

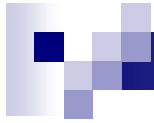


# Curso de Java

by Antonio Rodrigues Carvalho Neto



# Interface Gráfica



# Java Graphical User Interface

- Para criar janelas gráficas no Java é preciso utilizar uma das bibliotecas gráficas disponíveis (AWT, Swing ou SWT).
  - AWT foi uma primeira biblioteca gráfica criada para Java
  - Swing é uma biblioteca gráfica mais utilizada para Java, pois tem todas as características da AWT e é mais atualizada.
  - SWT foi criada inicialmente pela IBM e hoje é mantida pela Eclipse foundation. O SWT é mais rápido do Swing.
- Neste curso utilizaremos uma biblioteca gráfica Swing.

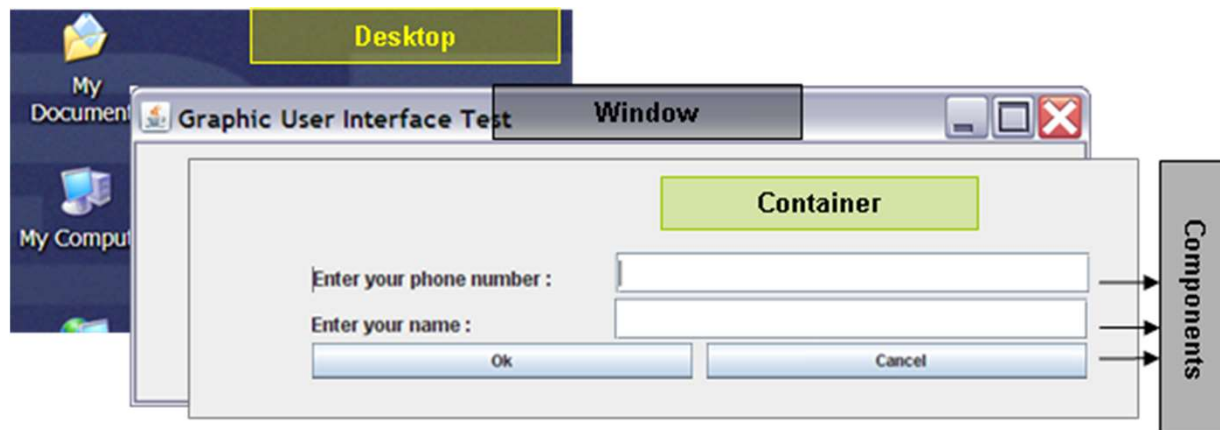


# Java Graphical User Interface

- Antes de começarmos será preciso entender alguns conceitos da biblioteca gráfica Swing.
- Assim como as outras aplicações gráficas, os objetos são colocados em janelas e não diretamente sobre o Desktop.
- E mesmo dentro das janelas eles são organizados em Containers ou Paineis.

# Java Graphical User Interface

- **Objeto gráfico** ou **Componente gráfico** são os elementos que permitem ao usuário interagir com uma aplicação. Assim como botões, caixas de textos, rótulos, etc..
- **Container** é uma área que permite uma “colagem” dos objetos gráficos.





# Elementos Gráficos

- Os objetos gráficos no Java são criados como objetos normais a partir das classes definidas na biblioteca gráfica (neste caso Swing).
- Para criar um elemento gráfico basta instanciar um objeto do tipo escolhido.
- **Exemplo** : Para criar uma caixa de texto é preciso instanciar um objeto da classe **JTextField** que é uma classe da biblioteca Swing que representa uma caixa de texto.
  - `JTextField caixa = new JTextField();`
- **Nota** : Alguns tipos objetos gráficos podem receber parametros quando estiverem sendo instanciados.

# Elementos Gráficos



- Para criar uma janela simples contendo uma frase “Hello World” é preciso seguir algumas etapas:
  - Criar uma janela gráfica
    - Instancie um objeto do tipo **JFrame** para criar uma nova janela.
    - **Sintaxe:**        **JFrame** <objeto> = **new JFrame**(<titulo>);
    - **Exemplo:**
      - **JFrame** janela = **new JFrame**(“My First Graphic Window”);
    - **Nota :** Cada instancia de **JFrame** corresponde uma uma nova janela.



