

Comparativo entre String, StringBuffer e StringBuilder:

Característica	String	StringBuilder	StringBuffer
Lançamento	JDK 1.0	JDK 1.5	JDK 1.0
Mutabilidade	Imutável	Mutável	Mutável
Thread Safety (Segurança da Trilha)	Sim	Não	Sim
Eficiência de memória	Alta	Suficientemente Eficiente	Menos Eficiente
Performance	Alta Sem Sincronização	Alta Sem Sincronização	Baixa - Com Sincronização
Uso	Quando deseja imutabilidade (não usar a memória heap e sim a "string pool")	Quando Thread Safety não é requerida	Quando Thread Safety é requerida

Códigos utilizados neste exemplo:

StringBuffer:

```
StringBuffer a1 = new StringBuffer("ABC");
StringBuffer b2 = new StringBuffer("ABC");

if (a1 == b2)
    System.out.println("As STRINGBUFFERS são iguais (COMPARAÇÃO ==)");
else
    System.out.println("As STRINGBUFFERS são diferentes (COMPARAÇÃO ==)");

System.out.println("As STRINGBUFFER são iguais? (Método EQUALS):" + a1.equals(b2));
```

Resultado:

As STRINGBUFFERS são diferentes (COMPARAÇÃO ==)

As STRINGBUFFER são iguais? (Método EQUALS):false

Strings:

```
String c1 = "ABC";
String d2 = "ABC";

if (c1 == d2)
    System.out.println("As STRINGS são iguais (COMPARAÇÃO ==)");
else
    System.out.println("As STRINGS são diferentes (COMPARAÇÃO ==)");

System.out.println("As STRINGS são iguais? (Método EQUALS):" + c1.equals(d2));

String c01 = new String("ABC");
String d02 = new String("ABC");

if (c01 == d02)
    System.out.println("As STRINGS são iguais (COMPARAÇÃO ==)");
else
    System.out.println("As STRINGS são diferentes (COMPARAÇÃO ==)");

System.out.println("As STRINGS são iguais? (Método EQUALS):" + c01.equals(d02));
```

```
String c001 = "ABC";
String d002 = new String("ABC");

if (c001 == d002)
    System.out.println("As STRINGS são iguais (COMPARAÇÃO ==)");
else
    System.out.println("As STRINGS são diferentes (COMPARAÇÃO ==)");
System.out.println("As STRINGS são iguais? (Método EQUALS):" + c001.equals(d002));
```

Resultado:

Primeiro Bloco de String: apenas declarando entre parênteses (string pool):

As STRINGS são iguais (COMPARAÇÃO ==)

As STRINGS são iguais? (Método EQUALS):true

Segundo Bloco de String: declaração com a palavra new nas duas variáveis:

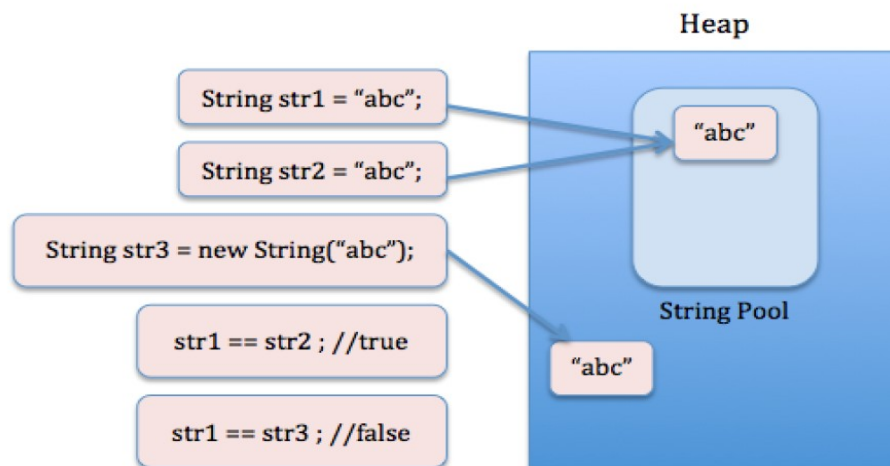
As STRINGS são diferentes (COMPARAÇÃO ==)

As STRINGS são iguais? (Método EQUALS):true

Terceiro Bloco de String: uma variavel declarando entre parênteses e outra com a declaração NEW:

As STRINGS são diferentes (COMPARAÇÃO ==)

As STRINGS são iguais? (Método EQUALS):true



StringBuilder:

```
StringBuilder e1 = new StringBuilder("ABC");
StringBuilder f2 = new StringBuilder("ABC");

if (e1 == f2)
    System.out.println("As STRINGSBUILDER são iguais (COMPARAÇÃO ==)");
else
    System.out.println("As STRINGSBUILDER são diferentes (COMPARAÇÃO ==)");
System.out.println("As STRINGSBUILDER são iguais? (Método EQUALS):" + e1.equals(f2));
```

Resultado:

As STRINGSBUILDER são diferentes (COMPARAÇÃO ==)

As STRINGSBUILDER são iguais? (Método EQUALS):false