

Programação III

Exceções
Jocélio Passos
joceliopassos@bol.com.br

Exceções - Conceitos

- Sebesta cap 13 e Deitel cap 14
- Erro que ocorre no decorrer do programa, ou seja, em tempo de execução
- Além das exceções predefinidas pelo Java, podemos criar as nossas próprias exceções
- Com exceções os programas se tornam mais:
 - Claros
 - Robustos
 - Tolerantes a falhas

2/2/2007

2

Exceções - Conceitos

- O tratamento de exceções em Java é semelhante àquele utilizado em C++
- Deitel 653 [740]
- Sebesta 523-533

2/2/2007

3

Um programa sem Tratamento de Exceção

- `import jutil.*;`
- `class DivisaoPorZeroTeste{`
- `public static void main(String[] args)`
- `{int x,y;`
- `JUtil.mostrln("Erro de Divisao por Zero !");`
- `JUtil.mostrln("Sem controle de Excecao");`
- `x = 10;`
- `y = 0;`
- `JUtil.mostrln ("Dividindo x = " + x + " por y = " + y);`
- `JUtil.mostrln(" = " + (x/y));`
- `JUtil.mostrln ("nFim do programa"); // Esta linha nunca será executada }`
- `Java.lang.ArithmeticException : / by zero`

2/2/2007

4

Um programa com Tratamento de Exceção

- `import jutil.*;`
- `class DivisaoPorZeroTeste2`
- `{public static void main(String[] args) throws Exception`
- `{int x,y;`
- `JUtil.mostrln("Erro de Divisao por Zero !");`
- `JUtil.mostrln("Com controle de Excecao");`
- `x = 10;`
- `y = 0;`
- `JUtil.mostrln ("Dividindo x = " + x + " por y = " + y);`
- `try { JUtil.mostrln(" = " + (x/y));}`
- `catch (ArithmeticException msg) {`
- `JUtil.mostrln ("n Denominador nao pode ser zero ! \n " + msg);}`
- `finally {JUtil.mostrln ("n Programa terminado ! \n ");}`
- `JUtil.pause ();}`

2/2/2007

5

Comando de Manipulação de Exceção

- **throws Exception** - Deixa o bloco de sobre aviso que o código seguinte pode gerar uma exceção Deitel 664 [751]
- **Try** - Envolve o bloco que pode gerar uma exceção
- **Catch** - Captura e trata uma exceção
- **Finally** - Código sempre executado independentemente de uma exceção ocorrer
- Deitel 656

2/2/2007

6

Criando uma Classe Exceção

```
■ class DivisaoPorZeroException extends
  ArithmeticException {
■ public DivisaoPorZeroException (String msg)
■ {super (msg);}

■ public DivisaoPorZeroException ()
■ {super ("Denominador nao pode ser zero !");}
■ }
```

2/2/2007

7

Lançando uma Exceção

```
■ import junit.*;
■ class DivisaoPorZeroTeste3 {
■ public static void main(String[] args) throws
  Exception
■ {int x,y;
■ JUtil.mostreln("Erro de Divisao por Zero !");
■ JUtil.mostreln("Com controle de Excecao");
■ x = 10;
■ y = 0;
■ JUtil.mostr ("Dividindo x = " + x + " por y = " + y );
```

2/2/2007

8

Lançando uma Exceção

```
■ try
■ {JUtil.mostr ("Resultado = " + divisao (x,y) );}
■ catch (DivisaoPorZeroException dpze)
■ {JUtil.mostreln ("\\n " + dpze.getMessage ());}
■ finally
■ {JUtil.mostreln ("\\n Programa terminado ! \\n ");}
■ JUtil.pause ();
■ }
```

2/2/2007

9

Lançando uma Exceção

```
■ public static double divisao (float x, float y)
  throws DivisaoPorZeroException
■ { if ( y == 0)
■ { throw new DivisaoPorZeroException (); }
■ return (x / y); }
■ }
```

2/2/2007

10

Comando de Manipulação de Exceção (Continuação)

```
■ throw - Lança uma exceção

■ Formato Geral :
■ throw new ClassException ();

■ Exemplo :
■ throw new RuntimeException ();
■ throw new ArrayIndexOutOfBoundsException ();
```

2/2/2007

11

Tratamento de Exceção : Resumo

```
■ try{
■ // Bloco Protegido}
■ catch (ClassException ce) {
■ //Bloco catch}
■ finally {
■ // Bloco finally}
```

2/2/2007

12

Manipulando Exceções

- 1) Crie um programa com base na Classe LerCliente (pacote) . Neste programa use uma classe chamada `StringVaziaException`. Esta classe é uma herança de `Exception` e deve enviar uma mensagem informando que nenhuma informação foi digitada.

2/2/2007

13

Exercícios Propostos

- 1) O hardware é capaz de detectar certas condições de erro em tempo de execução. Quais? [Sebesta 508](#), [Deitel 653 \[740\]](#) (último parágrafo)
- 2) O que acontece quando a linguagem de programação não é capaz de tratar uma exceção? [Sebesta 508](#)
- 3) Por que o C não verifica as faixas de subscrito? [Sebesta 508](#)
- 4) Por que muitas linguagens não incluem manipulação de exceções em seus recursos? [Sebesta 509](#)
- 5) O tratamento de algumas exceções pode ser desativado durante algum tempo? [Sebesta 509](#)

2/2/2007

14

Exercícios Propostos

- 6) Que vantagens temos em ter manipulações de exceções incorporadas na linguagem? [Sebesta 510](#)
- 7) Qual a primeira linguagem que permitiu o controle de exceções pelo usuário? [Sebesta 513](#)
- 8) Onde deve ser colocado o código de tratamento de erros? Há desvantagens? [Deitel 653 \[740\]](#)
- 9) Que tipo de eventos não são manipulados como exceção? [Deitel 654 \(3º parágrafo\)](#) [\[741 1º parágrafo\]](#)
- 10) Quando deve ser utilizado o tratamento de exceções? [Deitel 655 \[742\]](#)
- 11) Quando não deve ser utilizado o tratamento de exceções? [Deitel 655 \[742\]](#)

2/2/2007

15

Exercícios Propostos

- 12) Em relação a desempenho como se comporta o tratamento de exceções? [Deitel 656 \[742\]](#) (Dica de desempenho 14.3 [\[14.2\]](#))
- 13) Como é o comportamento de uma exceção não tratada em programas console (DOS) e em programas gráficos (GUI) como applets? [Deitel 656 \[743\]](#) (Dica de teste e de depuração 14.2)
- 14) Que comandos podemos colocar no bloco `finally`? [Deitel 656 \[742-743\]](#)
- 15) O que são pontos de disparo? [Deitel 656 \[742-743\]](#)
- 16) Estude o pacote `java.lang` e as exceções pré-definidas do java. [Deitel 665-668 \[753-756\]](#)
- 17) Qual o equivalente a `throw` no Pascal/Delphi/ADA? [Sebesta 524](#)

2/2/2007

16