

Linguagem Orientada a Objetos

Tratamento de exceções

Prof. Ms. Leonardo Rocha

2

- Unidade de Ensino: 3
- Competência da Unidade: Compreender o funcionamento e aplicação dos tratamentos de exceções.
- Palavras-chave: Aplicadores de formato, tratamento de exceções
- Título da Teleaula: Tratamento de exceções
- Teleaula nº: 3

Contextualização

Entrada de dados e leitura de valores
Especificadores de formato
Tratamento de exceções
Documentação e interfaces
Aplicações gráficas e suporte a GUI (Graphical User Interface)

3

Entrada de dados ou leitura de valores

Primeiro....

Existem duas formas principais para se interagir com uma aplicação.

- 1ª - acontece passando-se argumentos no momento da execução do programa.
- 2ª - se dá por meio da leitura de valores em tempo real.

5

Entendendo

```

public class ArgvLinhaDeComandoBasico {
    public static void main(String[] args) {
        System.out.printf("qtd de argumentos = %d\n",
args.length);
        for (int i = 0; i < args.length; i++) {
            System.out.printf("\targs[%d] = %s\n", i, args[i]);
        }
    }
}

```

Assinatura do ponto de entrada da aplicação

String[] - Vetor de strings que permite a passagem de quantos argumentos quisermos

args - parâmetro que recebe valores no momento da execução do código.

↑ ↑
Especificadores de formato

6

Sequências de scape

7

Sequência de Escape	Descrição	Exemplo de utilização
\n	Insere nova linha.	System.out.print("Introdução à programação\nmain.java");
\t	Insere tabulação na horizontal.	System.out.print("col A\tcol B\tcol C\tcol D\t");
\\	Insere barra invertida.	System.out.print("C:\\Windows\\system32");
\"	Insere aspa dupla.	System.out.print("Nome do livro \"Dom Quixote\" de Miguel de Cervantes");
\r	Realiza retorno de carro.	System.out.print("Texto Não Mostrado \rEste texto Apareceu");

Especificadores de formato

8

Especificador	Descrição	Exemplo de Utilização	Saída
%d	Valor inteiro em decimal com sinal (pode ser usado para <i>byte</i> , <i>short</i> , <i>int</i> e <i>long</i>).	System.out.println("na", 127);	127
%o	Valor inteiro em octal com sinal.	System.out.println("no", 127);	177

Especificadores de formato

9

Especificador	Descrição	Exemplo de Utilização	Saída
%e	Valor inteiro em hexadecimal com sinal (minúsculo).	System.out.println("e", 127);	7f
%E	Valor inteiro em hexadecimal com sinal (maiúsculo).	System.out.println("E", 127);	7F
%f	Valor real (float ou double).	System.out.println("f", 0.145);	0.145
%a	Valor real (notação exponencial) (minúsculo).	System.out.println("a", 0.145);	1.45E-01
%A	Valor real (notação exponencial) (maiúsculo).	System.out.println("A", 0.145);	1.45E-01
%g	Valor inteiro em hexadecimal com sinal (minúsculo).	System.out.println("g", 127);	7f
%G	Valor inteiro em hexadecimal com sinal (maiúsculo).	System.out.println("G", 127);	7F
%b	Valor lógico (boolean) (minúsculo).	System.out.println("b", 8);	true
%B	Valor lógico (boolean) (maiúsculo).	System.out.println("B", 8);	TRUE

Exemplo - especificador de formato

10

```

1 //
2 * @Leonardo Rocha
3 */
4
5 public class DefinePeso {
6     public static void main(String[] args) {
7
8         String nome = "Leonardo";
9         int idade = 30;
10        double peso = 55.5;
11        System.out.printf("%s, %d anos, pesa %.2f kg!", nome, idade, peso);
12    }
13 }
14

```

Estrutura de pasta - visão geral

11

Diretório	Descrição do Conteúdo
src/	Código-fonte da aplicação.
test/	Código de teste unitário (não utilizado neste livro).
lib/	Dependências do projeto.
dist/	Arquivos de distribuição como .jar e suas dependências.
build/	Arquivos gerados pelo processo de compilação.

Leitura de dados em tempo real

12

A leitura de dados em tempo real pode ser feita via classe Scanner, que fornece métodos de leitura com sintaxes diferentes.

```

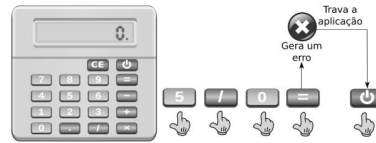
import java.util.Scanner;

public class ExemploLeituraDados {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Digite um valor inteiro (int): ");
        int entradaInt = scan.nextInt();
        System.out.print("Digite um valor real (double): ");
        double entradaDouble = scan.nextDouble();
        System.out.print("Digite um valor lógico (boolean): ");
        boolean entradaBoolean = scan.nextBoolean();
        System.out.print("Digite uma string (uma palavra): ");
        String entradaPalavra = scan.next();
    }
}

```

Tratamento de exceções

Entendendo as exceções



Imagine o comportamento atípico de uma aplicação calculadora que trava ao executar um cálculo como o mostrado na figura.

Tratando as exceções



Verificação de cálculo indevido e exibição de mensagem para evitar a exceção.

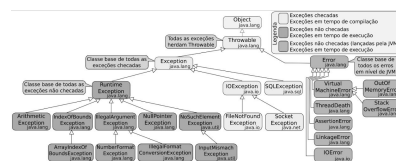
Trechos de códigos que lançam exceções

	Linhas de Código	Tipo de Exceção Lançada
1	<code>int divPor0 = 5/0;</code>	ArithmeticException
2	<code>int valorStr = Integer.parseInt("A");</code>	NumberFormatException
3	<code>int refNull = Integer.parseInt(null);</code>	NullPointerException
4	<code>String indiceNegativo = args[-1];</code>	ArrayIndexOutOfBoundsException
5	<code>System.out.printf("%d", "5");</code>	IllegalFormatConversionException
6	<code>Scanner sc = new Scanner(System.in);</code>	InputMismatchException

Tratamento

```
1 try {
2     //Bloco de código a ser executado
3 } catch (ClassEx e){
4     //Bloco para tratar a exceção
5 }
6
7 public static int div(int a, int b) {
8     try {
9         return a / b;
10    } catch (ArithmeticException ex) {
11        System.err.println("A divisão por zero é indefinida");
12        return 0;
13    }
14 }
```

Exceções



Definição

25

Construir um código legível e bem documentado é extremamente importante, pois ele necessitará de atualizações e poderá ser lido por outros programadores. A forma mais simples de documentar um código se dá por meio de comentários.

Tipos de comentários

26

Uma única linha - utiliza-se o comando `//` para inserção do comentário;

Em blocos - utiliza-se o comando `/* conteúdo */` para inserção de comentários.

For Each

27

é utilizado, geralmente, para se fazer a iteração em coleções como vetores, matrizes, listas, entre outros.

Exemplo

```
int soma = 0;
for (int i = 1; i < 6; i++) {
    soma += i;
}

int colecao[] = {1, 2, 3, 4, 5};
int soma = 0;
for (int elem : colecao) {
    soma += elem;
}
```

For

For Each

Argumentos variáveis - Varargs

28

A linguagem Java nos permite chamar um método diretamente passando *n* valores e os parâmetros enviados são automaticamente adicionados em um Array de mesmo tipo. Um varargs

Varargs - Exemplo

29

```
1 modif_acesso tipo_retorno nomeMetodo(TipoArg... args){
2     SeqBeConandos;
3 }
```

```
1 modif_acesso tipo_retorno nomeMetodo(TipoArg arg, TipoArg...
2     args){
3     SeqBeConandos;
4 }
```

1º Ex.: é caracterizado por três pontos (...) depois do tipo de argumento.

2º Ex.: um argumento normal seguido de um argumento do tipo varargs.

Interfaces

30

Algumas interfaces implementadas:

Comparable: é utilizada para impor uma ordem nos objetos de uma determinada classe que a implementa.

Runnable: é utilizada para especificar alguma tarefa a ser realizada.

Serializable: é utilizada para identificar classes em que os objetos podem ser gravados (também chamados de serializados) ou lidos (também chamados de desserializados) de algum dispositivo de armazenamento, como HD.

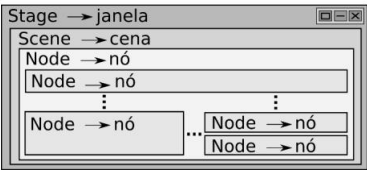
Aplicações gráficas

JAVAFX

A linguagem Java possui um suporte muito bom para a criação de Interfaces Gráficas de Usuário ou, em inglês, Graphical User Interface (GUI), e existem várias formas de se fazer a criação de interfaces gráficas em Java. Ao longo da evolução dessa linguagem, diversas bibliotecas gráficas foram criadas, como:

Abstract Window Toolkit (AWT), Swing, Standard Widget Toolkit (SWT), Apache Pivot, SwingX, JGoodies, QtJambi e JavaFX.

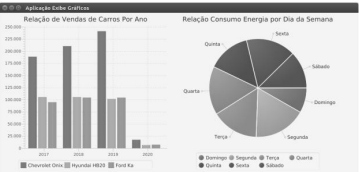
Estrutura



Componentes JAVAFX

Nome do Componente	Classe do Componente	Descrição do Componente	Aspecto Visual do Componente
Botão	Button	Simple botão de controle.	
Rótulo	Label	Campo que permite exibir um texto não editável.	
Campo de Texto	TextField	Campo que permite a inserção de texto em uma única linha.	
Área de Texto	TextArea	Campo que permite a inserção de texto com múltiplas linhas.	
Botão de Opção	RadioButton	Seleção de opções mutuamente excluídas.	
Caixa de Seleção	CheckBox	Seleção de opções não mutuamente excluídas.	
Caixa de Combinação	ComboBox	Seleção de opções mutuamente excluídas em forma de lista.	

Gráficos



Solicitar palavra do usuário e tratar exceção

```
1 package atividadespl.umd3;
2
3 public class AtividadeSPIUnd3 {
4
5
6     public static void main(String[] args) {
7         diminuiLetras ltpqn = new diminuiLetras();
8
9         ltpqn.lowCase();
10
11         try {
12             diminuiLetras ltpqn = new diminuiLetras();
13             ltpqn.lowCase();
14         } catch (NullPointerException e) {
15             System.out.println("Exceção NullPointerException identificada");
16         }
17         System.out.println("Metodo diminuiLetras() tratado "+e);
18     }
19 }
20 }
```

Solicitar palavra do usuário e tratar exceção

37

```
1 package atividadespl.und3;
2 import javax.swing.JOptionPane;
3
4 public class diminueletras {
5     public static void lowCase() throws NullPointerException
6     {
7         String word = null;
8         String newWord = null;
9
10
11         word = JOptionPane.showInputDialog(null, "Digite uma palavra: ");
12
13         newWord = word.toLowerCase();
14         System.out.println("A palavra informada é: "+word);
15         System.out.println("A palavra corrigida é: "+newWord);
16     }
17 }
```

Recapitulando

Entrada de dados e leitura de valores
Especificadores de formato
Tratamento de exceções
Documentação e interfaces
Aplicações gráficas e suporte a GUI (Graphical User Interface)

39

