

5. Manipulação de Strings

- Em Java, strings são objetos da classe String;
- Para construir um string:

```
String s = new String("Exemplo");
```

```
String x = "Curso de Java";
```

```
String str = new String(); // constrói um objeto string nulo
```

Principais métodos da classe String

Método **int length()** - retorna o número de caracteres do string.

```
String s = new String("Isaias C. Boratti");
```

```
int tam = s.length();
```

tam

17

Método **char charAt(int p)** – retorna o caracter que está na posição *p*. O primeiro caracter está na posição 0.

```
String x = “Exemplo” ;  
char character = x.charAt(2);
```

caracter

e

Método **boolean equals(String s)** – retorna um valor boolean informando se os dois strings são ou não iguais.

```
String s1 = new String(“Curso Java” );  
boolean b = s1.equals(“Curso JDK”); // b armazenará o valor false
```

Lembre-se: Strings são objetos.

Método **int compareTo(String s)** – compara o string com o string *s*.

```
String s1 = new String(“Joao”);  
String s2 = “Joana”;  
int k = s1.compareTo(s2);  
// k será igual a 0 se s1=s2  
// k será igual a 1 se s1>s2  
// k será igual a -1 se s1<s2
```

k

-1

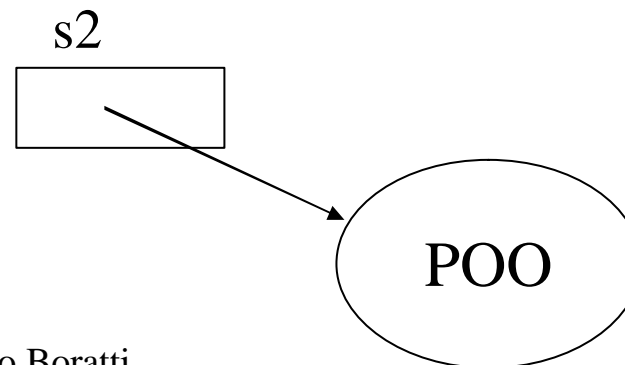
Método **int indexOf(char c)** – retorna a posição da primeira ocorrência do caracter *c* dentro do string. Caso *c* não exista no string retornará -1

Método **int indexOf(char c, int pos)** – retorna a posição da primeira ocorrência do caracter *c* após a posição *pos*.

```
String s = " Maria da Silva";  
int k = s.indexOf('a'); // k = 1  
k = s.indexOf('x'); // k = -1  
k = s.indexOf('a',5); // k = 7
```

Método **String substring(int inicio, int fim)** – retorna o string copiado do string executor, começando na posição *inicio* e terminando na posição *fim-1*.

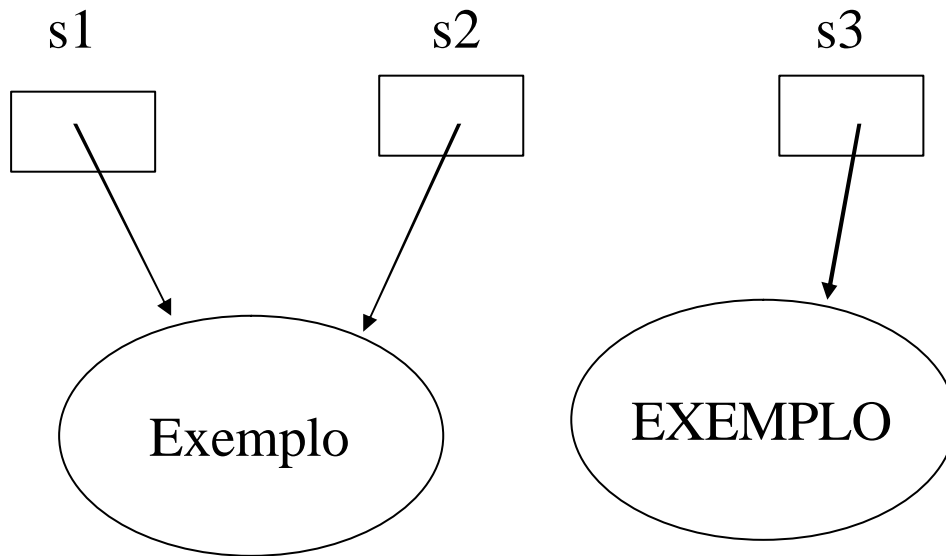
```
String s1, s2;  
s1 = "Curso de POO – Java";  
s2 = s1.substring(9,12);
```



Métodos **String toUpperCase()** e **String toLowerCase()** – retornam, respectivamente, um string igual ao executor com todas as letras em maiúsculo e um string com todas as letras em minúsculo.

Lembre-se: Strings são objetos.

```
String s1 = "Exemplo";  
String s2 = s1;  
String s3 = s2.toUpperCase();
```



Transformações envolvendo strings

```
int i = 10;
double a=15.53;
String str1 = "12.34";
String str2 = "127";
String str3 = "240.00";
double valor = Double.parseDouble(str1);
int b = Integer.parseInt(str2);
float sm = Float.parseFloat(str3);
String aux1 = Integer.toString(i);
String aux2 = Double.toString(a);
String aux3 = Character.toString('A');
```

A classe Character

- Apresenta métodos que manipulam um caracter. A maioria dos métodos são métodos de classe (métodos estáticos).

Método **static boolean isDigit(char car)** – retorna um valor boolean informando se o caracter *car* é ou não um dígito.

Método **static boolean isLetter(char car)** – retorna um valor boolean informando se o caracter *car* é ou não uma letra.

Método **static boolean isLowerCase(char car)** – retorna um valor boolean informando se o caracter *car* é ou não uma letra maiúscula.

Método **static boolean isUpperCase(char car)** – retorna um valor boolean informando se o caracter *car* é ou não uma letra maiúscula.

Método **static char toUpperCase(char car)** – retorna o maiúsculo relativo a *car*. Caso *car* não seja letra retorna o próprio caracter *car*.

Método **static char toLowerCase(char car)** – retorna o minúsculo relativo a *car*. Caso *car* não seja letra retorna o próprio caracter *car*.

Exercício: Escreva uma classe contendo métodos estáticos que, para um determinado string, determine:

- a) Quantidade de letras;
- b) Quantidade de vogais;
- c) Quantidade de consoantes;
- d) Quantidade de palavras;
- e) Se o string é ou não um palíndromo.

```
public class ProcessaString
{
    public static int quantLetras( String s)
    {
        int cont = 0;
        int comp = s.length();
        for (int i=0; i<comp; i++)
            if (Character.isLetter(s.charAt(i)))
                cont ++;
        return cont;
    }
}
```

```

public static boolean e_Vogal(char c)
{
    c = Character.toUpperCase(c);
    return (c=='A') || (c=='E') || (c=='I') || (c=='O') || (c=='U'); }
public static boolean e_Consoante( char c)
{ // implementar }
public static int quantVogais( String s)
{ // implementar
}
public static int quantConsoantes ( String s)
{ // implementar
}
public static int quantPalavras(String s)
{ // implementar
}
public static boolean e_Palindromo( String s)
{ // implementar }
}

```


O valor de um caracter é seu valor inteiro no conjunto de caracteres unicode.

```
int k = 'A'; // k = 65
```

```
int soma = 'A'+'B'; // soma = 131
```