

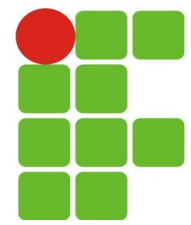
String em Java

João Paulo Q. dos Santos
joao.queiroz@ifrn.edu.br



Java

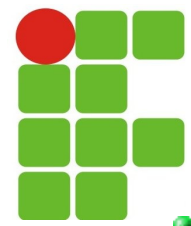




Roteiro

- Conhecendo as classes:
 - Character;
 - String;
 - StringBuilder;
 - StringBuffer.





Classe Character

- Classe que representa um único caractere:
 - método `charValue()` retorna o caractere desse objeto.

```
public class Caractere {
    public static void main(String[] args) {

        Character c1 = null, c2 = null;

        System.out.println("O valor de c1 é "+c1);

        c1 = new Character('a');

        System.out.println("O valor de c1 é "+c1.charValue());

        c2 = c1;

        if (c1 == c2){
            System.out.println("c1 e c2 são o mesmo (1)");
        }

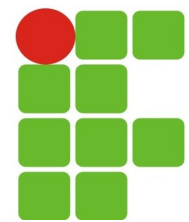
        if (c1.equals(c2)){
            System.out.println("c1 e c2 são iguais (2)");
        }

        c2 = new Character('a');

        if (c1 == c2){
            System.out.println("c1 e c2 são o mesmo (3)");
        }

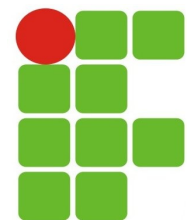
        if (c1.equals(c2)){
            System.out.println("c1 e c2 são iguais (4)");
        }

    }
}
```



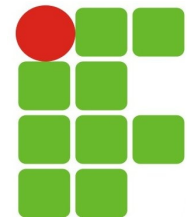
Classe String

- Três classes para Strings:
 - String:
 - Objetos são **imutáveis**;
 - Possuem tratamento especial;
 - Métodos de manipulação.
 - StringBuffer:
 - Objetos **mutáveis**;
 - Métodos de manipulação;
 - Garante Sincronização;
 - Pode ser utilizada em programação concorrente.
 - StringBuilder:
 - Mesmo que StringBuffer;
 - Não garante sincronização;
 - Não é recomendado o seu uso quando existe concorrência.



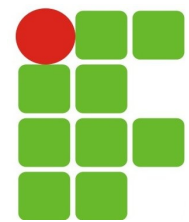
Classe String

- Operador **new**
 - `String s = new String("Java");`
 - `String s1 = new String ("É Muito Legal");`
- São indexados a partir do zero
 - "IFRN" usa os índices 0, 1, 2, 3;
- Pode ser concatenada com o operador +
 - Tratamento especial para String (**Não é necessário utilizar o operador new para criar um objeto String**);
 - `String s3 = s1 + s2;`
- Texto entre aspas são objetos da classe String
 - `String s = "Isto é uma String"`
 - mais eficiente que usar **new**



Classe String

- Alguns métodos:
 - **int length()**
Retorna o tamanho da String
 - **char charAt(int i)**
Retorna o caractere no índice i
 - **int indexOf(char c);**
Retorna o indice do caractere c
 - **char[] toCharArray();**
Retorna a String em forma de array
 - **String toLowerCase();**
Retorna nova String toda minúscula
 - **String toUpperCase();**
Retorna nova String toda maiuscula
 - **String trim();**
Retorna nova String sem os espaços no inicio e fim
 - **int compareTo(String s);**
Compara duas Strings

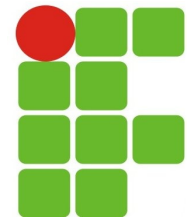


Classe String

Imutabilidade dos Objetos String:

```
1. String s = "Programando"  
2. s+= " com "  
3. s+= "Java"
```

- 1** - A operação acima colocou 3 objetos da classe String na memória;
- 2** - Cada atribuição que fazemos a um objeto String, na verdade estamos criando uma nova instância para aquele objeto, ou seja, estamos criando um novo objeto e descartando o outro;
- 3** - O garbage collector logo passa e recolhe os objetos String que não estão sendo mais referenciados.



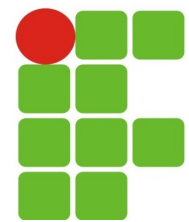
Exemplo String

```
public class StringExe {  
    public static void main(String[] args) {  
        System.out.println("ifrn".toUpperCase());  
        String site = "www.ifrn.edu.br";  
        String dominio = site.substring(site.indexOf(".")+1, site.length());  
        System.out.println(dominio);  
    }  
}
```

Java


```
public class StringExe {  
    public static void main(String[] args) {  
        String ifrn = "Instituto Federal de Educação, Ciência e Tecnologia do";  
        System.out.println(ifrn);  
        int tamanho = ifrn.length();  
        System.out.println("O tamanho é "+tamanho);  
        char c = ifrn.charAt(7);  
        System.out.println("O caractere no índice 7 é "+c);  
        int x = ifrn.indexOf('ç');  
        System.out.println("O ç está no índice "+x);  
        String ifrnMin = ifrn.toLowerCase();  
        System.out.println(ifrnMin);  
        String ifrnMai = ifrn.toUpperCase();  
        System.out.println(ifrnMai);  
        ifrn = ifrn + " Rio Grande do Norte";  
        System.out.println(ifrn);  
    }  
}
```

