

# Exercícios com tratamento de exceções

## Exercícios de exceções

1. Crie uma classe que aceite a digitação de dois números e faça a divisão entre eles exibindo seu resultado. Sua classe deve tratar as seguintes exceções:
  - `ArithmeticException` quando ocorrer uma divisão por zero.
  - `InputMismatchException` quando o valor informado não é numérico.
  
2. Crie uma classe que crie um vetor de inteiros de 10 posições. Feito isso, permita que o usuário digite valores inteiros a fim de preencher este vetor. Não implemente nenhum tipo controle referente ao tamanho do vetor, deixe que o usuário digite valores até que a entrada 0 seja digitada.

Uma vez digitado o valor 0, o mesmo deve ser inserido no vetor e a digitação de novos elementos deve ser interrompida. Feita toda a coleta dos dados, exiba-os em tela.

Sua classe deve tratar as seguintes exceções:

- `ArrayIndexOutOfBoundsException` quando o usuário informar mais que 10 valores.
- `InputMismatchException` quando o usuário informar um valor que não é numérico.

3. Crie uma classe Login com a seguinte modelagem:

Login

```
private String usuario; // Determina o nome do usuário
private String senha; // Determina a senha do usuário.
```

E os métodos:

```
public Login(String _usuario, String _senha) {
    // Construtor padrão
}

public void setSenha (String _senha) {
    // Troca a senha do usuário.
}

public boolean fazerLogin(String _usuario, String _senha){
    /*
        Deve receber informações de usuário e senha e compara-las com
        as da classe. Caso sejam realmente iguais, deve retornar verdadeiro,
        ou então deve lançar uma nova exceção dizendo qual credencial está
        errada, tratar essa exceção dentro do próprio método imprimindo o erro
        em tela e por fim retornar false.
        Ex:

        try{
            if(<<usuário incorreto>>) {
                throw new Exception("Usuário incorreto");
            }
        } catch (Exception e) {
            System.out.println("Erro");
        }
    */
}
```

Feito isso, crie uma classe para testar a inicialização de um objeto do tipo Login e que utilize o método fazerLogin, com informações digitadas pelo usuário.

4. O que será impresso se tentarmos compilar e executar a classe TesteExcecao?

```
class MinhaExcecao extends Exception {  
  
}  
  
public class TesteExcecao {  
    public static void teste() throws MinhaExcecao {  
        throw new MinhaExcecao();  
    }  
  
    public static void main(String[] args) {  
        MinhaExcecao me = null;  
        try {  
            System.out.println("try ");  
            teste();  
        } catch (MinhaExcecao e) {  
            System.out.println("catch ");  
            me = e;  
        } finally {  
            System.out.println("finally ");  
            throw me;  
        }  
  
        System.out.println("fim");  
    }  
}
```

- a) Imprime “try catch”
- b) Imprime “catch finally”
- c) Imprime “try catch finally ”
- d) Imprime “try catch finally fim”
- e) Erro de compilação

5. O código abaixo lança uma exceção (propositalmente) e interrompe sua execução. Utilizando o tratamento de exceções, corrija a classe com o objetivo de não parar sua execução. OBS:: A Exception lançada é `ArrayIndexOutOfBoundsException`.

```
public class TesteException {

    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 metodo1");
    }

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

6. Nesta questão você deve 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.

```
public class Questao2 {  
  
    public static void main(String[] args) {  
        Scanner teclado = new Scanner(System.in);  
        System.out.println("Eu sei dividir!");  
        System.out.print("Informe o primeiro valor: ");  
        int x = teclado.nextInt();  
        System.out.print("Informe o segundo valor: ");  
        int y = teclado.nextInt();  
        double r = (x / y);  
        System.out.println("O resultado da soma é " + r);  
    }  
}
```