

INSTITUTO FEDERAL

Prof. Dr. Bruno Queiroz Pinto

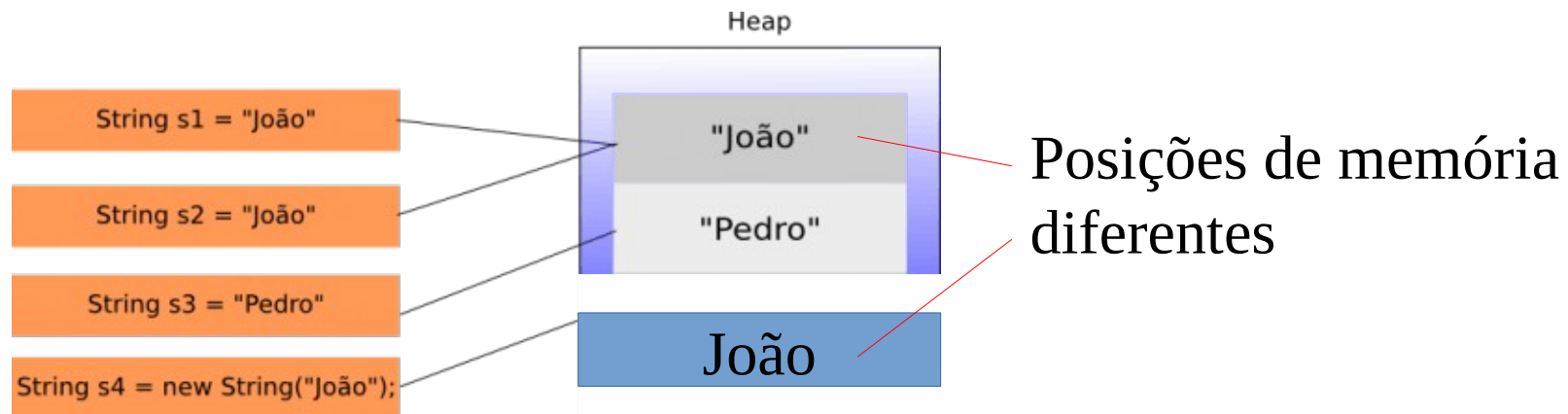
Manipulação de Texto

- **String é uma classe em Java.**
- String é um tipo de dado/estrutura/objeto que é utilizada para armazenar texto dentro de um programa sendo composta por uma série de caracteres.
- Variáveis do tipo String guardam referências a objetos, e não um valor, como acontece com os tipos primitivos.

Manipulação de Texto

o Diferentes formas de criar Strings?

- String criada com a instrução `new` cria um novo objeto.
- String que recebem conteúdo de algum método (`s.next()`), cria um novo objeto.
- String (simplesmente colocando caracteres entre aspas dupla) utilizam Pool de Strings que permite reaproveitar um objeto na memória.



Manipulação de Texto (Extra)

- **Utilizar a Classe/Biblioteca String.**
- **Inicialização:**

String texto1 = "Texto"; → Modo simplificado

String texto2 = new String();

String texto3 = new String("Texto");

char texto[] = {'T','e','x','t','o'};

String texto4 = new String(texto);

String texto5 = new String(texto,0,3);

