

Disciplina: Programação Orientada a Objetos
Prof. Rodrigo R. Silva (rodrigorosa@ifsul.edu.br)
<https://github.com/Prof-Rodrigo-Silva>

Exercícios POO - Lista 12

1. Liste três exemplos de exceções comuns em JAVA.
2. Se nenhuma exceção é lançada em um bloco try, onde o controle de execução prossegue quando o bloco try termina a execução?
3. Qual a vantagem fundamental de utilizar catch?
4. O que acontece se nenhum catch for executado quando ocorrer um exceção?
5. Qual é a razão para utilizar blocos finally?
6. Para que serve a instrução throw?
7. Explique detalhadamente o try,catch,finally.
8. Num programa que lê um valor inteiro, faça a sua validação através do tratamento de uma exceção. Caso não seja inteiro continuar pedindo a introdução do valor inteiro.
9. Dado o código abaixo, verifique se o mesmo apresenta algum erro e caso seja necessário utilize o tratamento de exceções adequado.

```
package classe.executavel;

public class Teste {

    public static void main(String[] args) {

        int[] x = new int[10];

        for(int i = 0; i <= x.length;i++) {
            x[i] = i;
        }
        for(int i = 0; i <= x.length;i++) {
            System.out.println(x[i]);
        }
    }
}
```

10. Crie um programa que receba n números e some esses números enquanto a soma não for superior a 100. O programa deverá imprimir o valor somado (antes de atingir o número maior que 100) e deverá informar quantos números foram somados e qual a média. Refaça seu programa utilizando as seguintes regras:
 - a) Utilize os tratamentos de exceção para lidar com a entrada de dados.
 - b) Quando a soma for superior a 100, o programa deverá gerar uma exceção criada pelo programador, com nome `ExcecaoAcimaDeCem`.
 - c) Lance essa exceção, com o uso de `throws`.
11. Escreva um programa para preencher valores de um vetor de inteiros com 10 posições. O usuário irá informar os valores a serem inseridos e suas respectivas posições no array. O programa deve tratar as exceções `ArrayIndexOutOfBoundsException` quando for informada uma posição inexistente do vetor e `NumberFormatException` quando o valor informado não for número.
12. O código abaixo lança uma exceção e interrompe sua execução. Utilizando o tratamento de exceções, corrija a classe com o objetivo de não parar sua execução.

```
package classe.executavel;

public class Teste {

    public static void main(String[] args) {

        System.out.println("Inicio do main");
        metodo1();
        System.out.println("Fim do main");
    }

    static void metodo1() {
        System.out.println("Inicio do metodo1");
        metodo2();
        System.out.println("Fim do metodo2");
    }

    static void metodo2() {
        System.out.println("Inicio do main");
        int[] array = new int[10];
        for(int i = 0; i <= 15; i++) {
            array[i] = 1;
            System.out.println(i);
        }
        System.out.println("Fim do main");
    }
}
```

13. Identificar as partes problemáticas do código e reescrevê-lo utilizando tratamento de exceções. Ou seja, devem ser identificadas todas as exceções que podem ser levantadas e, para cada uma, deve ser dado o tratamento adequado que, nesse exercício, significa alertar

o usuário quanto ao problema. Entretanto, nesse programa a leitura dos valores deve ser feita, mesmo que para isso o usuário tenha que tentar informar várias vezes os valores na mesma execução do programa.

```
package classe.executavel;

import java.util.Scanner;

public class Divisao {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.println("Eu consigo dividir!");
        System.out.println("Informe o 1º valor: ");
        float x = teclado.nextFloat();
        System.out.println("Informe o 2º valor: ");
        float y = teclado.nextFloat();

        float r = x / y;
        System.out.println("O resultado é: "+r);
    }
}
```