

FIA/P GRADUAÇÃO

DOMAIN DRIVEN DESIGN

Prof. Me. Thiago T. I. Yamamoto

#08 - MANIPULAÇÃO DE STRINGS

TRAJETÓRIA



- ✓ Orientação a Objetos
- ✓ Introdução ao Java
- ✓ IDE e Tipos de Dados
- ✓ Classes, atributos e métodos
- ✓ Encapsulamento
- ✓ Construtores
- ✓ Conversões e Tomada de decisões
- ✓ Manipulação de Strings

#08 - AGENDA



- Strings
- Inicialização de Strings
- Sequencia de Escape
- Conversões para String
- Concatenação
- Métodos



STRINGS

- A manipulação de **string** é essencial para fazer **validações** de dados de **entrada**, **exibir** informações para o usuário e outras informações baseadas em texto;
- Em **Java** uma **string** é um objeto da classe **String**;
- A string precisa ser **declarada, instanciada e inicializada**;
- Em Java, as string são **imutáveis**, após criar uma string é **impossível** alterar o seu valor;



- É possível inicializar uma string de várias maneiras distintas:

```
String endereco = ""; //string vazia
```

```
String endereco = new String();
```

```
endereco = "Av. Paulista";
```

```
String endereco = "Av. Paulista";
```

```
String endereco = new String("Av. Paulista");
```

- Se uma **string** não for instanciada seu valor será *null*;
- É preciso instanciar uma **string** para poder utilizar qualquer método dela ou a exceção *NullPointerException* poderá ser lançado;

- Exemplos:

- Correto:

```
String endereco = new String();
```

```
int tamanhoDaString = endereco.length();
```

- Não instanciou , o valor da string continua *null*:

```
String endereco = null;
```

```
int tamanhoDaString = endereco.length();
```

- Não compila:

```
String endereco;
```

```
int tamanhoDaString = endereco.length();
```


- Alguns **caracteres específicos** quando precedidos pela **contra barra** (\) são considerados **sequência de escape** e possuem um **significado especial** para o compilador;
- No **Java** as sequências de escape mais utilizadas são:

Sequência de Escape	Descrição
\t	Move o cursor para a próxima posição da tabulação horizontal
\n	Move o cursor para o começo da próxima linha
\'	Gera a saída do caractere de aspa simples (')
\"	Gera a saída do caractere de aspas duplas (")
\\	Gera a saída do caractere de barra invertida (\)

- Para utilizar uma sequência de escape basta coloca-la numa string;

Exemplo:

```
System.out.println("Eu disse \"Teste\"!");
```

Saída para o console:

Eu disse "Teste"!



- É possível criar uma **String** a partir da **concatenação** (ligação, junção) do **conteúdo de duas ou mais strings**;
- A maneira mais fácil de **concatenar** string é através do operador **+**

```
String endereco = "Av. Paulista";  
String numero = "478";  
String enderecoCompleto = endereco + ", " + numero;  
System.out.println(enderecoCompleto);
```

Saída para o console:

Av. Paulista, 478



- O **operador +=** (atribuição aditiva) também pode ser utilizado para realizar a **concatenação de strings**;

```
String endereco = "Av. Paulista";  
String numero = "478";  
String enderecoCompleto = endereco;  
enderecoCompleto += ", ";  
enderecoCompleto += numero;  
System.out.println(enderecoCompleto);
```

Saída para o console:

Av. Paulista, 478

- Também é possível utilizar o método **concat** para realizar a **concatenação** entre **strings**;

```
String endereco = "Av. Paulista";  
String numero = "478";  
String enderecoCompleto =  
endereco.concat(", ").concat(numero);  
System.out.println(enderecoCompleto);
```

Saída para o console:

Av. Paulista, 478

- A classe **String** fornece um vasto conjunto de métodos para a sua manipulação:
 - Métodos para **comparação**:
 - **equals, equalsIgnoreCase, startsWith, endsWith;**
 - Métodos para **acesso**:
 - **length, charAt;**
 - Métodos para **pesquisa**:
 - **indexOf, lastIndexOf;**
 - Métodos para “**alteração**”:
 - **substring, toLowerCase, toUpperCase, replace;**

- A comparação entre o conteúdo de duas strings deve ser realizada através de métodos;
- Os principais métodos para comparação de strings são:
 - `equals(String)`
 - `equalsIgnoreCase(String)`
- Nunca utilize o operador `==` para verificar se uma **string** é igual a outra;

