

Interface Gráfica



Prof. Bruno Gomes
bruno.gomes@ifrn.edu.br

Programação Orientada a Objetos

JFC

- *Java Foundation Classes*
 - Coleção de pacotes para criação de aplicações completas Desktop
 - Interfaces gráficas (GUIs - *Graphical User Interface*)
 - Funcionalidade e interatividade
- Características:
 - Componentes GUI Swing
 - Suporte a Look and Feel
 - API para acessibilidade
 - API Java 2D
 - Suporte a Drag and Drop
 - Internacionalização

História

- Quando o Java 1.0 foi introduzido, continha a biblioteca *Abstract Window Toolkit (ATW)*
 - Lida com elementos da interface com o usuário delegando a criação e o comportamento desses elementos ao conjuntos de ferramentas GUI nativo em cada plataforma alvo
- Problema:
 - Difícil escrever uma biblioteca gráfica portátil de alta qualidade

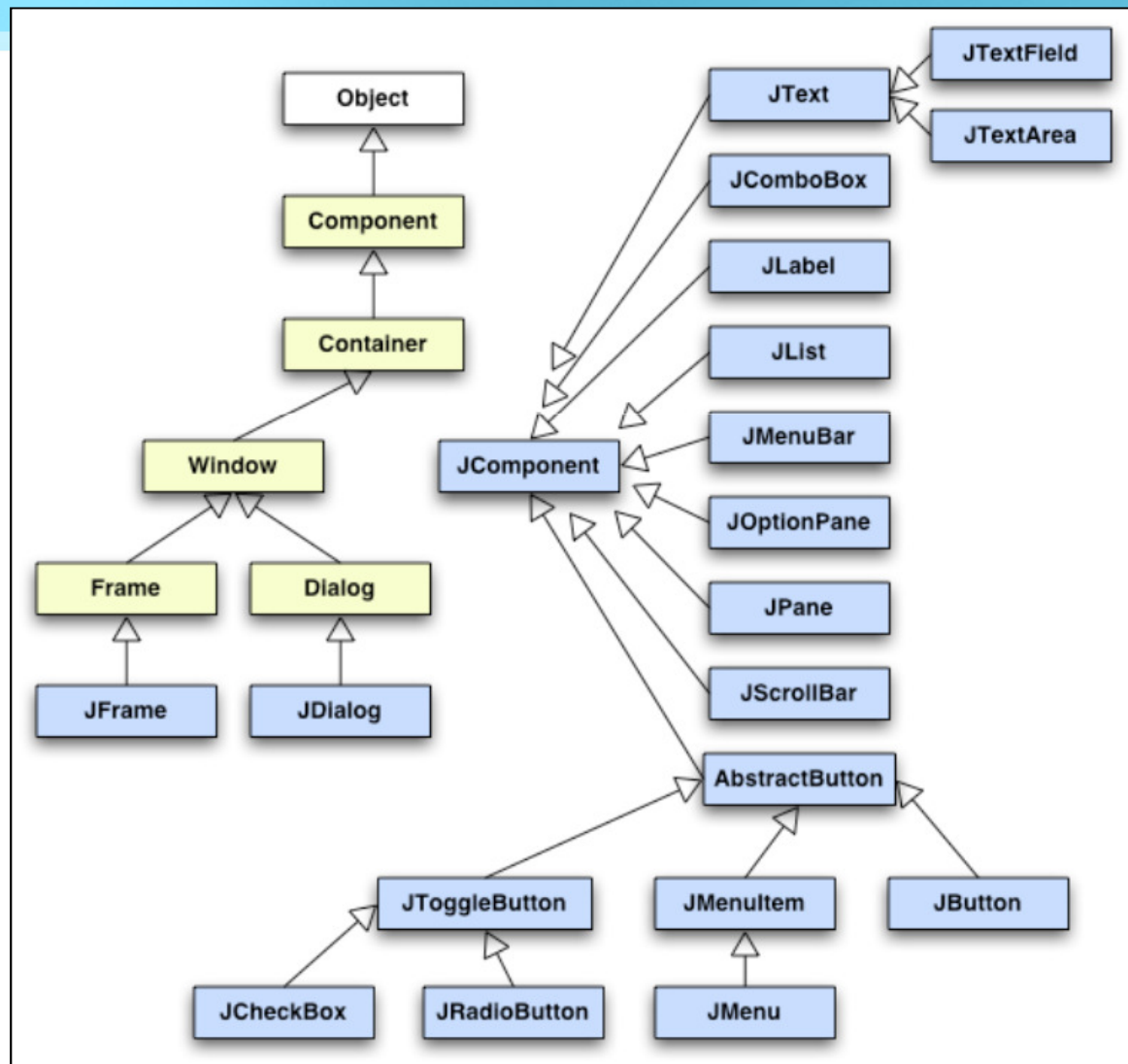
História

- Em 1996 a Netscape criou uma biblioteca de GUI chamada IFC (*Internet Foundation Classes*)
 - Elementos da interface com o usuário eram pintados em janelas em branco
 - Comportamento idênticos
- A Sun trabalhou com a Netscape para aperfeiçoar essa abordagem, criando uma biblioteca chamada “Swing”