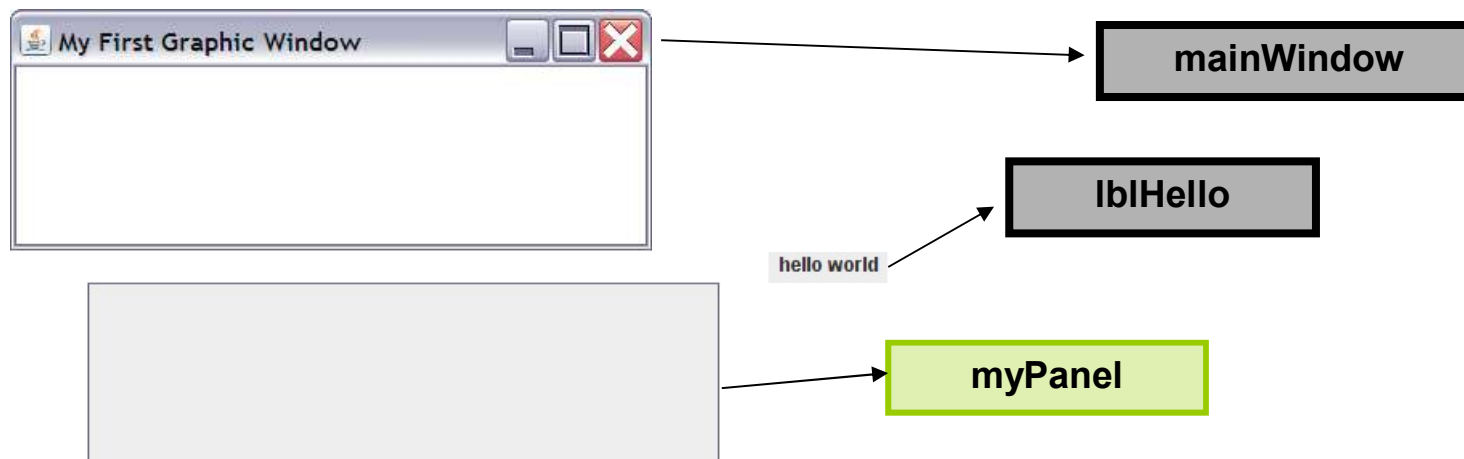
# Elementos Gráficos

- Crie um painel que servirá como container para os objetos gráficos.
  - Instancie um objeto do tipo **JPanel** para criar um novo painel.
  - **Sintaxe:**       **JPanel** <objeto> = **new JPanel()**;
  - **Exemplo:**     **JPanel** painel = **new JPanel()**;
  - **Nota** : Cada instancia de **JPanel** corresponde uma um novo painel.
- Criar o objeto gráfico desejado
  - Neste caso instancie um objeto do tipo **JLabel** para criar um novo rótulo.
  - **Sintaxe:**       **JLabel** <objeto> = **new JLabel(<rótulo>)**;
  - **Exemplo:**     **JLabel** lblHello = **new JLabel("Hello World")**;
  - **Nota** : Cada instancia de **JLabel** corresponde uma um novo rótulo.



# Conectando os elementos

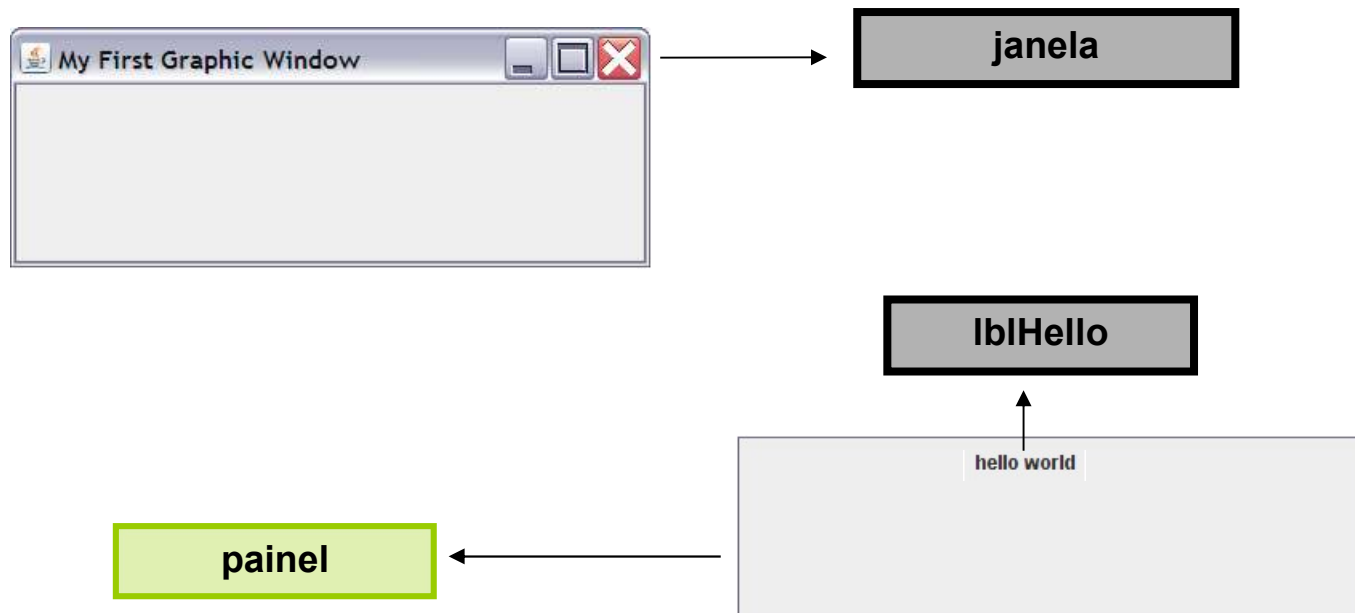
- A janela o container e os demais objetos gráficos criados ficam dispersos na memória.



- É preciso conectá-los uns aos outros para que apareçam dentro da janela.

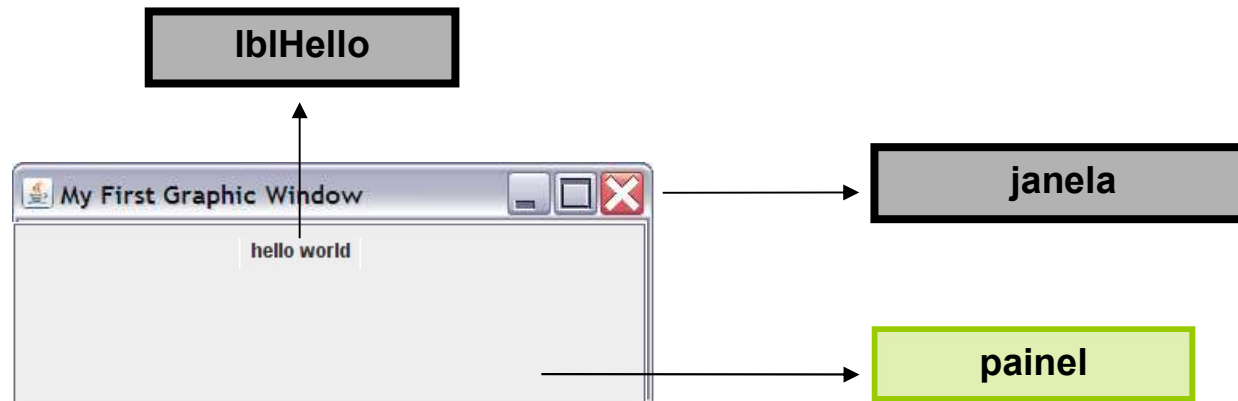
# Conectando os elementos

- Primeiro conecte os objetos gráficos no container.
  - **Sintaxe:** `<objeto container>.add(<objeto gráfico>);`
  - **Exemplo:** `painel.add(lblHello);`



# Conectando os elementos

- Conecte agora o container na janela gráfica
  - **Sintaxe:** `<objeto JFrame>.setContentPane(<objeto container>);`
  - **Exemplo:** `janela.setContentPane(painel);`
  - **Nota :** uma janela gráfica pode conter apenas um objeto do tipo container.