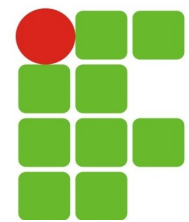
NÃO muda a String. Cria uma nova e referencia a nova



StringBuilder

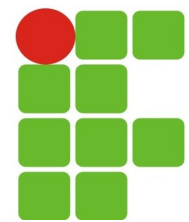
- Classe de texto mutável:
 - Necessário criar objeto com **new**;
 - Mais eficiente para tratar textos mutáveis:
 - Evita processamento com criação de objetos.
- Classe StringBuffer:
 - Mesmo que **StringBuilder**, porém usado quando há acesso concorrente.

Java



StringBuilder

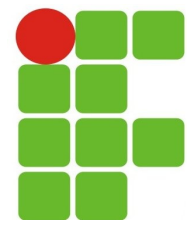
- Alguns Métodos:
 - `int length();`
retorna o tamanho do StringBuilder
 - `char charAt(int i);`
retorna o caractere no índice `i`
 - `append(x);`
 - Adiciona `x` no final da String
`x` pode ser: `boolean`, `char`, `char[]`, `CharSequence`, `double`, `float`, `int`, `long`, `Object`, `String`, `StringBuffer`.
 - `int capacity();`
Retorna o a capacidade do StringBuilder (espaço reservado) - **diferente de `length()`**
StringsBuilder são implementadas com arrays de `char`. `capacity` retorna o tamanho do array, enquanto `length` retorna o **espaço usado** no array.



StringBuilder

- Mais métodos
 - delete (int i , int f)
Apaga caracteres entre os índices i e f
 - insert (int i, x)
Insere x no índice i (similar a append)
 - replace (inicio, fim, String);
Substitui o texto entre inicio e fim por String
 - substring(inicio, fim);
Retorna String contida entre os índices inicio de fim
 - toString();
Retorna a String correspondente a StringBuilder

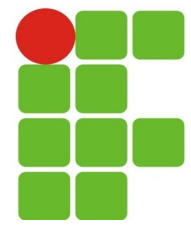
```
public class StringBuilderExe {  
    public static void main(String[] args) {  
        StringBuilder ifrn;  
        ifrn=new StringBuilder("Instituto Federal");  
        System.out.println(ifrn.toString());  
        int tamanho = ifrn.length();  
        System.out.println("O tamanho é " + tamanho);  
        char c = ifrn.charAt(9);  
        System.out.println("O caractere no índice 9 é " + c);  
        ifrn.append(" de Educação, ");  
        System.out.println(ifrn);  
        System.out.println("O tamanho é " + ifrn.length());  
        System.out.println("Capacidade " + ifrn.capacity());  
        ifrn.append("Ciência e Tecnologia do Rio Grande do Norte");  
        System.out.println("Capacidade " + ifrn.capacity());  
        System.out.println(ifrn);  
    }  
}
```



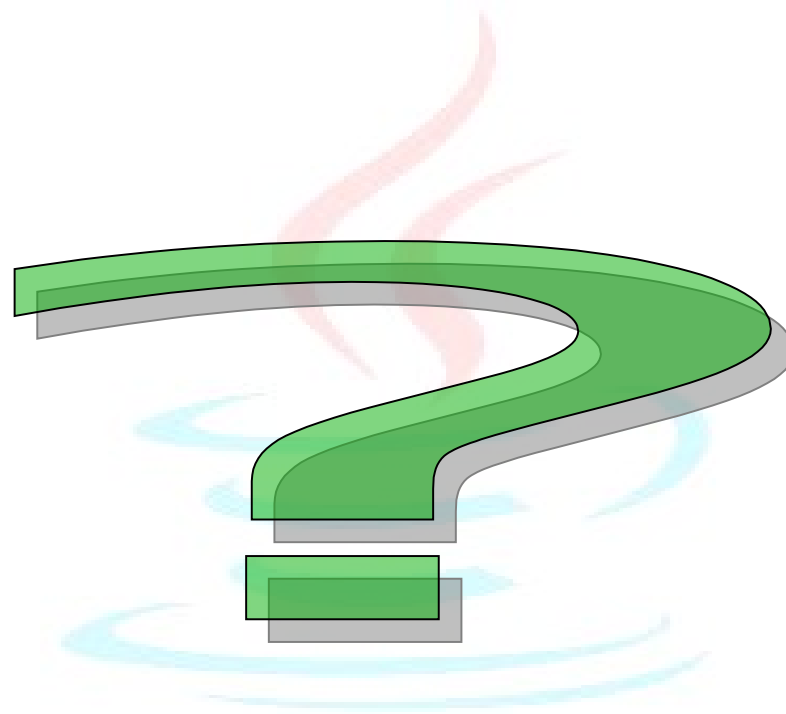
Exercício

Fazer uma aplicação que solicita o nome do usuário e depois imprime o seu nome ao contrário.

Java



Dúvidas



Java