

Unidade 0 - Nivelamento - Tratamento de Exceção



PUC Minas

Instituto de Ciências Exatas e Informática
Departamento de Ciência da Computação

Exception

- Evento que ocorre durante a execução de um programa e deturpa o fluxo normal de instruções como, por exemplo:
 - Divisão por zero
 - Abrir um arquivo inexistente
 - Leitura de um tipo inesperado
 - Acesso a posição inválida de um *array*
 - Erro de hardware

TratamentoExcecao01.java

- Quando executamos `java TratamentoExcecao01 NÚMERO`, o programa funciona corretamente. Contudo, se não tivermos argumentos, temos uma *ArrayIndexOutOfBoundsException*

```
public class TratamentoExcecao01 {  
    public static void main(String[] args) {  
        Integer i = new Integer(args[0]);  
        MyIO.println("A variável i vale " + i);  
        MyIO.println("FIM DE PROGRAMA!!!");  
    }  
}
```

Try, Catch e Finally

```
try {  
    // bloco de código a ser executado. No caso de exceção,  
    // interrompemos imediatamente a execução desse bloco  
} catch (ClasseDaExceção instância) {  
    // código executado no caso de exceção  
} finally {  
    // código sempre executado, existindo ou não exceção  
    // Boa prática: cleanup code no bloco do finally  
}
```

TratamentoExcecao02.java

- Trata ArrayIndexOutOfBoundsException usando try / catch

```
public class TratamentoExcecao02 {  
    public static void main(String[] args) {  
        try {  
            Integer i = new Integer(args[0]);  
            MyIO.println("A variável i vale " + i);  
        } catch (ArrayIndexOutOfBoundsException e) {  
            MyIO.println("Erro na passagem de parâmetros!!!");  
        } finally {  
            MyIO.println("FIM DE PROGRAMA!!!");  
        }  
    }  
}
```

TratamentoExcecao02.java

- O que acontece quando executamos o comando `java`

TratamentoExcecao02 palavra?

```
public class TratamentoExcecao02 {  
    public static void main(String[] args) {  
        try {  
            Integer i = new Integer(args[0]);  
            MyIO.println("A variável i vale " + i);  
        } catch (ArrayIndexOutOfBoundsException e) {  
            MyIO.println("Erro na passagem de parâmetros!!!");  
        } finally {  
            MyIO.println("FIM DE PROGRAMA!!!");  
        }  
    }  
}
```

Múltiplas Exceções

- Podem ser tratadas em um mesmo bloco com vários catches
- Exemplo em [TratamentoExcecao03.java](#)

TratamentoExcecao03.java

```
public class TratamentoExcecao03 {  
    public static void main(String[] args) {  
        try {  
            Integer i = new Integer(args[0]);  
            MyIO.println("A variável i vale " + i);  
        } catch (NumberFormatException e) {  
            MyIO.println("Erro de formatação!!! ");  
        } catch (ArrayIndexOutOfBoundsException e){  
            MyIO.println("Erro na passagem de parâmetros!!!");  
        } finally {  
            MyIO.println("FIM DE PROGRAMA!!!");  
        }  
    }  
}
```


Throws: Passando a Exceção para Frente

- Outra forma de “tratar” as exceções é passando-as para frente (entende-se quem chamou o método corrente)

```
public void metodo() throws ClasseDeExceção, OutraClasse, ... {  
  
    ...  
  
}
```

- Exemplo em [TratamentoExcecao04.java](#)

TratamentoExcecao04.java

```

public class TratamentoExcecao04 {
    public static void m1(String s) throws NumFormExc, ArrayIndexOutOfBoundsException {
        Integer i = new Integer(args[0]); MyIO.println("A variável i vale " + i);
    }
    public static void m2(String s) throws NumberFormatException {
        Integer i = new Integer(args[0]); MyIO.println("A variável i vale " + i);
    }
    public static void main(String[] args) {
        try {
            m1(args[0]);
        } catch (NumberFormatException e) {
            MyIO.println("Erro!!!");
        } catch (ArrayIndexOutOfBoundsException e){
            MyIO.println("Erro!!!");
        }
        MyIO.println("--- >>>>> <<<<< -----");
        try {
            m2(args[0]);
        } catch (NumberFormatException e) {
            MyIO.println("Erro!!! ");
        }
        MyIO.println("--- >>>>> <<<<< -----");
    }
}

```

Throw: Criando Exceções

- Temos exceções porque as mesmas foram criadas usando a sintaxe abaixo:

```
public void metodo(...) {  
    if (...) {  
        throw new Exception();  
    }  
}
```

- Exemplo em [TratamentoExcecao05.java](#)

TratamentoExcecao05.java

```

public class TratamentoExcecao05 {
    public static void metodo(int i) throws Exception {
        if (i % 2 == 1){
            throw new Exception ("Valor impar");
        }
    }
    public static void main(String[] args) throws Exception {
        int i = 2;
        MyIO.println("----->>>>>>> <<<<<<<-----");
        metodo(i);
        MyIO.println("----->>>>>>> <<<<<<<-----");
        metodo(++i);
        MyIO.println("----->>>>>>> <<<<<<<-----");
    }
}

```