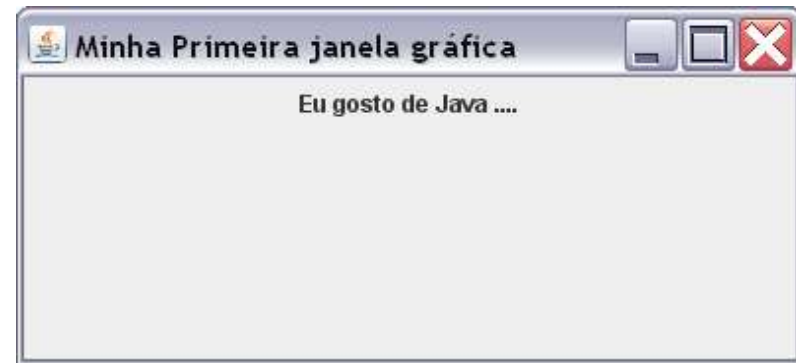


# Mostrando os elementos

- Agora que os elementos estão conectados resta apenas determinar o tamanho da janela e torna-la visível.
- Para isto utilize as seguintes funções:
  - **Determinar o tamanho**
    - **Sintaxe:**       <objeto janela>.setSize(<largura>, <altura>);
    - **Exemplo:**       janela.setSize(400, 200);
    - **Nota:** O tamanho da janela é medido em **pixels**
  - **Tornar uma janela visível**
    - **Sintaxe:**       <objeto janela>.setVisible(<visivel>);
    - **Exemplo:**       janela.setVisible(true);
    - **Nota:** Se deixar como **true** uma janela ficará visível e se deixar como **false** ela se tornará invisível.

# Exercício

- Desenvolva uma aplicação conforme o layout ao lado, usando os comandos abaixo:



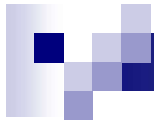
```
JFrame <objeto janela> = new JFrame(<titulo>);
JPanel <objeto painel> = new JPanel();
JLabel <objeto label> = new JLabel(<texto>);
JTextField <objeto textfield> = new JTextField();
JButton <objeto button> = new JButton(<texto>);
<objeto janela>.setContentPane( <objeto painel> );
<objeto painel>.add( <componente gráfico> );
<objeto janela>.setSize( <largura>, <largura>);
<objeto janela>.setVisible( <boolean true ou false> );
<objeto janela>.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
```



# Código Completo

- O código completo para este exemplo ficaria assim:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        // Cria os elementos gráficos (janela, painel e objeto)  
        JFrame janela = new JFrame("My first Graphic Application");  
        JPanel painel = new JPanel();  
        JLabel lblHello = new JLabel("Hello World");  
        // Conecta os elementos  
        painel.add(lblHello);  
        janela.setContentPane(painel);  
        // Mostra uma janela gráfica  
        janela.setSize(400, 200);  
        janela.setVisible(true);  
        janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```



# Agregando uma Janela

- Para manter uma melhor coesão a Janela pode ser feita em uma classe separada, tendo seus componentes como sendo variáveis de instância privadas. A instanciação e conexão dos componentes gráficos será feita no construtor da nova classe.



