

OO

Tratamento de Erro

Profa.: Márcia Sampaio Lima

EST - UEA

Tratamento de Erros

- Um erro indica que há um problema com a interpretação do programa.
 - Tipos de Erro:
 - ❑ Erros de sintaxe;
 - ❑ Erros lógicos;
 - ❑ Exceções (erros de run-time).
-

Tratamento de Erros

■ Erros de sintaxe:

- Pode ocorrer um erro quando um arquivo é compilado.
- Exemplo:

- Pontos-e-vírgulas ausentes
 - Erros de ortografia
 - Atribuir um valor a uma variável que não é do tipo correto
-

Tratamento de Erros

■ Erros Lógicos

Erros lógicos podem ocorrer em decorrência de uma lógica incorreta do programador.

- Esses erros não produzem um erro de compilação ou de tempo de execução.
 - Por exemplo, um loop é executado muitas vezes, ou o programa produz um resultado incorreto.
-

Tratamento de Erros

■ Exceções

Quando um arquivo é compilado com sucesso, pode ocorrer um erro no momento de testá-lo no runtime.

Esses erros de runtime são chamados de exceções e devem ser tratados pelo programador usando código no programa, situação conhecida como “lançar” a exceção.

Tratamento de Erros

- Bloco Try/catch: usados para tratar exceções.

```
try{  
    ...code that might cause an exception  
}  
  
catch (exception e){  
    ...code to handle the exception  
}
```

Tratamento de Erros

Exceções Não Verificadas

O tratamento de exceções não verificadas em Java é opcional. No entanto, se a exceção não verificada não for tratada e ocorrer um erro, o programa será interrompido.

Exceções não verificadas comuns:

- Índice de exceção fora dos limites
 - Exceção de ponteiro nulo
-

Tratamento de Erros

Índice de Exceção Fora dos Limites

Chame `example[3]` no seguinte array inicializado.

```
int[] example = {1, 2, 3};
```

A mensagem de exceção a seguir será exibida porque `example[3]` não existe no array.

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3  
em NullPointerException.main(NullPointerException.java:6)
```

Exemplo:

```
public class TratamentoErro {  
  
    public static void main(String[] args) {  
        int numeros[] = {1,2,3};  
  
        System.out.println(numeros[0]);  
        System.out.println(numeros[1]);  
        System.out.println(numeros[2]);  
        //Lanca Exceção não verificada  
        System.out.println(numeros[3]);  
  
        System.out.println("Terminou o programa!!!");  
    }  
}
```

- Como tratar:

- Usando blocos try/catch()

- Objetivo:

- Tratar a exceção

- Permitir a continuação da execução do programa

Exemplo:

```
public class TratamentoErro {  
  
    public static void main(String[] args) {  
        int numeros[] = {1,2,3};  
  
        System.out.println(numeros[0]);  
        System.out.println(numeros[1]);  
        System.out.println(numeros[2]);  
        Try{  
            System.out.println(numeros[3]);  
        } catch(Exception e){ //implementa o tratamento  
        }  
        System.out.println("Terminou o programa!!!");  
    }  
}
```

Tratamento de Erros

Tratando Exceções Verificadas

- Usar um bloco try/catch, que tratará da exceção “graciosamente”.
 - Usar uma instrução “throws” na declaração do método, o que é um risco. Uma instrução throws é quando um programador diz que correrá o risco de uma exceção não ser gerada.
-

Tratamento de Erros

■ Tratando Exceções Verificadas

Exceção de E/S

Este código usa um bloco try/catch para tratar de uma exceção de E/S.

```
try{
    FileReader reader = new FileReader("test.txt");
}
catch(IOException e){
    System.out.println("File not found");
}
```

Exemplo:

```
public class TratamentoErro {  
  
    public static void main(String[] args) {  
        int numeros[] = {1,2,3};  
        .....  
        Try{  
            System.out.println(numeros[3]);  
        } catch(Exception e){ //implementa o tratamento  
        }  
  
        ■ FileReader reader = new FileReader("Teste.txt");  
        ■ reader.close();  
    }  
}
```

Exemplo:

```
public class TratamentoErro {  
  
    public static void main(String[] args) {  
        int numeros[] = {1,2,3};  
        .....  
        Try{  
            System.out.println(numeros[3]);  
        } catch(Exception e){ //implementa o tratamento  
        }  
  
        ■ FileReader reader = new FileReader("Teste.txt");  
        ■ reader.close();  
    }  
}
```

FileNotFoundException

IOException

Tratamento de Erro

- Como tratar Exceções Verificadas:
 - Usando blocos try/catch()
 - Objetivo:
 - Tratar a exceção
 - Permitir a continuação da execução do programa
 - Esse tipo de exceção o uso do try/catch() é obrigatório.
-

Exemplo:

```
public class TratamentoErro {  
  
    public static void main(String[] args) {  
        int numeros[] = {1,2,3};  
  
        ....  
        Try{  
            FileReader reader = new FileReader("Teste.txt");  
            reader.close();  
        } catch(Exception e){ //implementa o tratamento  
            System.out.println("Erro na abertura do arquivo");  
        }  
        .....  
    }  
}
```

Tratamento de Erros

■ Testar:

```
int i = 5571;
```

```
i = i / 0;
```

```
System.out.println("Resultado de i: " + i);
```

Tratamento de Erros

■ Usando instrução ***throws***

- ❑ Usa uma instrução “throws” para tratar uma Exceção de E/S.
 - ❑ A instrução “throws” é usada para alertar que pode haver uma exceção gerada.
 - ❑ No entanto, o programa ainda será interrompido caso ocorra um erro que gere uma exceção.
-

```
public static void main(String[] args) throws IOException{  
    FileReader reader = new FileReader("test.txt");  
}
```

Exemplo

```
public class Calculo {  
  
    int numeros[] = {1,2,3,4};  
  
    public void mostrarNumeros() throws Exception{  
        for (int i =0; i < 5; i ++){  
            System.out.println(numeros[i]);  
        }  
    }  
}
```

```
public class Execucao {  
  
    public static void main(String[] args) {  
        Calculo obj = new Calculo();  
        try {  
            obj.mostrarNumeros();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Lançando Exceções

- Podemos tratar de exceções lançando-as.
- Ao lançar uma exceção, o mecanismo de interpretação interromperá a execução do programa nesse ponto.
- No código, uma exceção é gerada da seguinte forma:

```
throw new Exception("Array index" + i + " is out of bounds!");
```

Lançanco

```
public class Calculo {  
  
    int numeros[] = {1,2,3,4};  
  
    public void mostrarNumeros() throws Exception {  
  
        for (int i = 0; i < 5; i ++){  
            if (i > 3){  
                throw new Exception ("Indice do array esta fora dos limites!!!");  
            }else{  
                System.out.println(numeros[i]);  
            }  
        }  
    }  
}
```

Tratando

```
public class Execucao {  
  
    public static void main(String[] args) {  
        Calculo obj = new Calculo();  
  
        try {  
            obj.mostrarNumeros();  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

Tratamento de Erros

Uma pequena parte da Família Throwable:

