

Java: Interface Gráfica

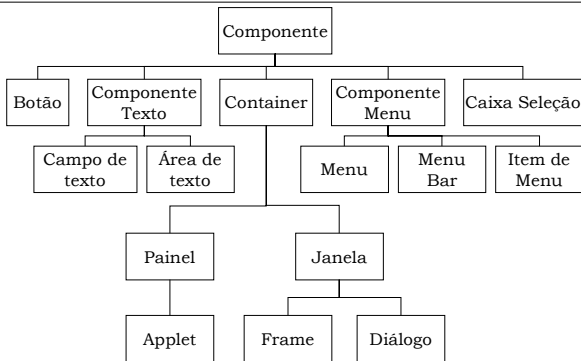
Exemplos de código em Java você pode encontrar em:

<http://java.sun.com/docs/books/tutorial/uiswing/components/example-swing/index.html>

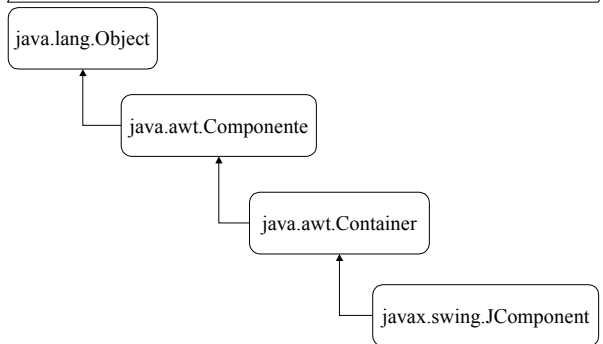
Bibliografia

- ▶ H. M. Deitel; P.J. Deitel. Java: como programar. Bookman, 2001.
- ▶ R. W. Sebesta. Conceitos de linguagens de programação. Bookman, 2000.
- ▶ K. Jamsa; L. Klander. Programando em c/c++: "a bíblia". Makron, 1999.
- ▶ K. C. Hopson; S. E. Ingram. Desenvolvendo applets com java. Campus, 1997.
- ▶ Tim Ritchey. Programando com java. Campus, 1997.
- ▶ Alexandre Newman. Usando java. Campus, 1997.
- ▶ Colin Fraizer. Api java: manual de referência. Makron, 1997.
- ▶ P. Coad; M. Mayfield. Projeto de sistemas em java. Makron, 1998.
- ▶ André Valle; Cláudia Guimarães. Java: manual de introdução. Axcel Books, 1996.

Parte da Hierarquia do pacote awt



Hierarquia do pacote swing (Pacote javax.swing)

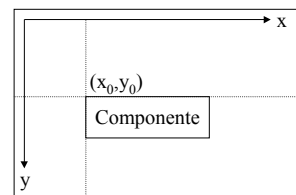


O que é o pacote awt?

- ▶ Pacote java.awt:
 - ▶ Awt (Abstract Windowing Toolkit) oferece uma API para componentes comuns de interface com o usuário do sistema.
- ▶ Pacote javax.swing:
 - ▶ Disponibiliza classes para criar componentes GUI (Grafic User Interface).
 - ▶ Os componentes do pacote swing são chamados de componentes "Java puro" pelo fato de serem escritos, manipulados e exibidos completamente em Java.

Container

- ▶ Um container é um componente que pode armazenar outros componentes;
- ▶ A utilização de componentes gráficos exige que que esses componentes sejam disponibilizados em containers.



* Frames (janelas)

import javax.swing.; //pacote para componentes gráficos*

```
public class FormularioPadrao extends JFrame
{
    public FormularioPadrao ()
    {
        this.setSize(200, 200); //largura, altura
    }
}
```



12/08/2002

7

* Frame: Métodos

- ▶ 2 construtores:
 - ▶ Frame(): sem título;
 - ▶ Frame("Título"): ao construir define um título.
- ▶ resize(largura, altura): define a largura e altura do frame;
- ▶ hide(): esconde o frame;
- ▶ dispose(): destroi o frame;
- ▶ setTitle(): define um título para o frame;
- ▶ getTitle(): retorna o título do frame;
- ▶ setResizable(boolean): define se o usuário pode alterar dimensões do frame;
- ▶ setBackground(Color): define uma nova cor para o frame;
- ▶ janela.setLocation(200,100): define a localização, onde 200 indica a posição topo e 100 a posição direita.