- O método `equals` compara duas strings quanto ao conteúdo, diferenciando caracteres maiúsculos de minúsculos;

```
String textoA = "fiap";
String textoB = "FIAP";

if (textoA.equals(textoB))
    System.out.println("#6 O texto (" + textoA + ") eh
    igual ao texto (" + textoB + ")");
else
    System.out.println("#6 O texto (" + textoA + ") NAO      eh
    igual ao texto (" + textoB + ")");
```

Saída para o console:

#6 O texto (fiap) NAO eh igual ao texto (FIAP)

- O método `equalsIgnoreCase` compara duas strings quanto ao conteúdo, não diferenciando caracteres maiúsculos de minúsculos;

```
String textoA = "fiap";  
String textoB = "FIAP";
```

```
if (textoA.equalsIgnoreCase(textoB))    System.out.println("#7 O  
texto (" + textoA + ")    eh igual ao texto (" + textoB + ")");  
else  
    System.out.println("#7 O texto (" + textoA + ")    NAO eh  
igual ao texto (" + textoB + ")");
```

Saída para o console:
#7 O texto (fiap) eh igual ao texto (FIAP)

- O método **startsWith** verifica se uma **string** começa por uma determinada palavra (conjunto de caracteres). Este **método** recebe como parâmetro a palavra a ser procurada;

```
String textoD = "tenha paciencia, existem muitos metodos";
if(textoD.startsWith("tenha"))
    System.out.println("#8 O texto (" + textoD + ") começa com (tenha)");
else
    System.out.println("#8 O texto (" + textoD + ") NAO começa com (tenha)");
```

Saída para o console:

#8 O texto (tenha paciencia, existem muitos metodos) começa com (tenha)

- O método **endsWith** verifica se uma **string** termina com uma determinada palavra. Este método recebe como **parâmetro** a palavra a ser procurada;

```
String textoD = "tenha paciencia, existem muitos metodos";
if(textoD.endsWith("dos"))
    System.out.println("#9 O texto (" + textoD + ") termina com (dos)");
else
    System.out.println("#9 O texto (" + textoD + ") NAO termina com (dos)");
```

Saída para o console:

#9 O texto (tenha paciencia, existem muitos metodos) termina com (dos)

- Para obter o **tamanho**, ou melhor, o **número de caracteres** em uma string, o método **length** é utilizado;

```
String nomeCompleto = "Paulo Roberto de Souza";
```

```
System.out.println("A string (" + nomeCompleto + ") contem " +  
nomeCompleto.length() + " letras");
```

Saída para o console:

A string (Paulo Roberto de Souza) contem 22 letras

- Para obter um caractere da string é utilizado o método `charAt`;
- Este método recebe como **parâmetro** o **índice** (posição) do caractere na string;

```
String nomeCompleto = "Paulo Roberto de Souza";
```

```
char terceiroChar = nomeCompleto.charAt(2);
```

```
System.out.println("O terceiro caractere da string (" + nomeCompleto  
+ ") eh: " + terceiroChar);
```

Saída para o console:

O terceiro caractere da string (Paulo Roberto de Souza) eh: u

- Para obter o índice da primeira ocorrência de um caractere na **string** é utilizado método **indexOf**;
- Este método recebe como **parâmetro** o **caractere** a ser localizado;

```
String nomeCompleto = "Paulo Roberto de Souza";
```

```
System.out.println("A letra 'a' aparece primeiro na posicao (" +  
nomeCompleto.indexOf('a') + ") da string (" + nomeCompleto + ")");
```

Saída para o console:

A letra 'a' aparece primeiro na posicao (1) da string (Paulo Roberto de Souza)

- Para obter o índice da última ocorrência de um caractere na string é utilizado método `lastIndexOf`;
- Este método recebe como **parâmetro** o caractere a ser localizado;

```
String nomeCompleto = "Paulo Roberto de Souza";
```

```
System.out.println("A letra 'u' aparece por ultimo na posicao (" +  
nomeCompleto.lastIndexOf('u') + ") da string (" + nomeCompleto  
+ ")");
```

Saída para o console:

A letra 'u' aparece por ultimo na posicao (19) da string (Paulo Roberto de Souza)

- É possível **criar uma string** a partir de um **trecho de outra string**, para isto o método **substring** é utilizado;
- Este método recebe como **parâmetros a posição inicial (inclui)** e a **posição final (exclui)** do conjunto de caracteres a serem copiados da string original;

```
String stringE = "um; dois; tres; quatro";  
String numeroDois = stringE.substring(4, 8);  
  
System.out.println("Numero 2: " + numeroDois);
```

Saída para o console:
Numero 2: dois

- Também é possível utilizar o método **substring** passando **somente** a **posição inicial(inclusive)** do conjunto de caracteres a serem copiados da string original;

```
String stringE = "um; dois; tres; quatro";  
String numeroQuatro = stringE.substring(16);  
  
System.out.println("Numero 4: " + numeroQuatro);
```

