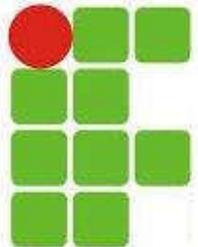

Programação Orientada a Objetos

Professor: Diego Oliveira



**Conteúdo 18:
Strings**





Strings

- Strings são objetos/instâncias da classe `java.lang.String`
- String não é um tipo primitivo do Java, como `int`, `double` ou `boolean`
- Elas contêm sequências de caracteres e são utilizadas para armazenar textos
- Strings são bastante utilizadas no Java, por isso estudaremos alguns métodos importantes desta classe





Strings

- Uma String pode ser inicializada com a palavra-chave 'new':

```
String senha = new String("12345");
```

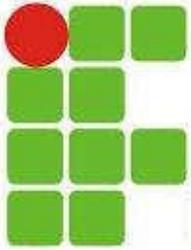
- Ou pode ser inicializada diretamente:

```
String senha = "12345";
```

- Além de poder ser declarada e ter um valor inserido posteriormente:

```
String senha;  
senha = "12345";
```





Strings

- Strings podem ser concatenadas com o operador '+':

```
String nome = "Diego";  
nome = nome + " Oliveira";  
System.out.println(nome);
```

```
Output - POO-Exercicio14 (run)  
run:  
Diego Oliveira  
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Ou com o método concat:

```
String nome1 = "Diego";  
String nome2 = "Oliveira";  
System.out.println(nome1.concat(nome2));
```

```
Output - POO-Exercicio14 (run)  
run:  
DiegoOliveira
```



Strings

- Vejamos alguns métodos interessantes da classe String:
 - Remover espaços antes e depois de um texto:

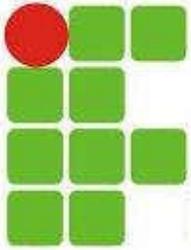
```
String nome = "    Diego Oliveira    ";  
System.out.println(nome.trim());
```

- Substituir partes do texto:

```
String ifrn = "IFRN";  
ifrn = ifrn.replace("IF", "Instituto Federal");  
ifrn = ifrn.replace("RN", " Rio Grande do Norte");  
System.out.println(ifrn);
```



```
Output - POO-Exercicio14 (run)  
run:  
Instituto Federal Rio Grande do Norte
```



Strings

- Vejamos alguns métodos interessantes da classe String:

- Recortar uma parte de uma String:

```
Output - POO-Exercicio14 (run)
run:
Federal
```

```
String ifrn = "Instituto Federal do Rio Grande do Norte";
System.out.println(ifrn.substring(10, 17));
```

- Tamanho de uma String:

```
String ifrn = "Instituto Federal do Rio Grande do Norte";
System.out.println(ifrn.length());
```

```
Output - POO-Exercicio14 (run)
run:
40
```





Strings

- Vejamos alguns métodos interessantes da classe String:
 - Comparando Strings:

```
String nome1 = "Diego Oliveira";  
String nome2 = "diego oliveira";  
System.out.println(nome1.equals(nome2));  
System.out.println(nome1.equalsIgnoreCase(nome2));
```

```
Output - POO-Exercicio14 (run)  
run:  
false  
true
```





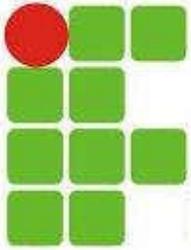
Strings

- Vejamos alguns métodos interessantes da classe String:
 - Dividindo Strings em partes:

```
String ifrn = "Instituto Federal do Rio Grande do Norte";  
String partes[] = ifrn.split(" ");  
for(int i=0; i<partes.length; i++){  
    System.out.println(partes[i]);  
}
```

```
Output - POO-Exercicio14 (run)  
run:  
Instituto  
Federal  
do  
Rio  
Grande  
do  
Norte
```





Strings

- Vejamos alguns métodos interessantes da classe String:
 - Verificando se a String contém uma palavra:

```
String texto = "Instituto Federal do Rio Grande Do Norte";  
System.out.println(texto.contains("Federal"));
```

```
Output - POO-Exercicio14 (run)  
run:  
true
```

- Verificando a posição de uma palavra no texto:

```
String texto = "Instituto Federal do Rio Grande Do Norte";  
System.out.println(texto.indexOf("Federal"));
```

```
Output - POO-Exercicio14 (run)  
run:  
10
```





Strings

- Vejamos alguns métodos interessantes da classe String:
 - Verificando início e fim de uma String:

```
String texto = "Instituto Federal do Rio Grande Do Norte";  
System.out.println(texto.startsWith("Inst"));  
System.out.println(texto.endsWith("rte"));
```

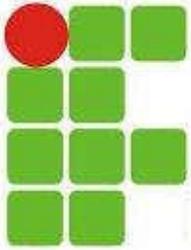
```
Output - POO-Exercicio14 (run)  
run:  
true  
true
```

- Pegando o último caractere:

```
String texto = "Instituto Federal do Rio Grande Do Norte";  
System.out.println(texto.charAt(texto.length()-1));
```

```
Output - POO-Exercicio14 (run)  
run:  
e
```

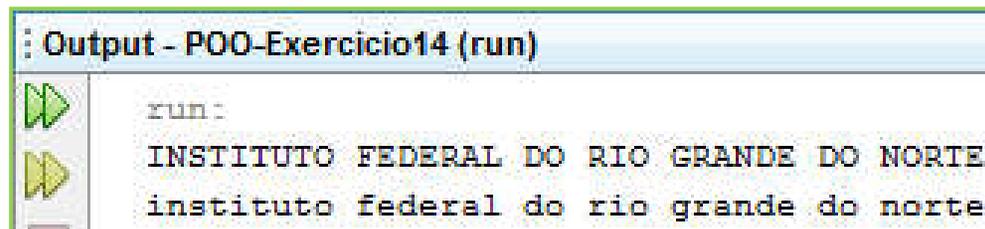




Strings

- Vejamos alguns métodos interessantes da classe String:
 - Passando um texto para maiúsculo e minúsculo:

```
String texto = "Instituto Federal do Rio Grande Do Norte";  
System.out.println(texto.toUpperCase());  
System.out.println(texto.toLowerCase());
```



```
: Output - POO-Exercicio14 (run)  
run:  
INSTITUTO FEDERAL DO RIO GRANDE DO NORTE  
instituto federal do rio grande do norte
```





Strings

- Cada caractere é representado por um número inteiro no Java. Vejamos a impressão de todas as letras minúsculas e maiúsculas:

```
System.out.println("Caracteres alfabéticos maiúsculos:");  
for (int i = 65; i <= 90; i++) {  
    System.out.printf("%c", i);  
}  
System.out.println("\nCaracteres alfabéticos minúsculos:");  
for (int i = 97; i <= 122; i++) {  
    System.out.printf("%c", i);  
}
```

```
Output - PGO-Exercicio14 (run)  
run:  
Caracteres alfabéticos maiúsculos:  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
Caracteres alfabéticos minúsculos:  
abcdefghijklmnopqrstuvwxyzAAA
```



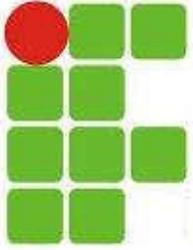


Exercício

- Criptografia: faremos um programinha para esconder informações
- Desenvolva uma aplicação que troque cada letra de um texto por outra do alfabeto X posições a frente, onde X pode ser escolhido por você (cifra de César)
- Exemplo com X=2:
 - **“DIEGO”** passará a ser **“FKGIQ”**
- Para fazer este exercício converta cada caractere para inteiro e depois de volta para caractere:

```
for ( FAÇA SEU FOR ) {  
    int letra = (int) nome.MÉTODO?;  
    LÓGICA?  
    nome2 += (char) letra;  
}
```





Perguntas?