12/08/2002

8

* Container

- ▶ A posição do componente são relativas ao canto superior esquerdo do container;
- ▶ Containers fornecem uma área retangular na qual os componentes são exibidos;
- ▶ Cada container e cada componente possui o seu próprio sistema de coordenadas relativas;
- ▶ Quando uma operação gráfica é efetuada dentro de um componente, são utilizadas as coordenadas do componente e não do container;
- ▶ A posição de um componente dentro de um container depende do layout adota por este.

12/08/2002

9

* Container: layout

- ▶ Os componentes são inseridos nos layouts utilizando-se do método add();
- ▶ Existem três métodos básicos para lidar com o posicionamento e dimensionamento dos componentes dentro do container:
 - ▶ move: move o componente para uma nova localização x,y no container: *public void move (x,y);*
 - ▶ resize: redimensiona o componente para a largura e altura especificadas: *public void resize (l,a);*
 - ▶ reshape: move o componente para a nova localização e redimensiona para as dimensões especificadas: *public synchronized void reshape (x, y, l, a);*

12/08/2002

10

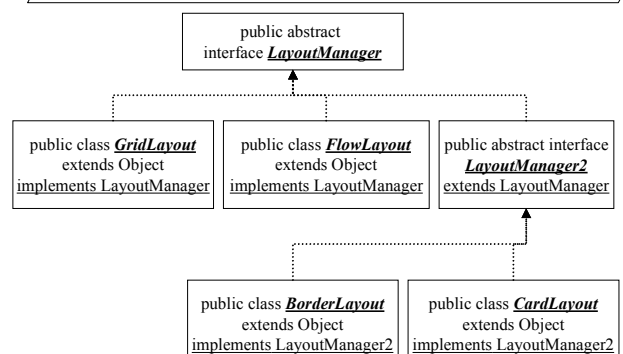
* Layouts (PACKAGE JAVA.AWT)

- ▶ O posicionamento do componente dentro do container é determinado por seu gerenciador de layout;
- ▶ Caso não seja especificado um layout, a maior parte dos containers especifica o BorderLayout como sendo o default;
- ▶ O gerenciador de layout é uma classe que implementa a interface LayoutManager;

12/08/2002

11

* LayoutManager: subclasses



12/08/2002

12

Classe FlowLayout

- ▶ Este layout determina que os componentes sejam desenhados da parte superior esquerda para a direita até que seja atingido o limite horizontal do container;
- ▶ Construtores:
 - » `public FlowLayout ()`;
 - » `public FlowLayout (int align)`;
 - » `public FlowLayout (int align, int hgap, int vgap)`.
- ▶ Onde:
 - » `int align`: alinhamento dos componentes;
 - » `hgap`: espaço horizontal entre os componentes;
 - » `vgap`: espaço vertical entre os componentes.

12/08/2002

13

Classe FlowLayout

- ▶ Caso seja utilizado o construtor sem parâmetros, o alinhamento default é ao centro e espaço de 5 unidades tanto na vertical como na horizontal;
- ▶ O valor para o parâmetro `align` precisa ser um entre: `FlowLayout.LEFT`, `FlowLayout.RIGHT` ou `FlowLayout.CENTER`;

Classe exemplo: `LayoutContainer.java`

12/08/2002

14

Exemplo: FlowLayout ...

```
import java.awt.Frame;
import java.awt.FlowLayout;
import java.awt.Button;
import java.awt.event.*;

public class LayoutContainer extends Frame
{
    public LayoutContainer ()
    {
        FlowLayout meuLayout = new FlowLayout(FlowLayout.CENTER);
        //outros valores: FlowLayout.LEFT, FlowLayout.RIGHT;

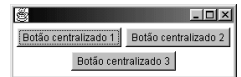
        setLayout(meuLayout);
    }
    ...
}
```

