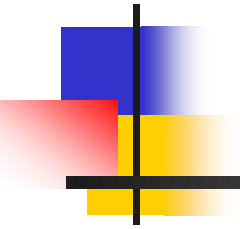
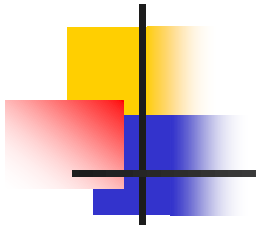


Classe String em Java

Profª. Rachel Reis

rachel@reis.ufpr.br



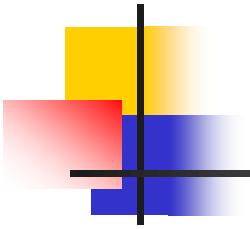
String

- Característica:
  - É uma classe Java.
  - São tratados como tipos primitivos (ex.: int, float), logo não é necessário fazer o import do pacote dessa classe (java.lang.String).

Exemplo 1

```
// Declaração
String texto;

// Instanciação
texto = new String("Seja bem vindo.");
```

```
// Declaração
String texto;

// Instanciação
String texto = "Seja bem vindo.";
```



Métodos da classe String

- A classe String em Java possui mais de 50 métodos.

- Exemplos:
  - método length()
  - método equals(...)
  - método concat(...)
  - método indexOf(...)

Ex2 - método length()

- Calcula o **tamanho** da String.
- Exemplo:

```
// Declaração e inicialização
String frase = "Bom dia Julieta!";
// Uso do método length()
int tam = frase.length();
// Saída
System.out.printf("O tamanho da sentença \"%s\" eh: %d \n", frase, tam);
```

Atividade 1

- Escreva um programa em Java que **leia** uma sentença digitada pelo usuário. Em seguida, imprima o tamanho da sentença.

→ Utilize a classe Scanner

Ex3 - método equals()

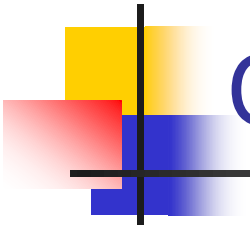
- **Compara** se duas strings são iguais ou não.
- Exemplo:

```
// Declaração
String p1 = "bola";
String p2 = "bala";

// Uso do método equals
boolean palavraIguais = p1.equals(p2);

if(palavraIguais == true)
  System.out.printf("%s e %s são palavras iguais.\n", p1, p2);
else
  System.out.printf("%s e %s são palavras diferentes.\n", p1, p2);
```

Atividade 2

- Escreva um programa em Java que **leia** duas palavras digitadas pelo usuário e verifique se elas são iguais.

→ Utilize a classe Scanner



Outros Métodos de Comparação

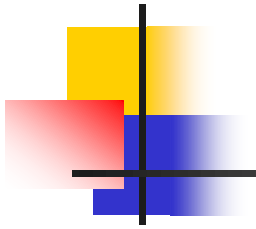
- **compareTo(String s)**
  - Compara duas strings em ordem alfabética retornando zero se as strings forem iguais e diferente de zero se forem diferentes.

- **compareToIgnoreCase(String s)**
  - Compara duas strings em ordem alfabética ignorando maiúsculas e minúsculas.

Ex4 - método concat()

- **Concatena** duas strings.
- Exemplo:

```
// Declaração e inicialização
String parte1 = "auto";
String parte2 = "ajuda";

// Uso do método concat()
String completa = parte1.concat(parte2);

//Saída
System.out.println("\nString concatenada: " + completa);
```



Atividade 3

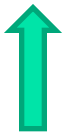
- Escreva um programa em Java que **leia** duas palavras digitadas pelo usuário, faça a concatenação e imprima a palavra concatenada na tela.

→ Utilize a classe Scanner

Método indexOf(...)

- Retorna o **índice** da primeira ocorrência de um caractere.

E	N	C	R	U	Z	I	L	H	A	D	A
0	1	2	3	4	5	6	7	8	9	10	11



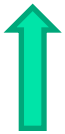
- Exemplo:

```
int posicao = palavra.indexOf('D');
System.out.println("A letra d está na posicão de índice: " + posicao);
```

Método indexOf (...)

- Retorna o **índice** do início de uma substring

E	N	C	R	U	Z	I	L	H	A	D	A
0	1	2	3	4	5	6	7	8	9	10	11



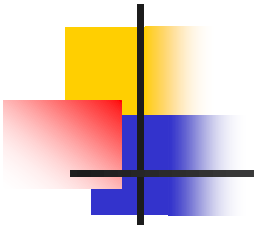
- Exemplo:

```
// Uso do método indexOf
String palavra = "ENCRUZILHADA";
int posicao1 = palavra.indexOf("CRUZ");
System.out.println("A palavra CRUZ inicia na posicão de índice: " + posicao1);
```

Outros Métodos

- **replace** (**char** caractere_antigo, **char** caracter_novo)
  - Retorna uma nova string substituindo todas as ocorrências do caractere_antigo pelo caractere_novo
  - Exemplo:

```
frase.replace('p', 'b');
```