Saída para o console:
Numero 4: quatro

- Para converter os caracteres de uma string para maiúsculo é utilizado o método `toUpperCase`;

```
String nomeCompleto = "Paulo Roberto de Souza";  
String nomeMaiusculo = nomeCompleto.toUpperCase();
```

```
System.out.println("A string (" + nomeCompleto + ") em maiusculo fica  
(" + nomeMaiusculo + ")");
```

Saída para o console:

A string (Paulo Roberto de Souza) em maiusculo fica(PAULO ROBERTO DE SOUZA)

- Para converter os caracteres de uma string para minúsculo é utilizado o método `toLowerCase`;

```
String nomeCompleto = "Paulo Roberto de Souza";
```

```
System.out.println("A string (" + nomeCompleto + ") em minuscuro  
fica (" + nomeCompleto.toLowerCase() + ")");
```

Saída para o console:

A string (Paulo Roberto de Souza) em minuscuro fica (paulo roberto de souza)

- Para **substituir** caracteres de uma string é utilizado o método **replace**;

```
String nomeCompleto = "Paulo Roberto de Souza";  
String nomeCerto = nomeCompleto.replace('o', 'a');
```

```
System.out.println("O nome (" + nomeCompleto + ") foi corrigido para  
(" + nomeCerto + ")");
```

Saída para o console:

O nome (Paulo Roberto de Souza) foi corrigido para (Paula Raberta de Sauza)

- Para substituir **palavras**, conjunto de letras, de uma string é utilizado o método **replace**;

```
String nomeCompleto = "Paulo Roberto de Souza";  
String nomeCerto = nomeCompleto.replace("o ", "a ");
```

```
System.out.println("O nome (" + nomeCompleto + ") foi corrigido para  
(" + nomeCerto + ")");
```

Saída para o console:

O nome (Paulo Roberto de Souza) foi corrigido para (Paula Roberta de Souza)

- No projeto chamado **ManipulaString**, crie uma classe de teste chamada **Exemplos**, dentro do método `main()` digite:

```
String nomeCompleto = "Braufagélío Zicunstrumbi da Silva";  
char terceiroChar = nomeCompleto.charAt(2);  
System.out.println("O terceiro caractere da string (" + nomeCompleto + ") é:  
" + terceiroChar);  
System.out.println("Parte: " + nomeCompleto.substring(6,11));  
System.out.println("Final: " + nomeCompleto.substring(12));  
  
String textoA = "alface";  
String textoB = "ALFACE";
```

```

if(textoA.equals(textoB)){
    System.out.println("O texto ("+textoA+") é igual ao texto ("+textoB+");
} else {
    System.out.println("O texto ("+textoA+") não é igual ao texto ("+textoB+");
} if(textoA.equalsIgnoreCase(textoB)){
    System.out.println("O texto ("+textoA+") é igual ao texto ("+textoB+");
} else {
    System.out.println("O texto ("+textoA+") não é igual ao texto ("+textoB+");
}
    
```

```
String textoC = "A alface têm vitamina que engorda e faz crescer";
if(textoC.startsWith("A")){
    System.out.println("O texto (" + textoC + ") começa com (A)");
}
if(textoC.endsWith("cer")){
    System.out.println("O texto (" + textoC + ") termina com (cer)");
} else {
    System.out.println("O texto (" + textoC + ") NAO termina com (cer)");
}
```

```
System.out.println("A letra 'a' aparece primeiro na posicao
(" + nomeCompleto.indexOf('a') + ") da string (" + nomeCompleto + ")");
System.out.println("A letra 'u' aparece por ultimo na posicao
(" + nomeCompleto.lastIndexOf('u') + ") da string (" + nomeCompleto + ")");
```


PRÁTICA



1. Faça uma classe teste chamada **Maiuscula** que solicite ao usuário uma palavra e em seguida a exiba em maiúsculo. Por exemplo, se o usuário digitar “cadeira”, o programa deverá retornar “CADEIRA”.

2. Crie uma classe teste chamada **Vogal** que solicite ao usuário uma String e em seguida exiba o conteúdo desta String de forma que todas as vogais (a,e,i,o,u) sejam representadas por * (asterisco). Por exemplo, se o usuário digitar “cadeira”, o programa deverá exibir “c*d**r”.



Copyright © 2020 - 2025 Prof. Thiago T. I. Yamamoto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

*"A lógica pode levar de um ponto A a um ponto B.
A imaginação pode levar a qualquer lugar"*
Albert Einstein