



Programação Java

## ▶ Introdução à linguagem Java

// Prática Integrativa - Programação Orientada a Objetos

### Objetivo

O objetivo deste guia prático é fortalecer e aprofundar ainda mais os conceitos de exceções e classes de utilitários. Para isso, propõe-se uma série de exercícios simples que nos permitirão revisar os tópicos que estamos estudando.

Are you ready? 🤖👍



## Exercício 1

1. Crie uma classe `PracticaExcecoes` que defina os atributos `a = 0` e `b = 300` do tipo `int`. Calcule o quociente de `b / a`. Trate a exceção que é lançada indicando a mensagem "Ocorreu um erro". Ao final do programa deve sempre indicar a mensagem "Programa finalizado"
2. Modifique o programa anterior para que, quando o erro ocorrer, em vez de imprimir a mensagem "Ocorreu um erro" no console, ele lance como uma exceção do tipo `IllegalArgumentException` com a mensagem "Não pode ser dividido por zero"



## Exercício 2

A operação realizada no programa a seguir lançará uma exceção:

```
//Mensagem final
String mensagemFinal = "Esta é a última mensagem";

//Código que lança exceção
int[] numeros = new int[5];
numeros[5] = 10;
```



Eles são solicitados a realizar a gestão correspondente para atingir o seguinte comportamento:

- \* Permitir a execução de código e capturar a exceção lançada
- \* Imprima pelo console a mensagem de erro do mesmo ao fazer a captura
- \* Imprima o texto da variável mensagemFinal pelo console, para que seja sempre exibido (ou seja, se uma exceção é lançada ou não)



### Exercício 3

Ao projetar o método a seguir, os possíveis erros que podem ocorrer não foram levados em consideração, portanto, é necessário concluí-lo aplicando o tratamento de exceção adequado.

```
public void lerArquivo(String nomeArquivo) {  
    String texto = null;  
  
    BufferedReader bufferedReader = new  
    BufferedReader(new FileReader(nomeArquivo));  
  
    while((texto = bufferedReader.readLine()) != null) {  
        System.out.println(texto);  
    }  
    bufferedReader.close();  
}
```

O código apresentado pode lançar as seguintes exceções: `FileNotFoundException` e `IOException`. Solicita-se:



- Trate a exceção do tipo `FileNotFoundException` no método, caso seja lançada, a seguinte mensagem é solicitada a ser impressa pelo console: “O arquivo especificado não foi encontrado”.
- Trate a exceção do tipo `IOException` no método, caso seja acionada, a seguinte mensagem é solicitada a ser impressa pelo console: “Erro ao ler o arquivo especificado”.



## Exercício 4

Você precisa criar a classe `Calculos`, com uma série de métodos que permitem resolver operações matemáticas. De acordo com as seguintes instruções, crie os métodos correspondentes usando as operações disponíveis na classe `java.lang.Math`:

- Crie um método que tenha dois números como parâmetro e indique qual é o maior dos dois.
- Crie um método que tenha dois números como parâmetro e indique qual é o menor dos dois.
- Crie um método que calcule a potência de um número, ele receberá como parâmetro um valor numérico e a potência à qual deseja elevar esse número.
- Crie um método que calcule o cosseno de um ângulo.
- Crie um método que calcule a raiz quadrada de um número.
- Crie um método que calcule um número aleatório de 0 a 999;