12/08/2002

15

Exemplo: FlowLayout

```
public static void main (String args [] ) {
    LayoutContainer janela = new LayoutContainer();
    Button botao1 = new Button("Botão centralizado 1");
    Button botao2 = new Button("Botão centralizado 2");
    Button botao3 = new Button("Botão centralizado 3");
    janela.add(botao1);
    janela.add(botao2);
    janela.add(botao3);
    janela.setSize(300,200);
    janela.show();
    janela.addWindowListener (
        new WindowAdapter() {
            public void windowClosing (WindowEvent e)
            {
                System.exit(0);
            }
        }
    );
}
```

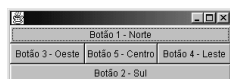


12/08/2002

16

Classe BorderLayout

- ▶ Esse gerenciador de layout dispõe os componentes utilizando os pontos cardeais para a definição da posição do componente;
- ▶ Construtor:
 - » `BorderLayout meuLayout = new BorderLayout()`;
- ▶ Definindo a posição dos botões:
 - » `add("North", botao1)`;
 - » `add("South", botao2)`;
 - » `add("West", botao3)`;
 - » `add("East", botao4)`;
 - » `add("Center", botao5)`;



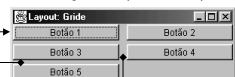
Classe exemplo: `LayoutBordaContainer.java`

12/08/2002

17

Classe GridLayout

- ▶ Dispõe os componentes em colunas e linhas;
- ▶ Todos os componentes possuem um único tamanho.
- ▶ Construtor:
 - » `GridLayout meuLayout = new GridLayout (3,2,8,5)`;
 - » onde:
 - » 3: número de linhas;
 - » 2: número de colunas;
 - » 8: espaçamento horizontal;
 - » 5: espaçamento vertical.



Classe exemplo: `LayoutGridContainer.java`
Classe exemplo: `OutroExemploJanela.java`

12/08/2002

18

Rótulos: classe ...

```
import javax.swing.*; import java.awt.*; import java.awt.event.*;
public class FormRotulo extends JFrame
{
    private JLabel rotulos[ ];
    public FormRotulo (String titulo) {
        super(titulo);
        rotulos = new JLabel[3];
        Container areaForm = getContentPane();
        areaForm.setLayout(new GridLayout(rotulos.length,1));
        for (int i = 0; i < rotulos.length; i++) {
            rotulos[i] = new JLabel("Rotulo "+(i+1));
            rotulos[i].setForeground(Color.black);
            areaForm.add(rotulos[i]);
        }
        setSize(250,80);
        show();
    }
}
```

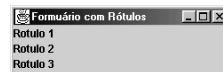
12/08/2002

19

Rótulos: main

```
public static void main (String args[ ])
{
    FormRotulo formRotulo = new FormRotulo("Formuário com Rótulos");

    formRotulo.addWindowListener (new WindowAdapter () {
        public void windowClosing (WindowEvent e) {
            System.exit(0);
        }
    });
} //chave da classe FormRotulo
```



Código exemplo: FormRotulo.java

12/08/2002

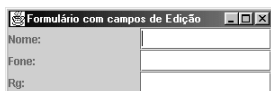
20

Campo de texto (Edits)

```
private JLabel rotulos [];
private JTextField edits [];

...
rotulos = new JLabel[3];
edits = new JTextField[3];

...
for (int i = 0; i < 3; i++)
{
    rotulos[i] = new JLabel(nomes[i]);
    edits[i] = new JTextField();
    painel.add(rotulos[i]);
    painel.add(edits[i]);
}
```



Código exemplo: FormCamposEdicao.java

12/08/2002

21

Múltipla Marcação: CheckBox

```
private JCheckBox checkBox [];

...
checkBox = new JCheckBox[5];

...
for (int i = 0; i < qtdeCheck; i++)
{
    checkBox[i] = new JCheckBox("CheckBox "+ (i+1));
    areaForm.add(checkBox[i]);
}

...
checkBox[1].setSelected(true);
```