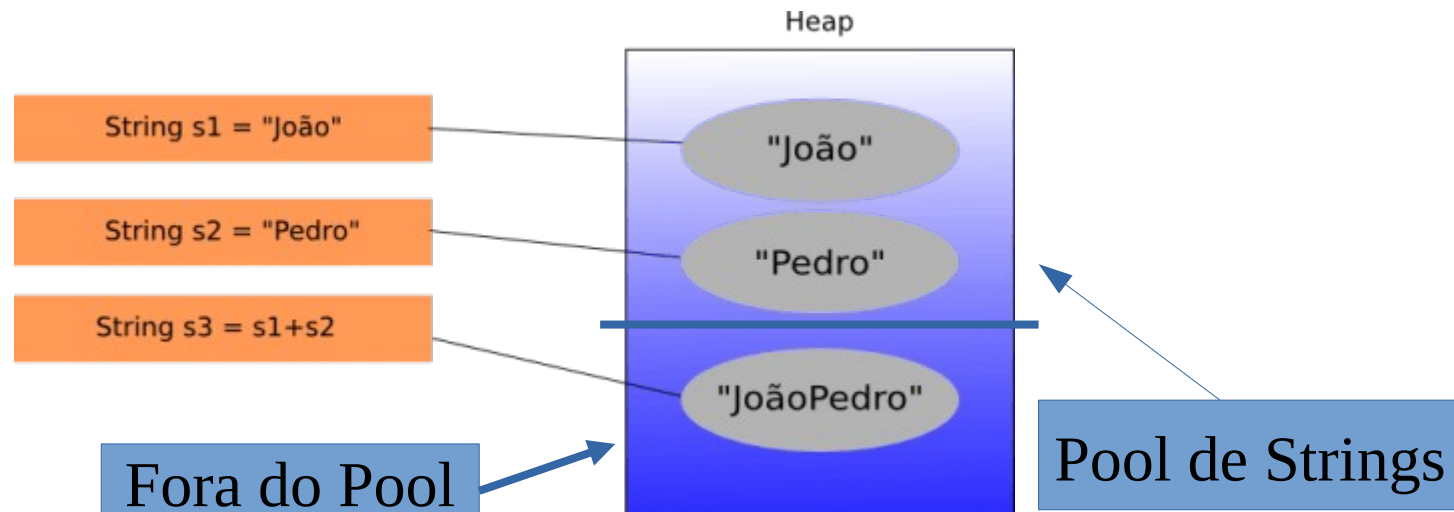
// texto5 contém "Tex"

Cria um
novo objeto
String

Manipulação de Texto

○ Imutabilidade

- String são imutáveis : depois de serem criados eles não podem ser alterados.
- Quando concatenamos (juntamos) duas Strings, estamos criando um novo objeto em um outro local da memória.



Manipulação de Texto

○ Imutabilidade

1. String s1 = "João";

2. s1 = s1 + "Pedro";

- Na linha 1 é criado um objeto no Pool(ou pegamos um já armazenado, não sabemos) e colocamos sua referência em s1.
- Na linha 2 é criado um objeto no Pool “Pedro”, não armazenamos esta referência e sim utilizamos para concatenar com o objeto referenciado por s1, criando um novo objeto **new String("JoãoPedro")** que será armazenado em algum outro lugar na memória e sua referência será armazenada em s1, deste modo perdemos a referência da String “João”, mas o objeto não foi destruído ou modificado.

Leitura do Teclado

o Utilizar a Classe/Biblioteca Scanner

```
Scanner s = new Scanner(System.in);  
String texto1 = s.nextLine(); // todo o texto da linha  
String texto2 = s.next(); // a primeira palavra
```

```
import java.util.Scanner;
```

Leitura do Teclado

○ Utilizar a Classe/Biblioteca Scanner

```
Scanner s = new Scanner(System.in);  
System.out.println("Digite o número");  
int x = s.nextInt();  
System.out.println("Digite a frase");  
String texto1 = s.nextLine(); // todo o texto da linha  
  
System.out.println("Digite a palavra");  
String texto2 = s.next(); // a primeira palavra
```

Como é a execução??

Leitura do Teclado

- Isso acontece após ler um valor não alfanumérico (números, por exemplo), onde o valor é armazenado na variável, mas o ENTER continua no buffer.
- Você precisa remover o ENTER.

Inserir um `s.nextLine()` após a linha com `s.nextInt()`

Ou

Criar uma função que verifica se há o ENTER
e então executa o `s.nextLine()`

```
:  
int x = s.nextInt();  
s.nextLine();  
System.out.println("Digite a frase");  
:
```

OU

```
int x = s.nextInt();  
clearBuffer(s);  
System.out.println("Digite a frase");
```

```
public static void clearBuffer(Scanner scanner) {  
    if (scanner.hasNextLine()) {  
        scanner.nextLine();  
    }  
}
```

Comparar Strings

```
String x,y;  
Scanner s = new Scanner(System.in);  
x = "Texto";  
y = "Texto";  
if (x.equals(y))  
    System.out.println(x + " igual a " + y);  
else  
    System.out.println(x + " diferente a " + y);  
if (x==y)  
    System.out.println(x + " igual a " + y);  
else  
    System.out.println(x + " diferente a " + y);  
Implementar, resultado?
```



Pool de Strings

Onde ficam armazenadas as Strings com valores únicos, inicializadas de forma literal.

