```
String a = "Marta".replace('a','o');
```

Retorna com a String Morto para a variável a.

length() - Retorna o numero de elementos de uma string.

Exemplo :

```
String texto = "Laranja";  
int x = texto.length();
```

Retorna com o número 7 que vem a ser o número de caracteres da palavra "Laranja".

Obs : Se for um vetor o método length deve ser escrito sem os simbolos de ().

substring() - Retorna com uma parte da string mediante ao argumento numérico dado.

Exemplo :

```
String s = "Morango";  
String l = s.substring(3,6);
```

Irá retornar a string **ang** retirando os três primeiro caracteres e considerando os demais até a sexta posição.

toUpperCase() - converte um string de minúscula para maiúscula.

Exemplo :

```
String s = "morango";
```

```
String teste = s.toUpperCase();
```

O resultado da variável teste será
MORANGO.

toLowerCase() - converte um string de maiúscula para minúscula.

Exemplo :

```
String s = "MORANGO";
```

```
String teste = s.toLowerCase();
```

O resultado da variável teste será morango.

trim() - Retira os espaços em branco de uma String.

Exemplo :

```
String b = "    Lixo    ";
```

```
String x = b.trim();
```

O resultado para a variável x será "Lixo".

`indexOf()` - Retornar em que posição encontra-se a primeira ocorrência de uma string em uma outra string. Caso não encontre a string procurado ele irá retornar -1.

Exemplo :

```
String lista = "Constiuição";  
int x = lista.indexOf("s");
```

O resultado será três tomando como base de referencia a posição (0).

lastIndexOf() - Retorna em que posição encontra-se a ultima ocorrência de uma string em uma outra string.

Exemplo :

```
int y = "Constiuição".lastIndexOf("i");
```

Ir  retorna 7 que   o n mero da posi  o da ultima ocorr ncia da letra "i" na String.