



INSTITUTO DE  
INFORMÁTICA  
UFG

Universidade Federal de Goiás  
Instituto de Informática  
Disciplina: Algoritmos e Programação II  
Prof.: Paulo César Ferreira Melo  
E-mail: paulomelo@inf.ufg.br

## Lista de Exercícios – nº 9 Tratamento de Exceções

### Tipos de Exceções em Java

- Exception
- IOException
- FileNotFoundException
- ArrayIndexOutOfBoundsException
- ArithmeticException
- NullPointerException
- ClassCastException
- IllegalArgumentException

### Exercícios

1. Crie uma classe Divisao que possua o método calcular. Este método deve receber como parâmetro dois valores do tipo int e retornar o resultado da divisão. Contudo, você deve tratar a exceção de divisão por zero.
2. Escreva e teste um método robusto que imprime uma String contendo um inteiro lido. Use um tratador de exceção. Por exemplo:  
*Digite um inteiro: sete*  
*Este não é um inteiro numérico.*  
*Digite um inteiro: ok*  
*Este não é um inteiro numérico.*  
*Digite um inteiro: 123.45*  
*Este não é um inteiro numérico.*  
*Digite um inteiro: 123*  
*Você digitou: 123*
3. Escreva um programa que dispara e captura a exceção *ArrayIndexOutOfBoundsException*.



INSTITUTO DE  
INFORMÁTICA  
UFG

Universidade Federal de Goiás  
Instituto de Informática  
Disciplina: Algoritmos e Programação II  
Prof.: Paulo César Ferreira Melo  
E-mail: paulomelo@inf.ufg.br

## Lista de Exercícios – nº 9 Tratamento de Exceções

4. Faça um prognóstico sobre a saída do seguinte programa. Depois implemente o tratamento para a(s) referida(s) exceção(ões).

```
public class Ex4 {  
    public static void main(String[] args) {  
        String[] a = {"ABC", "123", "0", ""};  
  
        for(int i=0; i<=4; i++){  
            String s = a[i];  
            System.out.println("\ts: "+s);  
            int n = Integer.parseInt(s);  
            System.out.println("\ts: "+n);  
            int m = 1234/n;  
            System.out.println("\tm: "+m);  
            System.out.println("Adeus.");  
        }  
    }  
}
```

5. Na classe Conta, modifique o método que realiza o depósito, para que lance uma exceção chamada *IllegalArgumentException*, sempre que o valor passado como argumento for inválido (por exemplo, quando for negativo). Use comando **throw new**.

Para testar, crie uma classe TestaDeposito. Tente depositar valores inválidos na Conta e em uma ContaPoupança.

6. Adicione **try/catch** para tratar o erro do exercício anterior.
7. Crie sua própria *Exception*, denominada *ValorInvalidoException*. Para isso, crie sua classe que seja filha de *RuntimeException*.
- Coloque um construtor na classe *ValorInvalidoException* que receba valor inválido que ele tentou passar (isto é, ele vai receber um double valor);
  - Insira no construtor, uma mensagem, informando a exceção. Passe essa mensagem para a classe pai através do método `super()`;
  - Use o método `getMessage` para disparar a mensagem de exceção.
8. Se a classe *ValorInvalidoException* fosse filha direta de *Exception* ao invés de *RunTimeException* o que aconteceria uma vez que ela passaria para o tipo de exceção que são verificadas?