## Agregando uma Janela (cont.)

- Crie uma classe, contendo uma agregação para a classe JFrame:

```
public class MinhaJanela {  
    private JFrame janela;  
}
```

- Crie o painel e os elementos gráficos como sendo variáveis de instância dentro da janela.

```
public class MinhaJanela {  
    private JFrame janela;  
    private JPanel painel;  
    private JLabel lblHello;  
}
```





# Agregando uma Janela (cont.)

- Modifique o construtor para instanciar e conectar os elementos gráficos uns aos outros.

```
public class MinhaJanela {  
    private JFrame janela;  
    private JPanel painel;  
    private JLabel lblHello;  
  
    public MinhaJanela() {  
        // Instância os componentes gráficos  
        janela = new JFrame();  
        painel = new JPanel();  
        lblHello = new JLabel("Eu gosto de Java...");  
        // Conecta, uns aos outros  
        painel.add(lblHello);  
        janela.setContentPane(painel);  
        // Mostra a janela na tela  
        janela.setSize(300, 300);  
        janela.setVisible(true);  
        janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```



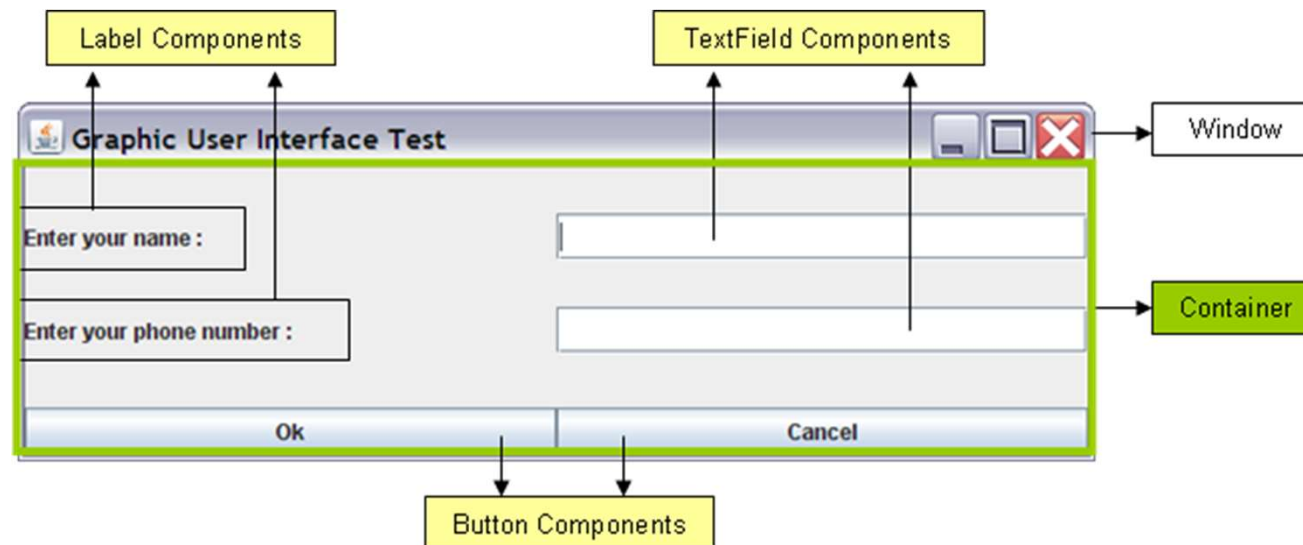
## Agregando uma Janela (cont.)

- Para mostrar a janela na tela basta criar um objeto do tipo **MinhaJanela**



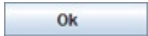