Nesse caso funciona

E se não for um texto com valor constante?

Comparar Strings

```
String x,y;  
Scanner s = new Scanner(System.in);  
x = s.nextLine();  
y = s.nextLine();  
if (x.equals(y))  
    System.out.println(x + " igual a " + y);  
else  
    System.out.println(x + " diferente a " + y);  
if (x==y)  
    System.out.println(x + " igual a " + y);  
else  
    System.out.println(x + " diferente a " + y);
```

equals() : correto

Pool de String

Implementar, resultado?





Descobrir a quantidade de caracteres

```
String x;  
Scanner s = new Scanner(System.in);  
x = s.nextLine();  
System.out.println("O texto " + x + " contém " + x.length());
```

Implementar, resultado?

Variavel.length()

Diferente do tamanho de vetores

Vetores: nomeVetor.length

String: nomeString.length()



Manipulando caracteres na String

```
String x;
```

```
Scanner s = new Scanner(System.in);
```

```
x = s.nextLine();
```

```
int pos = s.nextInt();
```

```
char c = x.charAt(pos);
```

```
System.out.println("O caractere : " + c + " esta na  
posição " + pos + " do texto " + x);
```

Se digitar algo que não seja número ou uma posição não existente, o programa irá dar erro durante a execução.



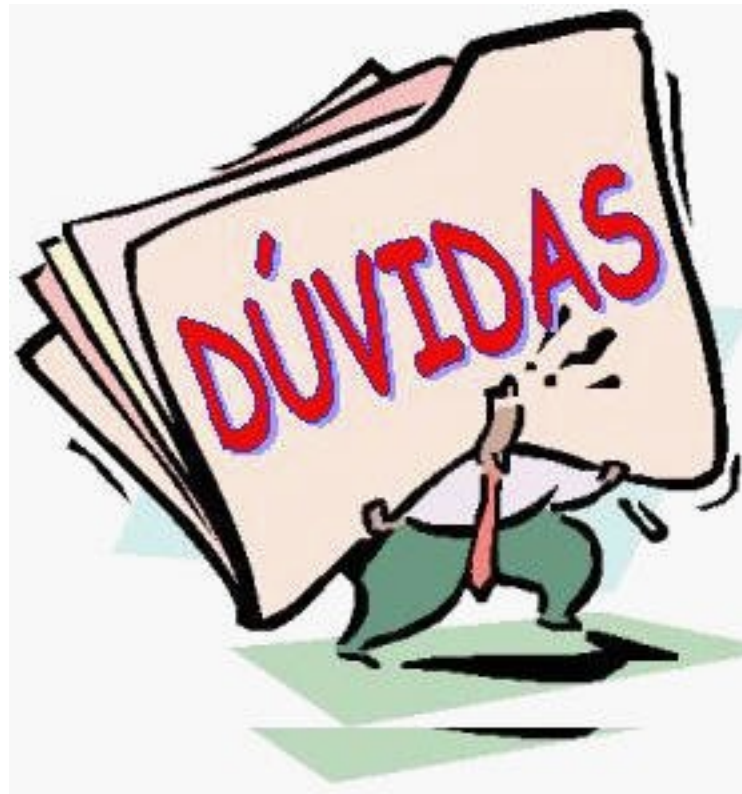
Ler letra a letra de um texto.

```
String x;  
Scanner s = new Scanner(System.in);  
x = s.nextLine();  
for (int pos = 0; pos < x.length(); pos++) {  
    char c = x.charAt(pos);  
    System.out.println("O caractere : " + c + " esta na posição  
        " + pos + " do texto " + x);  
}
```

Implementar, resultado?



Fim aula 01.....



bruno.queiroz@iftm.edu.br