Código exemplo: FormCheckBox.java

12/08/2002

22

Única Marcação: RadioButton

```
private JRadioButton solteiro, casado, divorciado;
private ButtonGroup grupoRadio;
private String rotulos[ ] = {"Solteiro", "Casado", "Divorciado"};

...
solteiro = new JRadioButton(rotulos[0],true);
casado = new JRadioButton(rotulos[1],false);
divorciado = new JRadioButton(rotulos[2],false);

...
grupoRadio = new ButtonGroup ( );
grupoRadio.add(solteiro);
grupoRadio.add(casado);
grupoRadio.add(divorciado);
```



Código exemplo: FormRadioButton.java

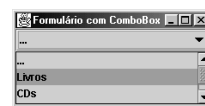
12/08/2002

23

Caixa de Seleção: ComboBox

```
private JComboBox comboBox;
private String itens [ ] = { "...", "Livros", "CDs", "DVDs" };

...
comboBox = new JComboBox ( itens );
comboBox.setMaximumRowCount ( 3 );
comboBox.setSize ( 100, 30 );
```



Código exemplo: FormComboBox.java

12/08/2002

24

✧ Lista de Seleção: ListBox

```
private JList lista;
private String itens [ ] = { "Apto", "Casa", "Sítio", "Sobrado" };
...
lista = new JList ( itens );
lista.setVisibleRowCount ( 3 );
lista.setSelectionMode ( ListSelectionMode.SINGLE_SELECTION );
...
areaForm.add ( new JScrollPane ( lista ) );
```



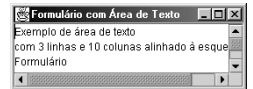
Código exemplo: FormListBox.java

12/08/2002

25

✧ Área de Texto

```
private JTextArea areaTexto;
private String texto;
...
texto = "Exemplo de área de texto\n" +
        "com 3 linhas e 10 colunas alinhado à esquerda do\n" +
        "Formulário\n\n\n\n\n\n\n\n\n\n";
...
areaTexto = new JTextArea(texto, 3, 10);
areaForm.add(new JScrollPane (areaTexto), BorderLayout.CENTER);
```



Código exemplo: FormAreaTexto.java

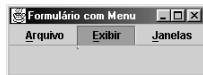
12/08/2002

26

✧ Barra de Menu

```
public class FormBarraMenu extends FormBase
...
protected JMenuBar barraMenu;
protected JMenu menuArquivo, menuExibir, menuJanelas;
...
barraMenu = new JMenuBar ( );
setJMenuBar( barraMenu );

menuArquivo = new JMenu( "Arquivo" );
menuArquivo.setMnemonic( 'A' );
barraMenu.add( menuArquivo );
```



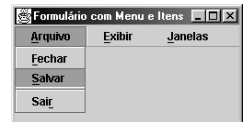
Código exemplo: FormBarraMenu.java

12/08/2002

27

✧ Menu com Itens

```
public class FormMenuItens extends FormBarraMenu
...
private JMenuItem itemFechar, itemSalvar, itemSair;
...
itemFechar = new JMenuItem("Fechar");
itemFechar.setMnemonic('F');
menuArquivo.add(itemFechar);
...
menuArquivo.addSeparator( );
```



Código exemplo: FormBarraMenu.java

12/08/2002

28

✧ Menu com Itens CheckBox

```
public class FormMenuCheck extends FormMenuItens
...
private JCheckBoxMenuItem itemCheckRegua, itemCheckGuias;
...
itemCheckRegua = new JCheckBoxMenuItem("Régua",false);
itemCheckRegua.setMnemonic('R');
menuExibir.add(itemCheckRegua);
```



