

# Programação Orientada a Objetos

## Semana 02 Classes Math e String

## Reflexão

“Aquilo que escuto eu esqueço,  
Aquilo que vejo eu lembro,  
Aquilo que **faço** eu aprendo.”

Confúcio

# Java Classes

A linguagem Java traz algumas classes já prontas, recursos que podem ser usados a qualquer momento dentro de uma aplicação.

Exemplo:

- ✓ Math
- ✓ String

## Classe Math

Possui diversos métodos especializados em realizar cálculos matemáticos.

Sintaxe:

***Math.<nome do método>(argumentos ou lista de argumentos)***

Não é necessário importar a classe Math, pois faz parte do pacote java.lang

## Constantes da classe

- ✓ Math.PI – valor de pi (3.14159265358979323846)
- ✓ Math.E – logaritmos naturais (2.7182818284590452354)

Exemplo:

```
public class ComprimentoCirculo {  
    public static void main(String[] args) {  
        float raio = 2.4f;  
        double comprimento = 2 * raio * Math.PI;  
        System.out.println(comprimento);  
    }  
}
```

## Métodos da classe Math

- ✓ **ceil( )** → arredonda um valor para cima
- ✓ **floor( )** → arredonda um valor para baixo
- ✓ **max( )** → retorna o maior valor entre dois números
- ✓ **min( )** → retorna o menor valor entre dois números
- ✓ **sqrt( )** → extrai a raiz quadrada de um número.
- ✓ **pow( )** → retorna o quadrado de um número
- ✓ **random( )** → retorna um valor aleatório do tipo double entre 0.0 e 1.0 (o valor 1 nunca é gerado)

## Método ceil

Retorna um número do tipo **double** para o seu **próximo inteiro** (arredonda sempre para cima).

Sintaxe:

**Math.ceil(<valor do tipo double>)**

Exemplos:

```
int a = Math.ceil(10.1);
```

```
System.out.println(Math.ceil(10.6));
```

```
int c = Math.ceil(valor);
```

## Método floor

Retorna um número do tipo **double** para seu inteiro anterior (arredonda sempre para baixo).

Sintaxe:

**Math.floor(<valor do tipo double>);**

Exemplos:

```
int a = Math.floor(10.1);
```

```
System.out.println(Math.floor(10.6));
```

```
int c = Math.floor(valor);
```



## Método max

Retorna o maior valor entre dois números, que podem ser do tipo double, float, int ou long.

Sintaxe:

**Math.max( <valor1>, <valor2> );**

Exemplos:

```
double maior = Math.max(10.1, 8.7);
```

```
System.out.println(Math.max(10, 12));
```

```
double maiorValor = Math.max(valor1, valor2);
```

## Método min

Retorna o menor valor entre dois números. Também podem ser do tipo double, float, int ou long.

Sintaxe:

**Math.min( <valor1>,<valor2> );**

Exemplos:

```
double menor = Math.min(10.1,8.7);
```

```
System.out.println(Math.min(10,12));
```

```
double menorValor = Math.min(valor1,valor2);
```

## Método sqrt

Retorna a raiz quadrada de um determinado número. O número que se deseja extrair a raiz deve ser do tipo double.

Sintaxe:

**Math.sqrt(<valor do tipo double>);**

Exemplos:

```
double raiz = Math.sqrt(400);
```

```
System.out.println(Math.sqrt(82.5));
```

```
double raizQuadrada = Math.sqrt(valor);
```

## Método pow

Retorna um determinado número ao quadrado ou a qualquer outro valor de potência. Os números utilizados devem ser do tipo double.

### Sintaxe:

**Math.pow( <valor da base>, <valor da potência> );**

### Exemplos:

```
double quadrado = Math.pow(5.5,2);
```

```
System.out.println(Math.pow(25,5));
```

```
double numQuadrado = Math.pow(base,potencia);
```

## Método random

É utilizado para retornar valores de forma aleatória.  
Utilizar a classe `java.util.Random` para gerar números pseudoaleatórios.

Sintaxe:

**`Math.random();`**

Exemplos:

```
double n = new Math.random();  
System.out.println(Math.random()*10);
```

## Classe String

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
```

Da mesma forma que as funções matemáticas, existem diversos métodos para manipulação de strings. Esses métodos acompanham a seguinte sintaxe:

**<Nome da string>.<nome do método>(argumentos)**

## Método length()

Retorna o número de caracteres existentes em uma String, incluindo também os espaços em branco. Sempre retorna um valor do tipo int.

Sintaxe:

**<String>.length();**

Exemplos:

```
String s = "rosto bonito";  
int tam = s.length();
```

## Método charAt

Retorna o caracter que está na posição do índice fornecido. O primeiro caracter está na posição 0.

Sintaxe:

**<String>.charAt(<índice>);**

Exemplos:

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

e



## Método equals

Retorna um valor boolean informando se os dois strings são ou não iguais.

Sintaxe:

**<String>.equals(<string>);**

Exemplos:

String s1 = "Curso Java";

boolean b = s1.equals("Curso JDK");

false

## Método compareTo

Retorna a comparação entre duas strings, retornando um dos três valores: -1, 0 ou 1.

### Sintaxe:

**<String>. compareTo(<string>);**

### Exemplos:

```
String s1 = "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
```

-1

## Método substring

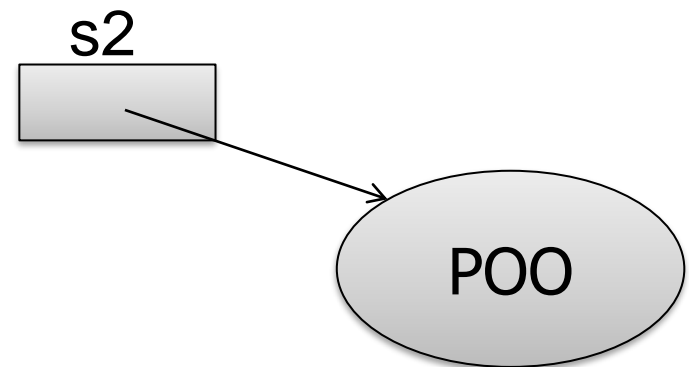
Retorna uma cópia de caracteres de uma string a partir de dois índices inteiros especificados, funcionando basicamente da mesma forma que o método charAt dentro de um looping.

### Sintaxe:

**<String>.substring(<índice inicial>,[<índice final>])**

### Exemplo:

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



## Método toUpperCase() e toLowerCase()

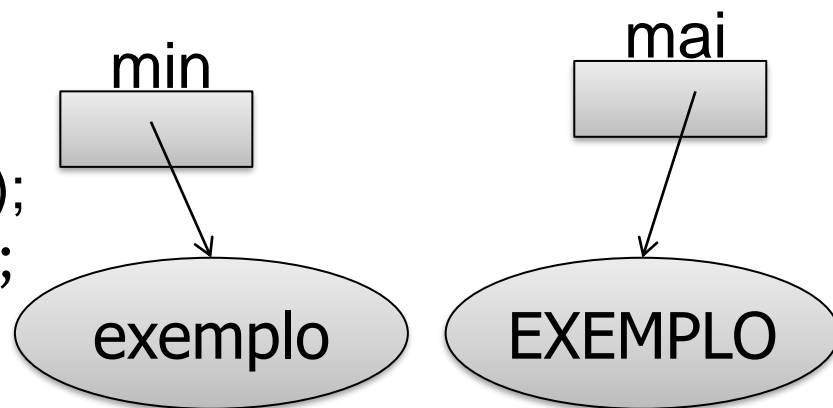
Retornam, respectivamente, um string igual ao executor com todas as letras em maiúsculo e um string com todas as letras em minúsculo.

Sintaxe:

**<String>.toUpperCase() ou <String>.toLowerCase()**

Exemplo:

```
String texto = "Exemplo";  
String min = texto.toLowerCase();  
String mai = texto.toUpperCase();
```



## Método trim

Retorna o valor da string sem os espaços em branco que aparecem no início e no final. Serão removidos apenas os espaços do início e do fim da string; não serão removidos os espaços entre as palavras.

Sintaxe:

**<String>.trim();**

Exemplos:

```
String frase = "  Cristo: o rei dos reis ";  
System.out.println("Com espaços: " + frase );  
System.out.println("Sem espaços: " + frase.trim());
```

## Método replace

Retorna o valor com a substituição de caracteres, ou grupo de caracteres, em uma determinada string.

### Sintaxe:

**<String>.replace(<caracteres a serem substituídos>, <substituição>)**

### Exemplo:

```
String nome = "mesquito";  
String nomeAlterado = nome.replace('e', 'o');  
System.out.println(nomeAlterado);
```

mosquito

## Exemplo valueOf

Retorna o valor de diversos tipos de dados convertidos em strings. Esse método aceita vários tipos de argumento (números ou cadeia de caracteres) e transforma-os em strings.

### Sintaxe

**String.valueOf(<nome da variável a ser convertida>)**

### Exemplo:

```
boolean myBoolean = true;  
System.out.println(String.valueOf(myBoolean));  
double myDouble = 10.30;  
System.out.println(String.valueOf(myDouble));
```

## Método indexOf

Retorna a posição de caracteres ou substrings em uma String. Caso haja insucesso na busca é retornado o valor inteiro -1.

### Sintaxe:

**String.indexOf(<caractere ou substring a ser localizada, [posição inicial]>)**

### Exemplo:

```
String str = "Welcome to home");  
System.out.println(str.indexOf( 'o' ));  
System.out.println(str.indexOf( 'e' ,2));
```

4

6



## Exercícios

1. Faça uma aplicação que leia duas strings do usuário e
  - a) apresente seus conteúdos em tela
  - b) apresente seus comprimentos em tela
  - c) informe se os conteúdos são iguais ou diferentes

Exemplo de execução:

Entrada da String 1: "Brasil".

Entrada da String 2: "Brasileiro"

Saídas:

- a) Brasil, Brasileiro
- b) 6, 10
- c) possuem conteúdos diferentes.

## Exercícios

2. Faça uma aplicação que solicite a digitação de um valor numérico de quatro dígitos e imprima em tela cada um dos seus dígitos por extenso.

Exemplo de execução:

Número digitado: 4571

Saída em tela: quatro, cinco, sete, um

3. Faça uma aplicação que calcule quantos metros cúbicos de água suporta uma determinada caixa-d'água em forma de cubo (todos os lados são iguais). O usuário deverá informar o valor do lado e o volume será calculado pela fórmula:  $\text{Volume} = \text{lado}^3$ .

4. Faça uma aplicação que leia as coordenadas de dois pontos e calcule a distância entre eles. A distância entre dois pontos é dada pela seguinte equação:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

## Exercícios

5. Faça uma aplicação que solicite ao usuário a escolha de uma operação:

1-Soma, 2-Subtração, 3-Multiplicação, 4-Divisão, 5-Potência e 6-Raiz.

Para as cinco primeiras opções solicite dois números e mostre em tela o valor da operação selecionada. No caso da raiz (item 6), solicite apenas um número e mostre em tela a raiz quadrada desse número.

Exemplo de execução:

Entrada: "Soma".

Entrada do número 1: 8

Entrada do número 2: 10

Saída:

A soma dos números 8 e 10 é igual a 18.

## Exercícios

6. Faça uma aplicação em Java para calcular a área de um círculo. Solicite que o usuário entre com o raio e retorne a área. Leve em consideração a equação:

$$\text{Area} = \text{Pi} * \text{raio}^2$$

7. Faça uma aplicação que simule a jogada de um dado. A cada execução mostre na tela um número aleatório entre 1 e 6. Mostre o número sorteado e sua escrita por extenso. Lembrando que deve aparecer apenas os números de um dado: 1, 2, 3, 4, 5 ou 6.

Saída:

Número sorteado: 6 – Seis

8. Faça uma aplicação que solicite uma String contendo a data de nascimento (dd/mm/aaaa) do usuário e imprima a data com o nome do mês por extenso.

Entrada: 29/10/1973

Saída: Você nasceu em 29 de Outubro de 1973.

## Exercícios

9. Faça uma aplicação que solicite uma frase e um caractere. Sempre que o caractere lido aparecer na frase ele deve ser substituído por asterisco.

Exemplo de execução:

Frase: “o dia esta nublado”.

Caractere: a

Saída:

o di\* est\* nubl\*do

10. Faça uma aplicação que solicite um login e uma senha. A aplicação deverá checar se os dados digitados pelo usuário são iguais aos dados internos do programa, que são: Login = “Admin” Senha = “Admin”. Se os dados estiverem corretos, o programa deverá imprimir a frase “BEM VINDO”, caso contrário, deverá ser impressa “Login e/ou Senha incorretos!”.