



# Projeto de Sistemas I

Faculdade Prof. Miguel Ângelo da Silva Santos

## Material 3 – Tratamento de exceção

Professor: Isac Mendes Lacerda, M.Sc., PMP, CSM  
e-mail: [isac.curso@gmail.com](mailto:isac.curso@gmail.com)

# Tópicos

- ▣ Tratamento de exceção

# Tratamento de exceção: para que serve?

- Quando um programa encontra **situações não previstas**, diz-se que ocorreu uma **condição excepcional** (ou uma *exceção*).
- Se a condição excepcional não é prevista (e tratada) pelo analista/programador, o programa termina com uma mensagem de rastreamento.

# Tratamento de exceção:

## exemplo 1

```
public static void divisao () {  
    int nr1 = Integer.parseInt(JOptionPane.showInputDialog("Número 1: "));  
    int nr2 = Integer.parseInt(JOptionPane.showInputDialog("Número 2: "));  
  
    float resultado = nr1 / nr2;  
    JOptionPane.showMessageDialog(null, resultado);  
}
```

E se digitar algo que não seja número?

# Tratamento de exceção:

## exemplo 1

```
public static void divisao () {  
    int nr1 = Integer.parseInt(JOptionPane.showInputDialog("Número 1: "));  
    int nr2 = Integer.parseInt(JOptionPane.showInputDialog("Número 2: "));  
  
    float resultado = nr1 / nr2;  
    JOptionPane.showMessageDialog(null, resultado);  
}
```

Saída - Excecao (run) X

```
run:  
Exception in thread "main" java.lang.NumberFormatException: For input string: "A"  
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)  
    at java.lang.Integer.parseInt(Integer.java:580)  
    at java.lang.Integer.parseInt(Integer.java:615)  
    at excecao.Excecao.divisao(Excecao.java:25)  
    at excecao.Excecao.main(Excecao.java:21)  
C:\Users\marina\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1  
FALHA NA CONSTRUÇÃO (tempo total: 4 segundos)
```

# Tratamento de exceção:

## exemplo 1

```
public static void divisao () {  
    int nr1 = Integer.parseInt(JOptionPane.showInputDialog("Número 1: "));  
    int nr2 = Integer.parseInt(JOptionPane.showInputDialog("Número 2: "));  
  
    float resultado = nr1 / nr2;  
    JOptionPane.showMessageDialog(null, resultado);  
}
```

Saída - Excecao (run) X

```
run:  
Exception in thread "main" java.lang.ArithmeticException: / by zero  
    at excecao.Excecao.divisao(Excecao.java:28)  
    at excecao.Excecao.main(Excecao.java:21)  
C:\Users\marina\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1  
FALHA NA CONSTRUÇÃO (tempo total: 6 segundos)
```

# Tratamento de exceção: o que houve?

- Tanto no caso da tentativa de conversão para *integer*, quanto na divisão por zero, Java não consegue resolver.
- O programa foi abortado indicando uma exceção específica (*NumberFormatException* ou *ArithmeticException*).

# Estrutura de tratamento

- Em Java as exceções são do tipo **Checked** (tratamento obrigatório antes da compilação) ou **Unchecked** (tratamento opcional antes da compilação). Em ambos os casos se pode usar a estrutura:

***try – catch .... catch - finally***



# Estrutura de tratamento

```
try {  
  <conjunto de instruções>  
}  
catch (Nome da exceção) {  
  <tratamento do erro 1>  
}  
catch (Nome da exceção) {  
  <tratamento do erro n>  
}  
finally {  
  <conjunto de instruções>  
}
```

***Quais são as possibilidades?***

# Tratamento de exceção:

## exemplo 1

```
try{
    int nr1 = Integer.parseInt(JOptionPane.showInputDialog("Número 1: "));
    int nr2 = Integer.parseInt(JOptionPane.showInputDialog("Número 2: "));

    float resultado = nr1 / nr2;
    JOptionPane.showMessageDialog(null, resultado);
}
catch (NumberFormatException erro){
    JOptionPane.showMessageDialog(null, "Digite apenas números inteiros!" +
        " O texto do erro é: " + erro.toString(), "Erro", JOptionPane.ERROR_MESSAGE);
}
catch (ArithmeticException erro){
    JOptionPane.showMessageDialog(null, "O denominador não pode ser 0!" +
        "o texto do erro é: " + erro.toString());
}
```

***Exemplo 1 tratado!***

# Estrutura de tratamento

```
try {  
    <conjunto de instruções>  
}  
catch (Exception erro) {  
    <tratamento genérico>  
  
}
```

***E se não souber  
o nome da  
exceção?***

# Estrutura de tratamento

```
try {  
    int y = x / 0;  
}  
catch (Exception erro) {  
    System.out.println(erro.getMessage());  
    erro.printStackTrace();  
}
```

***getMessage():*** Retorna a mensagem de uma exceção

***printStackTrace():*** Retorna o tipo de exceção

# Exercício 1

Use os métodos `getMessage()` e `printStackTrace()` para fazer o tratamento adequado para o programa abaixo:

```
x = 10;  
int n = 5 + "5";  
int [] nr = new int[2];  
nr[2] = 10;  
float r = 10 / 0;
```

# Exercício 2

Inclua um loop permanente para que o programa abaixo só termine quando o usuário informar valores corretos de entrada.

```
public static void divisao (){  
    int nr1 = Integer.parseInt(JOptionPane.showInputDialog("Número 1: "));  
    int nr2 = Integer.parseInt(JOptionPane.showInputDialog("Número 2: "));  
  
    float resultado = nr1 / nr2;  
    JOptionPane.showMessageDialog(null, resultado);  
}
```

# Exercício 3

Inclua tratamento de exceção em todos exercícios de ArrayList.