Código exemplo: FormMenuCheck.java

12/08/2002

29

✧ Menu com Itens RadioButton

```
public class FormMenuRadio extends FormMenuCheck
...
protected JRadioButtonMenuItem itemHorizontal, itemVertical,
        itemCascata;
protected ButtonGroup grupoRadio;
...
grupoRadio = new ButtonGroup ( );
...
itemHorizontal = new JRadioButtonMenuItem("Horizontal",false);
itemHorizontal.setMnemonic('H');
menuJanelas.add(itemHorizontal);
grupoRadio.add(itemHorizontal);
```



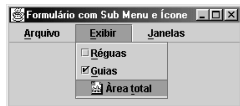
Código exemplo: FormMenuRadio.java

12/08/2002

30

Menu com Itens com Ícone

```
public class FormSubMenu extends FormMenuRadio
...
protected JMenu itemGrade;
protected JMenuItem itemIcone;
protected JCheckBoxMenuItem gVertical, gHorizontal;
...
itemIcone = new JMenuItem ("Área total", new ImageIcon ("doc.gif"));
itemIcone.setMnemonic('t');
menuExibir.add(itemIcone);
```



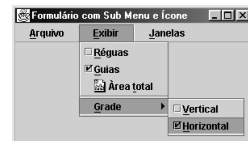
Código exemplo: FormSubMenu.java

12/08/2002

31

Menu com Sub Itens

```
menuExibir.addSeparator();
...
itemGrade = new JMenu ("Grade");
itemGrade.setMnemonic('G');
menuExibir.add(itemGrade);
...
gVertical = new JCheckBoxMenuItem ("Vertical",false);
gVertical.setMnemonic('V');
itemGrade.add(gVertical);
```



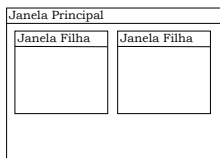
Código exemplo: FormSubMenu.java

12/08/2002

32

Formulários Internos - MDI

- ▶ Muitos aplicativos precisam ser divididos em diversas janelas;
- ▶ Interface de múltiplos documentos (multiple document interface - MDI) permitem organizar um aplicativo em uma janela principal contendo diversas janelas filhas.



12/08/2002

33

Tratamento de Eventos

12/08/2002

34

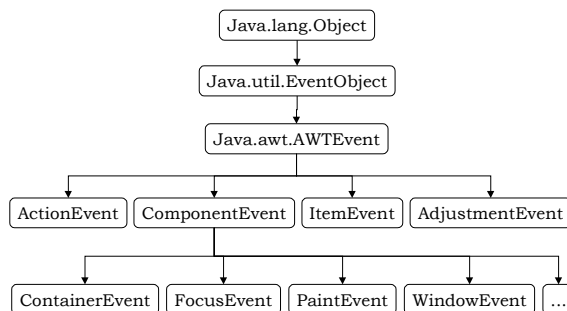
Tratamento de Eventos

- ▶ As GUIs são baseados em eventos;
- ▶ Algumas interações comuns como mover o mouse, clicar num botão, digitar num campo de texto, selecionar um item de menu, fechar uma janela etc;
- ▶ Informações sobre eventos são armazenados em um objeto de uma classe que estende `awt.event`;
- ▶ Novos tipos de eventos de componente do swing são definidos no pacote `javax.swing.event`.

