

INSTITUTO FEDERAL

Prof. Dr. Bruno Queiroz Pinto



Classes **StringBuilder** e **StringBuffer**

- A classe `String` não deve ser usada para concatenação de outras `Strings` ou caracteres.
- Cada concatenação gera uma nova `String`, um novo objeto.

Strings são imutáveis



Classes **StringBuilder** e **StringBuffer**

- As classes *StringBuffer* e *StringBuilder* são usadas quando há necessidade de fazer muitas modificações em Strings.
- Ao contrário de Strings, objetos do tipo *StringBuffer* e *StringBuilder* podem ser modificados repetidamente sem deixar para trás muitos novos objetos não utilizados.
- Foram propostos a partir do Java 5.
- São bem mais rápidas na concatenação.



Classes `StringBuilder` e `StringBuffer`

```
// Criando um objeto da classe StringBuilder
```

```
StringBuilder texto = new StringBuilder();
```

```
// Inserindo texto no fim da String
```

```
texto.append("Meu texto");
```

```
// Inserindo texto em alguma parte da String
```

```
texto.insert(0, "meu texto");
```

```
// Apagar texto em uma parte da String
```

```
texto.delete(0,2);
```

Resumo da classe *`StringBuilder`*

Geralmente utilizamos mais o *`append`*.



Classes StringBuilder e StringBuffer

```
String strFinal = "";  
long tInicio = System.currentTimeMillis();  
for (int i = 0; i < 100000; i++) {  
    strFinal += "a";  
}  
long tFim = System.currentTimeMillis();  
long tResultado = tFim - tInicio;  
System.out.println("Tempo de Execução com operador '+' = " + tResultado + " ms");
```

```
StringBuilder strBuilder = new StringBuilder();  
tInicio = System.currentTimeMillis();  
for (int i = 0; i < 100000; i++) {  
    strBuilder.append("a");  
}  
tFim = System.currentTimeMillis();  
tResultado = tFim - tInicio;  
System.out.println("Tempo de Execução com StringBuilder = " + tResultado + " ms");
```

Código de comparação que faz a concatenação de 100000 letras 'a'.

- Método append permite inserir um novo texto ao fim do atual.

Desempenho:

Tempo de Execução com operador '+' = 1716 ms

Tempo de Execução com StringBuilder = 6 ms



Classes StringBuilder e StringBuffer

```
String strFinal = "Como " +  
    "devemos " +  
    "utilizar o " +  
    "operador +...";
```

```
int x = 10;  
int y = 20;  
System.out.println("x:" + x + " y:" + y);
```

Em que situação utilizar o “+” e a classe String: “multi-line Strings”

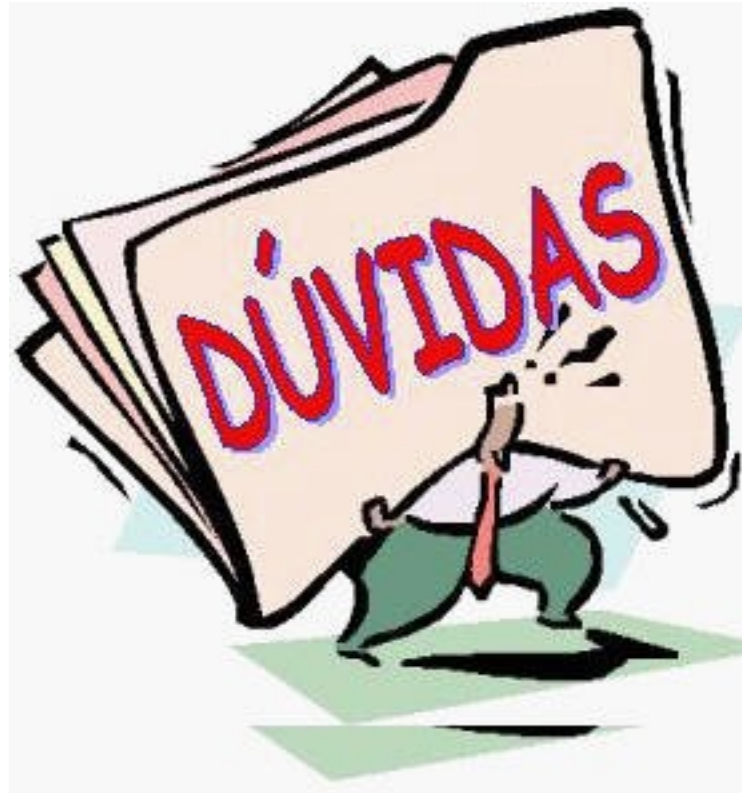
Poucas acréscimos também não apresenta problemas



Classes **StringBuilder** e **StringBuffer**

- Ambos são bem mais rápidos para concatenação de valores do que a String comum e fazem exatamente a mesma função.
- A principal diferença é que o StringBuffer é sincronizado, enquanto que o outro não.
- Assim, você garante a consistência do seu código quando há diversas threads lendo ou modificando a mesma String. Para esses casos, o ideal é usar o StringBuffer.
- Porém o StringBuilder ainda é mais rápido do que o StringBuffer

Fim aula 02-parte04.....



bruno.queiroz@iftm.edu.br