Swing

- `javax.swing.*`
- Biblioteca oficial da Java GUI
- Todo programa que usa Swing deve ter pelo menos um componente do tipo *container*.
- Principais Características:
 - Independente de plataforma
 - Personalizável
 - Extensível
 - Configurável
 - Leve

Swing e AWT



Container

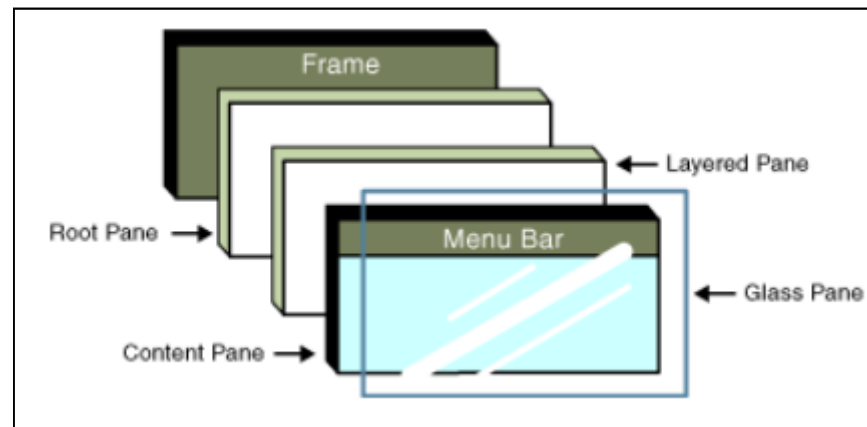
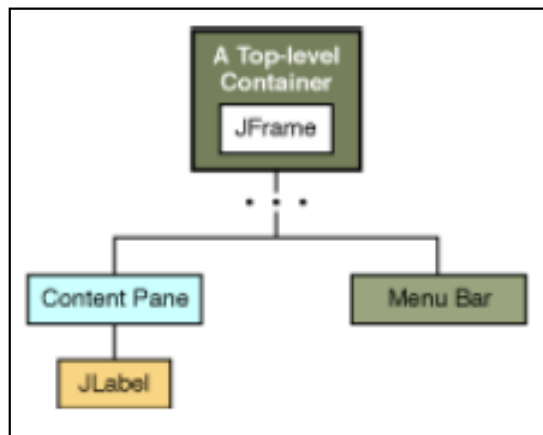
- Descendentes da classe *java.awt.Container*
- Componentes que podem conter outros componentes
- Oferece suporte para desenho e tratamento de eventos

Containers “top level”

- Provê o suporte que os componentes swing necessitam para realizar o desenho da tela e o tratamento de eventos.
 - **JFrame**
 - Objeto que implementa janela principal
 - **JDialog**
 - Objeto que implementa janela secundária
 - **JApplet**
 - Objeto que implementa uma área de visualização de applets em browsers;

JFrame

- Um frame é uma janela que, por default, tem decorações tais como:
 - Uma borda
 - Um título
 - Botões para minimizar e fechar a janela
- Aplicações com uma GUI usam, tipicamente, pelo menos um frame.



JFrame

- Exemplo de uso:

```
public class HelloFrame {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("HelloWorldSwing");  
        ...  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```

pack() faz com que a janela seja ajustada para o tamanho de todos os seus sub-componentes

JFrame

- Exemplo de uso 2:

```
import javax.swing.*;

public class HelloFrame extends JFrame {

    public HelloFrame() {
        super("HelloSwing");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
        setVisible(true);
    }

    public static void main(String[] args) {
        HelloFrame frame = new HelloFrame();
    }
}
```

JContentPane

- Todo container “top-level” contém um container intermediário denominado **Content Pane**
- Contém todos os componentes visíveis

```
frame.getContentPane().add(label);
```

Também funciona:

```
frame.add(label);
```

JComponent

- Com exceção dos containers “top level”, todos os componentes Swing descendem da classe JComponent.
- Todo container “toplevel” tem um “content pane” que contém, direta ou indiretamente, todos os componentes visíveis no container (exceto menus e decorações de janelas).
 - Por exemplo: Um botão deve ser adicionado ao “content pane” do frame ao invés de diretamente ao frame.

Alguns Componentes:

Componente	Descrição
JLabel	Área em que podem ser exibidos texto não-editável ou ícones
TextField	Área que se insere dados pelo teclado. Pode também exibir informações
Button	Área que aciona um evento quando você clica
CheckBox	Componente GUI que tem dois estados: selecionado ou não-selecionado
ComboBox	Lista de itens a partir do qual o usuário pode fazer uma seleção clicando em um item na lista ou digitando na caixa, se permitido
List	Área em que uma lista de itens é exibida. O usuário pode fazer uma seleção clicando uma vez em qualquer elemento na lista.
Panel	Container em que os componentes podem ser colocados

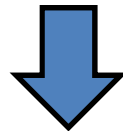
Componente - JLabel

- Também conhecidos como rótulos
- Fornecem instruções de texto ou informações em uma GUI
- Classe JLabel
 - Subclasse de JComponent
- Exibe:
 - Uma única linha de texto **somente de leitura**
 - Uma Imagem
 - Ou texto e imagem

Componente - JLabel

- Somente Texto:

```
JFrame frame = new JFrame("Titulo");  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
JLabel label = new JLabel("Somente Texto");  
label.setToolTipText("Texto que aparece com o mouse sobre o label");  
  
frame.add(label);  
  
frame.pack();  
frame.setVisible(true);
```



Componente - JLabel

- Imagem e Texto:

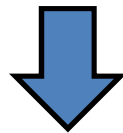
```
Icon imagem = new ImageIcon("ifrn.png");  
  
JLabel label = new JLabel("Imagem e texto",  
    imagem, SwingConstants.LEFT);  
label.setToolTipText("Texto que aparece com o mouse sobre o label");
```



Componente - JLabel

- Sem argumentos no construtor:

```
Icon imagem = new ImageIcon("ifrn.png");  
  
JLabel label = new JLabel();  
label.setText("Imagem e texto");  
label.setIcon(imagem);  
label.setHorizontalTextPosition(SwingConstants.CENTER);  
label.setVerticalTextPosition(SwingConstants.BOTTOM);  
label.setToolTipText("Texto que aparece com o mouse sobre o label");
```



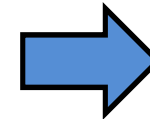
Componente – JTextField e JPasswordField

- São áreas de uma única linha em que o usuário insere texto pelo teclado, ou simplesmente serve para exibir texto
- JPasswordField oculta os caracteres assumindo que eles representam uma senha

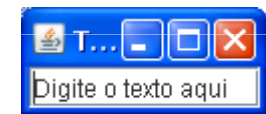
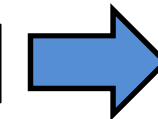
Componente – JTextField e JPasswordField

- Exemplos:

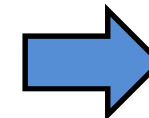
```
JTextField textField = new JTextField(10);
```



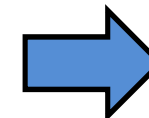
```
JTextField textField = new JTextField("Digite o texto aqui");
```



```
JTextField textField = new JTextField("textField não-editável");  
textField.setEditable(false);
```



```
JPasswordField textField = new JPasswordField("Texto Oculto");
```



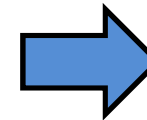
Componente - JButton

- Botão em que o usuário clica para disparar uma ação específica
 - Botões de comando
 - Caixas de marcação
 - Botões de alternância
 - Botões de opção

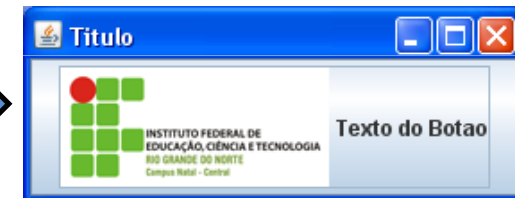
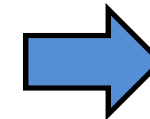
Componente - JButton

- Exemplos:

```
JButton botao = new JButton("Texto do Botao");
```



```
Icon imagem = new ImageIcon("ifrn.png");  
JButton botao = new JButton("Texto do Botao", imagem);
```



Próximos Componentes

- JCheckBox e JRadioButton
- JComboBox
- JList
- JPanel

Antes: Precisamos aprender
como Gerenciar o Layout

Gerenciamento de Layouts

- Fornecidos para organizar componentes GUI em um container
- É o processo de determinar o tamanho e a posição dos componentes na janela gráfica do programa
- Java tem 6 tipos de layouts:
 - **BorderLayout**
 - BorderLayout
 - **FlowLayout**
 - GridBagLayout
 - **GridLayout**
 - CardLayout

Gerenciamento de Layouts

- Quando chamamos o método "add" para adicionar um componente na tela, ele será posicionado de acordo com o gerenciador de layout previamente configurado
- A escolha do gerenciador de layout depende muito das necessidades do programa
- Por default:
 - **JFrame** e **JDialog** usam o **BorderLayout**
 - **JPanel** usa o **FlowLayout**

Gerenciamento de Layouts

- Utiliza o método “setLayout” para mudar o Layout do container:

```
getContentPane().setLayout(new GridLayout(3, 2));
```

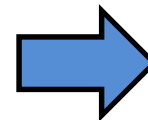
Layout - BorderLayout

- Padrão para os painéis de conteúdo de JFrames
- Organiza os componentes em cinco áreas: norte, sul, leste, oeste e centro
- Até cinco elementos podem ser adicionados
- Os componentes colocados em cada região podem ser containers ao qual podem receber outros elementos

Layout - BorderLayout

- Exemplo:

```
JButton botao1 = new JButton("Botão 1");  
JButton botao2 = new JButton("Botão 2");  
JButton botao3 = new JButton("Botão 3");  
JButton botao4 = new JButton("Botão 4");  
JButton botao5 = new JButton("Botão 5");  
frame.add(botao1, BorderLayout.NORTH);  
frame.add(botao2, BorderLayout.CENTER);  
frame.add(botao3, BorderLayout.WEST);  
frame.add(botao4, BorderLayout.SOUTH);  
frame.add(botao5, BorderLayout.EAST);
```

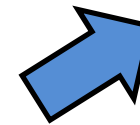


Layout - FlowLayout

- Coloca os componentes sequencialmente (esquerda para direita) na ordem em que foram adicionados
- Gerenciador mais básico
- Quando se alcança a borda do container, continuam na próxima linha
- Permite que elementos:
 - Sejam alinhados à esquerdas
 - Centralizados
 - Sejam alinhados à direita

Layout - FlowLayout

- Exemplo:



```
JButton botao1 = new JButton("Botão 1");
JButton botao2 = new JButton("Botão 2");
JButton botao3 = new JButton("Botão 3");
JButton botao4 = new JButton("Botão 4");
JButton botao5 = new JButton("Botão 5");
frame.add(botao1);
frame.add(botao2);
frame.add(botao3);
frame.add(botao4);
frame.add(botao5);

FlowLayout flowLayout = new FlowLayout();
flowLayout.setAlignment(FlowLayout.CENTER);
frame.getContentPane().setLayout(flowLayout);
```

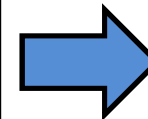
Layout - FlowLayout

```
JLabel label1 = new JLabel("Usuario:");  
JTextField textField1 = new JTextField(10);  
JLabel label2 = new JLabel("Senha:");  
JPasswordField passwordField1 = new JPasswordField(11);  
JButton botao1 = new JButton("OK");  
JButton botao2 = new JButton("Sair");
```

```
frame.add(label1);  
frame.add(textField1);  
frame.add(label2);  
frame.add(passwordField1);  
frame.add(botao1);  
frame.add(botao2);
```

```
FlowLayout flowLayout = new FlowLayout();  
flowLayout.setAlignment(FlowLayout.LEFT);  
frame.getContentPane().setLayout(flowLayout);
```

```
frame.setSize(200, 125);
```



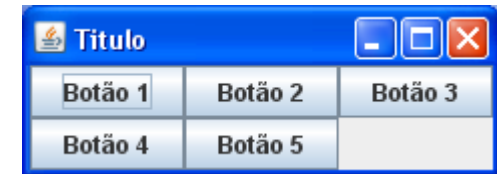
Utilizado no lugar do
pack() para definir
manualmente o
tamanho do container

Layout - GridLayout

- Organiza os elementos em linhas e colunas
- Os elementos são adicionados nas células da esquerda para a direita até a linha estar cheia

Layout - GridLayout

- Exemplo:



```
 JButton botao1 = new JButton("Botão 1");  
 JButton botao2 = new JButton("Botão 2");  
 JButton botao3 = new JButton("Botão 3");  
 JButton botao4 = new JButton("Botão 4");  
 JButton botao5 = new JButton("Botão 5");  
  
 frame.add(botao1);  
 frame.add(botao2);  
 frame.add(botao3);  
 frame.add(botao4);  
 frame.add(botao5);  
  
 frame.getContentPane().setLayout(new GridLayout(2, 3));
```

Agora vamos voltar!

- JCheckBox e JRadioButton
- JComboBox
- JList

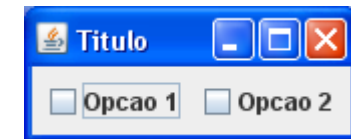
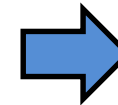
Componentes JCheckBox e JRadioButton

- Valores ativados/desativados e verdadeiro/falso
- Subclasses de JToggleButton
- JRadioButton
 - Vários agrupados e somente um pode estar selecionado (verdadeiro/falso)
- JCheckBox
 - Vários, e vários podem estar selecionados (ativados/desativados)

Componentes JCheckBox e JRadioButton

- Exemplo JCheckBox :

```
JCheckBox checkBox1 = new JCheckBox("Opcao 1");  
JCheckBox checkBox2 = new JCheckBox("Opcao 2");  
  
frame.add(checkBox1);  
frame.add(checkBox2);  
  
FlowLayout flowLayout = new FlowLayout();  
flowLayout.setAlignment(FlowLayout.CENTER);  
frame.getContentPane().setLayout(flowLayout);  
  
frame.pack();
```



Componentes JCheckBox e JRadioButton

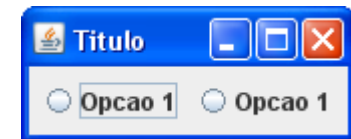
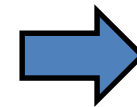
- Exemplo JRadioButton:

```
JRadioButton radioButton1 = new JRadioButton("Opcao 1");
JRadioButton radioButton2 = new JRadioButton("Opcao 2");
ButtonGroup radioButtonGroup = new ButtonGroup();

frame.add(radioButton1);
frame.add(radioButton2);

radioButtonGroup.add(radioButton1);
radioButtonGroup.add(radioButton2);

FlowLayout flowLayout = new FlowLayout();
flowLayout.setAlignment(FlowLayout.CENTER);
frame.getContentPane().setLayout(flowLayout);
```



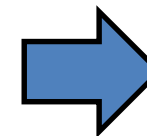
Componente JComboBox

- Também conhecido como Caixa de Combinação
- Fornece uma lista de itens da qual o usuário pode fazer uma seleção

Componente JComboBox

- Exemplo:

```
String nomes[] = {"Nome 1", "Nome 2", "Nome 3"};  
JComboBox comboBox = new JComboBox(nomes);  
  
frame.add(comboBox);  
  
FlowLayout flowLayout = new FlowLayout();  
flowLayout.setAlignment(FlowLayout.CENTER);  
frame.getContentPane().setLayout(flowLayout);  
  
frame.setSize(150, 150);
```



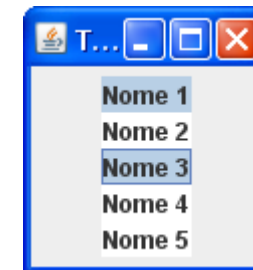
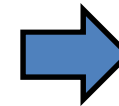
Componente JList

- Exibe uma série de itens da qual o usuário pode selecionar um ou mais itens

Componente JList

- Exemplo 1:

```
String nomes[] = {"Nome 1", "Nome 2", "Nome 3",  
"Nome 4", "Nome 5"};  
JList list = new JList(nomes);  
  
frame.add(list);  
  
FlowLayout flowLayout = new FlowLayout();  
flowLayout.setAlignment(FlowLayout.CENTER);  
frame.getContentPane().setLayout(flowLayout);  
  
frame.pack();
```



JPanel

- GUIs mais complexas exigem que cada componente seja colocado em uma posição exata
- Consistem muitas vezes em painéis com os componentes de cada painel organizados

JPanel

- Exemplo:

```
JPanel panel = new JPanel();
```

```
JButton botao1 = new JButton("Botão 1");
```

```
JButton botao2 = new JButton("Botão 2");
```

```
JButton botao3 = new JButton("Botão 3");
```

```
panel.add(botao1);
```

```
panel.add(botao2);
```

```
panel.add(botao3);
```

```
String nomes[] = {"Nome 1", "Nome 2", "Nome 3",  
"Nome 4", "Nome 5"};
```

```
JList list = new JList(nomes);
```

```
frame.add(list, BorderLayout.NORTH);
```

```
frame.add(panel, BorderLayout.SOUTH);
```