```
public class Teste {  
    public static void main(String[ ] args) {  
        MinhaJanela jan = new MinhaJanela();  
    }  
}
```

# Outros objetos gráficos

- Existem outros objetos gráficos que podem ser utilizados para uma criação da interface gráfica.

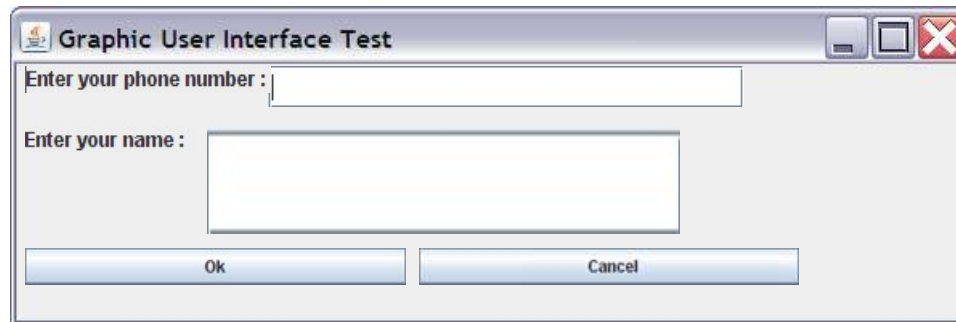


# Outros objetos gráficos

Java Swing Objeto Gráfico	Descrição	Métodos Importantes
<b>JTextField</b> 	<p>Cria uma caixa de texto que pode ser utilizada pelo usuário para digitar informações.</p> <p><b>Sintaxe :</b>  <code>JTextField &lt;objeto&gt; = new JTextField(&lt;número colunas&gt;);</code></p> <p><b>Exemplo :</b>  <code>JTextField nome = new JTextField(20);</code></p>	<p><b>String getText()</b> retorna um String com o texto digitado dentro da caixa.</p> <p><b>void setText(String t)</b> define um texto dentro da caixa.</p> <p><b>boolean isEditable()</b> retorna um <b>boolean</b> indicando se este objeto é editável</p> <p><b>void setEditable(boolean b)</b> define se o objeto será editável ou não.</p>
<b>JLabel</b> 	<p>Cria um espaço onde pode ser exibido um texto curto uma imagem ou ambos.</p> <p><b>Sintaxe :</b>  <code>JLabel &lt;objeto&gt; = new JLabel(String &lt;conteúdo&gt;);</code></p> <p><b>Exemplo :</b>  <code>JLabel name = new JLabel("hello world");</code></p>	<p><b>String getText()</b> retorna um String com o conteúdo do rótulo.</p> <p><b>void setText(String t)</b> coloca um novo texto no rótulo.</p>
<b>JButton</b> 	<p>Implementa um botão o qual mostra um texto. Sua utilização é para acionar uma determinada ação quando for pressionado.</p> <p><b>Sintaxe :</b>  <code>JButton &lt;objeto&gt; = new JButton (String &lt;conteúdo&gt;);</code></p> <p><b>Exemplo :</b>  <code>JButton name = new JButton("Ok");</code></p>	<p><b>String getText()</b> retorna um String com o conteúdo do botão.</p> <p><b>void setText(String t)</b> define um novo texto para ser o conteúdo do botão.</p> <p><b>void setActionCommand(String actionCommand)</b> define o nome da ação que botão irá fazer. Normalmente a ação tem o mesmo valor contido no nome do botão.</p> <p><b>String getActionCommand()</b> retorna o nome da ação do botão.</p>
<b>JTextArea</b> 	<p>Cria uma área de texto permitindo a visualização ou edição de um grande texto.</p> <p><b>Sintaxe :</b>  <code>JTextArea &lt;objeto&gt; = new JTextArea( &lt;linhas&gt;, &lt;colunas&gt;);</code></p> <p><b>Exemplo :</b>  <code>JTextArea name = new JTextArea(5, 20);</code></p>	<p><b>String getText()</b> retorna uma String com o texto digitado dentro da area.</p> <p><b>void setText(String t)</b> define um texto dentro da area</p> <p><b>boolean isEditable()</b> retorna um <b>boolean</b> indicando se este objeto é editável</p> <p><b>void setEditable(boolean b)</b> define se o objeto será editável ou não.</p>

# Exercício (para casa)

- Desenvolva uma aplicação conforme o layout abaixo:



The image shows a screenshot of a Java Swing window titled "Graphic User Interface Test". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there is a form with two text input fields. The first field is preceded by the label "Enter your phone number :". The second field is preceded by the label "Enter your name :". At the bottom of the window, there are two buttons: "Ok" and "Cancel".



# Referências

- Java como programar 6<sup>a</sup> edição

Capítulo 11 e Capítulo 22

- Use a cabeça Java 2<sup>a</sup> edição

Capitulo 12

pags. 251 a 253