

POO - PROGRAMAÇÃO ORIENTADA A OBJETOS

Erros e Exceções

Rodrigo R Silva

Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense
Campus Bagé

Nesta Aula Veremos...

1 Erros e Exceções

Erros e Exceções

Tratando várias exceções

- Podemos adicionar várias cláusulas catch a um bloco try, tratando várias exceções ao mesmo tempo.

```
3 public class ClasseExcecoes {  
4  
5 public static void main(String[] args) {  
6  
7     int v1[] = {5,8,9,6,5,1,4};  
8     int v2[] = {2,0,4,0,8};  
9     for(int i = 0; i < v1.length; i++) {  
10        try {  
11            System.out.println(v1[i] + "/" + v2[i] + "=" + v1[i]/v2[i]);  
12        }catch (ArithmeticException e) {  
13            System.out.println("Erro de divisão por Zero - "+e);  
14        }catch (ArrayIndexOutOfBoundsException e) {  
15            System.out.println("Índice fora do intervalo do array -"+e);  
16        }  
17    }  
18 }
```

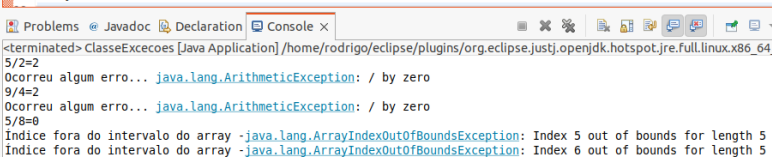
Problems @ Javadoc Declaration Console x

<terminated> ClasseExcecoes [Java Application] /home/rodrigo/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_5/2=2
Erro de divisão por Zero - [java.lang.ArithmeticException](#): / by zero
9/4=2
Erro de divisão por Zero - [java.lang.ArithmeticException](#): / by zero
5/8=0
Índice fora do intervalo do array -[java.lang.ArrayIndexOutOfBoundsException](#): Index 5 out of bounds for length 5
Índice fora do intervalo do array -[java.lang.ArrayIndexOutOfBoundsException](#): Index 6 out of bounds for length 5

Exceções de subclasses

- Para captura tanto de exceções de uma superclasse como de subclasses, primeiro devemos tratar a exceção da subclasse.

```
3 public class ClasseExcecoes {
4
5     public static void main(String[] args) {
6
7         int v1[] = {5,8,9,6,5,1,4};
8         int v2[] = {2,0,4,0,8};
9         for(int i = 0; i < v1.length; i++) {
10             try {
11                 System.out.println(v1[i] + "/" + v2[i] + "=" + v1[i]/v2[i]);
12             } catch (ArrayIndexOutOfBoundsException e) {
13                 System.out.println("Índice fora do intervalo do array -" + e);
14             } catch (Exception e) {
15                 System.out.println("Ocorreu algum erro... " + e);
16             }
17         }
18     }
19 }
```



Problems @ Javadoc Declaration Console x

<terminated> ClasseExcecoes [Java Application] /home/rodrigo/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_5/2=2
Ocorreu algum erro... [java.lang.ArithmeticException](#): / by zero
9/4=2
Ocorreu algum erro... [java.lang.ArithmeticException](#): / by zero
5/8=0
Índice fora do intervalo do array -[java.lang.ArrayIndexOutOfBoundsException](#): Index 5 out of bounds for length 5
Índice fora do intervalo do array -[java.lang.ArrayIndexOutOfBoundsException](#): Index 6 out of bounds for length 5

Bloco finally

- Sempre é executado, ocorrendo erros ou não.

```
public class ClasseExcecoes {  
    public static void main(String[] args){  
        int v1[] = {5,8,9,6,5,1,4};  
        int v2[] = {2,2,4,7,8,9};  
        for(int i = 0; i < v1.length; i++) {  
            try {  
                System.out.println(v1[i]+" / "+v2[i]+"="+v1[i]/v2[i]);  
            } catch (Exception e) {  
                StringBuilder saida = new StringBuilder();  
                e.printStackTrace();  
                System.out.println("Mensagem: "+e.getMessage());  
  
                for(int j = 0; j < e.getStackTrace().length; j++) {  
                    saida.append("\nClasse de erro: "+e.getStackTrace()[j].getClassName());  
                    saida.append("\nMétodo de erro: "+e.getStackTrace()[j].getMethodName());  
                    saida.append("\nMensagem de erro: "+e.getMessage());  
                    saida.append("\nLinha de erro: " + e.getStackTrace()[j].getLineNumber());  
                }  
                JOptionPane.showMessageDialog(null, saida);  
            } finally {  
                JOptionPane.showMessageDialog(null, "Ok, executou o finally!");  
            }  
        }  
    }  
}
```

Criando classes de Exceções

- Podemos criar nossas próprias classes de exceção.
- Para isso, devemos estender Exception, vejamos:

```
public class NumeroInvalidoException extends Exception{  
    private int numero;  
  
    public NumeroInvalidoException(int numero) {  
        this.numero = numero;  
    }  
  
    public String toString() {  
        return "O número (" + this.numero + ") informado é inválido";  
    }  
}
```

Lançando uma exceção

- Após definirmos o código de tratamento da exceção em uma classe, podemos utilizá-la em aplicativos.
- Para lançar uma exceção usamos **throw**.

```
 7 public class ClasseExcecoes {
 8
 9     public static void main(String[] args) {
10
11         int nro = Integer.parseInt(JOptionPane.showInputDialog("Informe um valor inteiro [1-10]: "));
12         try {
13             if(nro < 1 || nro > 10) {
14                 throw new NumeroInvalidoException(nro);
15             }
16         } catch (Exception e) {
17             System.out.println(e);
18         }
19     }
20 }
```

Problems @ Javadoc Declaration Console ×

<terminated> ClasseExcecoes [Java Application] /home/rodrigo/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x
0 número (-1) informado é inválido

Lançando mas tratando fora

- Podemos lançar uma exceção em um método, mas deixar seu tratamento para o método que invocou.
- Para isso, usamos **throws**.

```
6
7 public class ClasseExcecoes {
8
9     public static void main(String[] args) throws NumeroInvalidoException{
10
11         int nro = Integer.parseInt(JOptionPane.showInputDialog("Informe um valor inteiro [1-10]: "));
12         if(nro < 1 || nro > 10) {
13             throw new NumeroInvalidoException(nro);
14         }
15     }
16 }
```

Problems @ Javadoc Declaration Console ×

<terminated> ClasseExcecoes [Java Application] /home/rodrigo/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux:
Exception in thread "main" O número (11) informado é inválido
at classe.executavel.ClasseExcecoes.main(ClasseExcecoes.java:13)

OBRIGADO!