12/08/2002

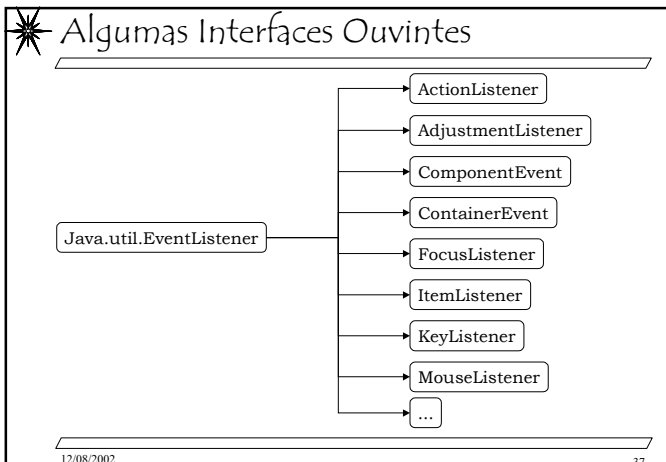
35

Classes de evento do pacote awt



12/08/2002

36



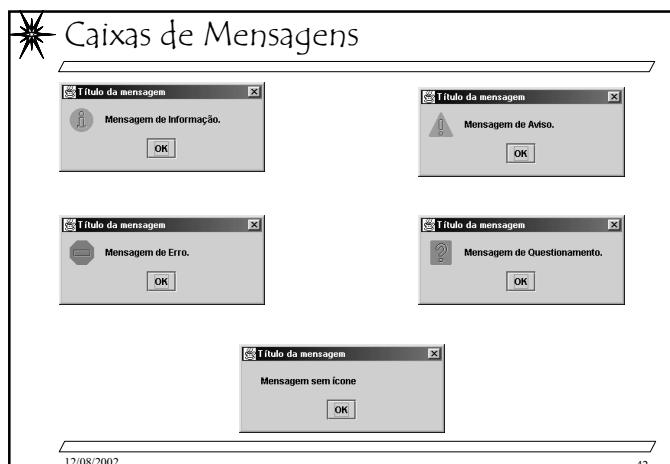
- ## Tratamento de Eventos
- ▶ Para processar um evento de interface gráfica, o programador deve realizar duas tarefas:
 - ▶ registrar um ouvinte de eventos e,
 - ▶ implementar um “tratador” (handler) de eventos.
 - ▶ Um objeto ouvinte de eventos “ouve” tipos específicos de eventos gerados no mesmo objeto ou por outros objetos num programa;
 - ▶ Um tratador de eventos é um método que é automaticamente chamado em resposta a um tipo de evento em particular.
- 12/08/2002 38

- ## Tratamento de Eventos
- ▶ Devido às classes de captação/reconhecimento de eventos fazerem parte do grupo de classes `java.awt.event`, você deve importar o pacote:
 - ▶ `import java.awt.event.*;`
 - ▶ A classe deve empregar a instrução *implements* para declarar que estará usando uma ou mais interfaces de captação/reconhecimento.
- 12/08/2002 39

- ## EventListener
- ▶ As interfaces `EventListener` possibilitam que um componente de uma interface gerem eventos;
 - ▶ Sem um dos dispositivos de captação, um componente não pode fazer com que seja percebido por outras partes do programa;
 - ▶ Um programa deve dispor de uma interface de captação para cada tipo de componente;
 - ▶ Por exemplo: a resposta a um clique num botão ou à tecla “Enter” pressionada deve incluir a interface `ActionListener`.
- 12/08/2002 40

Eventos de Componentes

12/08/2002 41





Caixas de Mensagens

```
import javax.swing.JOptionPane;
```

```
JOptionPane.showMessageDialog( null, "Mensagem de Informação.",  
    "Título da mensagem", JOptionPane.INFORMATION_MESSAGE);
```

- ▶ JOptionPane.ERROR_MESSAGE
- ▶ JOptionPane.WARNING_MESSAGE
- ▶ QUESTION_MESSAGE
- ▶ JOptionPane.PLAIN_MESSAGE



Código exemplo: FormCaixaDialogo.java

12/08/2002

43



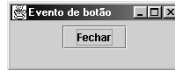
Eventos para Botão

```
botao.addActionListener ( new ActionListener () {  
    public void actionPerformed ( ActionEvent e ) {  
        System.exit(0);  
    }  
});
```

Classe interna

Método da classe interna

- ▶ Registra um ouvinte de ação (ActionListener) para ouvir os eventos do botão, p.e. clique no botão;
- ▶ new ActionListener () define uma classe interna anônima;
- ▶ A classe ActionListener herda uma implementação padrão (método) actionPerformed (ActionEvent e) que é invocado quando um evento de ação ocorre.



Código exemplo: EventoBotao.java

12/08/2002

44