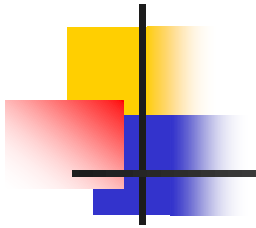
Outros Métodos

- **replace** (**char** caractere_antigo, **char** caracter_novo)
  - Retorna uma nova string substituindo todas as ocorrências do caractere_antigo pelo caractere_novo
  - Exemplo:

```
frase.replace('p', 'b');
```

- **substring** (**int** inicio, **int** fim)
  - Retorna uma parte da string original, delimitada pelos índices de início e fim.
  - Exemplo:

```
String sub = frase.substring (5, 10);
```



Outros Métodos

- **charAt(int** pos**)**
  - Retorna o caractere que está na posição pos da String (lembrando que o primeiro caractere da String está na posição 0).
  - Exemplo:

```
char letra5 = frase.charAt(4);
```

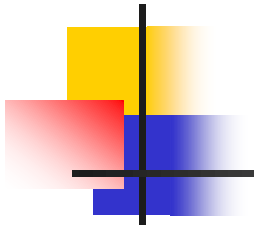
Outros Métodos

- **charAt(int** pos**)**
  - Retorna o caractere que está na posição pos da String (lembrando que o primeiro caractere da String está na posição 0).
  - Exemplo:

```
char letra5 = frase.charAt(4);
```

- **indexOf(char** caractere**)**
  - Retorna a posição da primeira ocorrência de um caractere na String.
  - Exemplo:

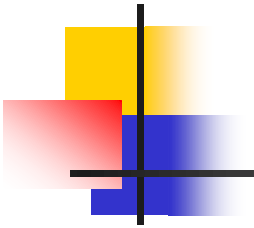
```
int posicao = frase.indexOf('c');
```



Outros Métodos

- **toLowerCase**()
  - Retorna uma nova string com todas as letras em minúsculo. Esse método não possui argumentos.
  - Exemplo:

```
frase.toLowerCase();
```



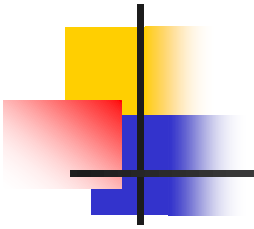
Outros Métodos

- **toLowerCase**()
  - Retorna uma nova string com todas as letras em minúsculo. Esse método não possui argumentos.
  - Exemplo:

```
frase.toLowerCase();
```

- **toUpperCase** ()
  - Retorna uma nova string com todas as letras em maiúsculo. Esse método não possui argumentos.
  - Exemplo:

```
frase.toUpperCase();
```



Exemplo 4

- Escreva um programa em Java que exiba uma String na ordem inversa.

- Exemplo:
  - Entrada:

    engenharia de software
  - Saída:

    erawtfos ed airahnegne

```
public class Principal
{
    public static void main(String[ ] args)
    {
        /*
            A classe Principal contém o método main() que
            é responsável por iniciar o programa Java.
        */

    }
}
```

```
public class Principal
{
	public static void main(String[] args)
	{
		String palavra = "engenharia de software";

		/*
			Declaração e inicialização da string palavra.
		*/

	}
}
```

```
public class Principal
{
	public static void main(String[] args)
	{
		String palavra = "engenharia de software";
		int tam = palavra.length();

		/*
			Calcula o número de caracteres da string
			palavra e armazena na variável tam.
		*/


	}
}
```

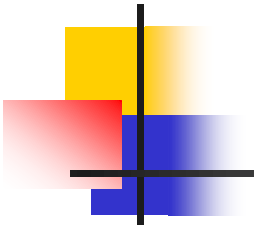
```
public class Principal
{
	public static void main(String[] args)
	{
		String palavra = "engenharia de software";
		int tam = palavra.length();
		int i;

		for (i = tam - 1; i >= 0; i--)
		{

		}

	}
}
```

```
/*
	Percorre a string palavra na ordem
	inversa.
*/
```

```
public class Principal
{
	public static void main(String[] args)
	{
		String palavra = "engenharia de software";
		int tam = palavra.length();
		int i;

		for (i = tam - 1; i >= 0; i--)
		{
			System.out.print(palavra.charAt(i));
		}

	}
}
```

```
/*
	Exibe o caractere na posição de índice i.
*/
```

```
public class Principal
{
	public static void main(String[] args)
	{
		String palavra = "engenharia de software";
		int tam = palavra.length();
		int i;

		for (i = tam - 1; i >= 0; i--)
		{
			System.out.print(palavra.charAt(i));
		}
		System.out.print("\n");
	 }
}
```

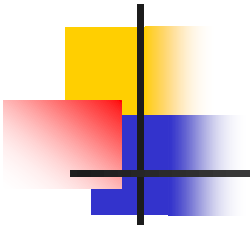


Praticar...

1) Crie um programa Java que leia uma sentença e substitua todas as letras a por @

2) Crie um programa Java que leia uma sentença e remova todos os espaços em branco

3) Fazer um programa em Java que leia um nome no formato comum, por exemplo “José Silva”, e o imprima no formato “Silva, J.”.



Referências

- Deitel, P. J.; Deitel, H. M. (2010). Java como programar. 8a edição. São Paulo: Pearson Prentice Hall.

- Deitel, P. J.; Deitel, H. M. (2017). Java como programar. 10a edição. São Paulo: Pearson Prentice Hall.