



# UML e Desenvolvimento Java

Strings  
Etapa 4

Gustavo de Miranda Gonçalves  
gustavo.miranda@prof.infnet.edu.br

---

*Strings*

---



## Manipulando Strings

Uma string é uma sequência de caracteres tratada como um elemento único. Ela pode conter letras, dígitos e caracteres especiais. Uma string é um objeto da classe String, e deve ser escrita como uma sequência de caracteres entre aspas duplas.

## Função de Comparação: equals X equalsIgnoreCase

Esse método é declarado na classe Object, e testa dois objetos quaisquer quanto à igualdade. Na classe String, ele foi sobrescrito e compara se os valores das strings são idênticos.

O método **equals** retorna **true** se os conteúdos dos objetos forem iguais, ou **false**, se forem diferentes.

Considere que maiúsculas e minúsculas são diferentes, ou seja, “Infnet” é diferente de “INFNET”. Uma opção que retornaria **true** neste caso seria o método **equalsIgnoreCase**, que ignora as diferenças entre maiúsculas e minúsculas.



## Por que o operador “==” não funciona?

Esse operador serve para comparar tipos primitivos e retorna **true** se os valores forem iguais. No entanto, quando objetos são comparados com o operador de igualdade, ele retornará **true** somente se as referências à memória forem iguais.

Observe no código abaixo, que a instrução da linha 5 deverá imprimir **false**, enquanto que a da linha 6 vai imprimir **true**. Fique atento sobre este caso, pois um dos erros comuns em programação ocorre quando referências são comparadas com o operador de igualdade, quando o desejado era comparar os conteúdos.

```
public class ComparacaoDeStrings {  
    public static void main(String[] args) {  
        String string1 = new String("Sou Infnet!");  
        String string2 = new String("Sou Infnet!");  
        System.out.println(string1 == string2);  
        System.out.println(string1.equals(string2));  
    }  
}
```



## Método: compareTo

Ao comparar strings, o método **compareTo** também pode ser utilizado. Ele retorna 0 se as strings forem iguais, um número negativo se a string que chama o método for menor que a string que é passada como parâmetro, e um número positivo se a string que invoca o método for maior que o parâmetro.



## Métodos: startsWith X endsWith

Outros métodos muito úteis são: **startsWith** e **endsWith**. O método **startsWith** testa se uma string inicia com a string passada como parâmetro. Da mesma forma, o método **endsWith** testa se ela termina com o parâmetro informado. Ambas retornam true ou false.



## **Métodos:** indexOf X lastIndexOf

É muito útil pesquisar em uma string um caractere ou conjunto de caracteres.

Os métodos **indexOf** e **lastIndexOf** realizam esta tarefa para você.

O método **indexOf** realiza uma busca em determinada string por um caractere específico ou uma substring, e retorna sua posição de sua primeira ocorrência.

O método **lastIndexOf** retorna a posição da última ocorrência de um caractere ou substring em uma string.

## Métodos: indexOf X lastIndexOf

```
public class ComparacaoDeStrings {  
    public static void main(String[] args) {  
        String string = new String("Infnet!");  
        System.out.println(string.indexOf('n'));  
        System.out.println(string.lastIndexOf('n'));  
    }  
}
```

Observe a diferença entre os métodos no código abaixo. Note que serão impressos os valores 1 para a chamada de `indexOf` e 3 para `lastIndexOf`.

**Método:** substring

```
public class ComparacaoDeStrings {  
    public static void main(String[] args) {  
        String string1 = new String("Infnet");  
        System.out.println(string1.substring(2, 4));  
    }  
}
```

O método substring possui duas variações, ou seja, uma sobrecarga de métodos.

O primeiro recebe como parâmetro o índice inicial da string a partir da qual os caracteres devem ser copiados. O segundo recebe os índices inicial e final, sendo que o final é até ele, mas sem incluí-lo. Observe no código a seguir, que será impresso na tela o valor “fn”.



## Outros métodos

- concat
- trim
- replaceAll
- replaceFirst
- split
- toUpperCase
- toLowerCase