

Interface gráfica em Java

Técnicas de Programação

Prof. Iális Cavalcante

Engenharia da Computação

2015.2

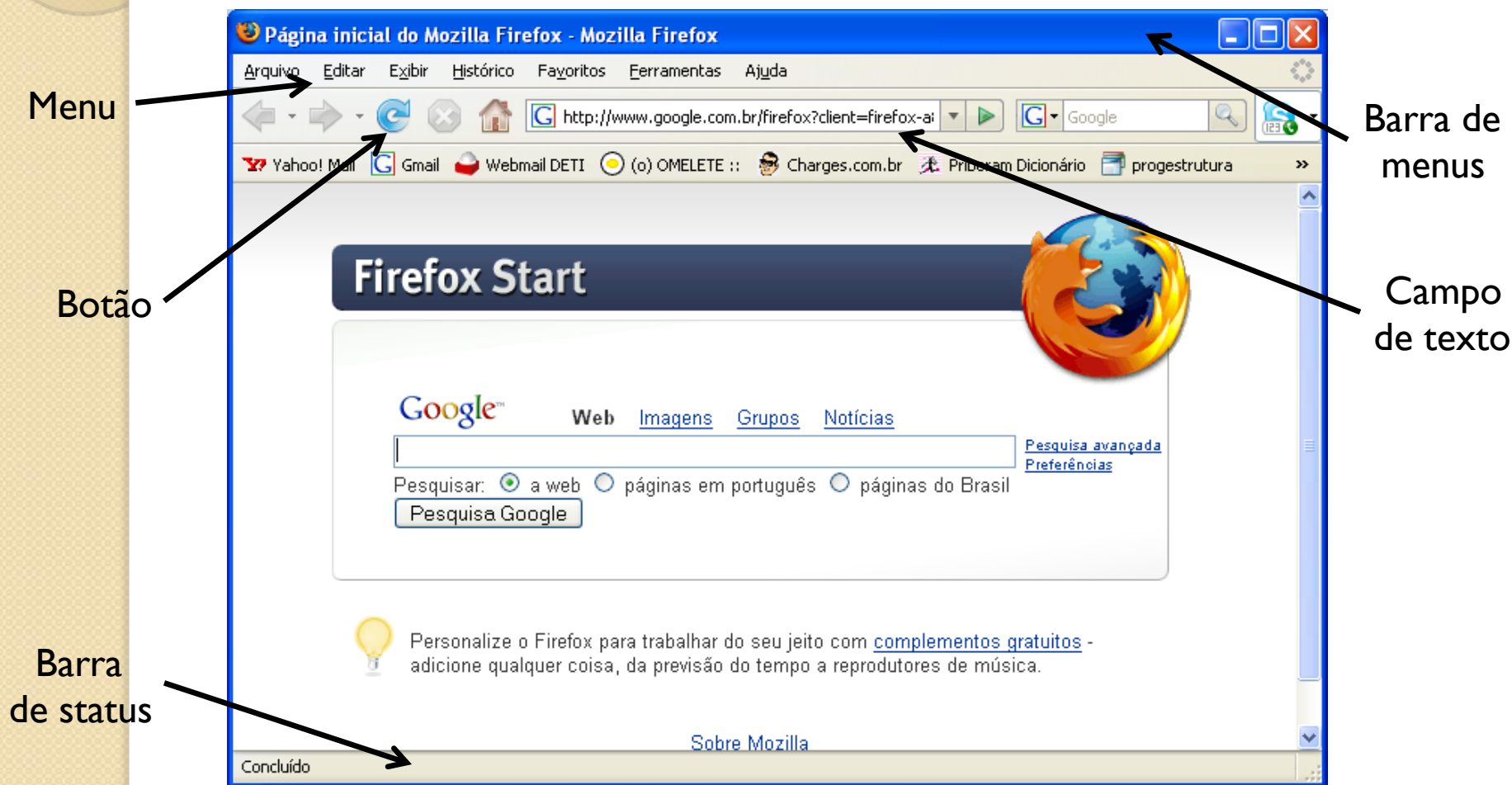


Introdução

- GUI – *graphical user interface*
 - Interface gráfica com o usuário
- Fornece ao programa “aparência” e comportamentos “diferenciados”:
 - Nível básico de familiaridade com o programa;
 - Reduz tempo de aprendizado;
 - Aumenta habilidade de uso produtivo do programa.

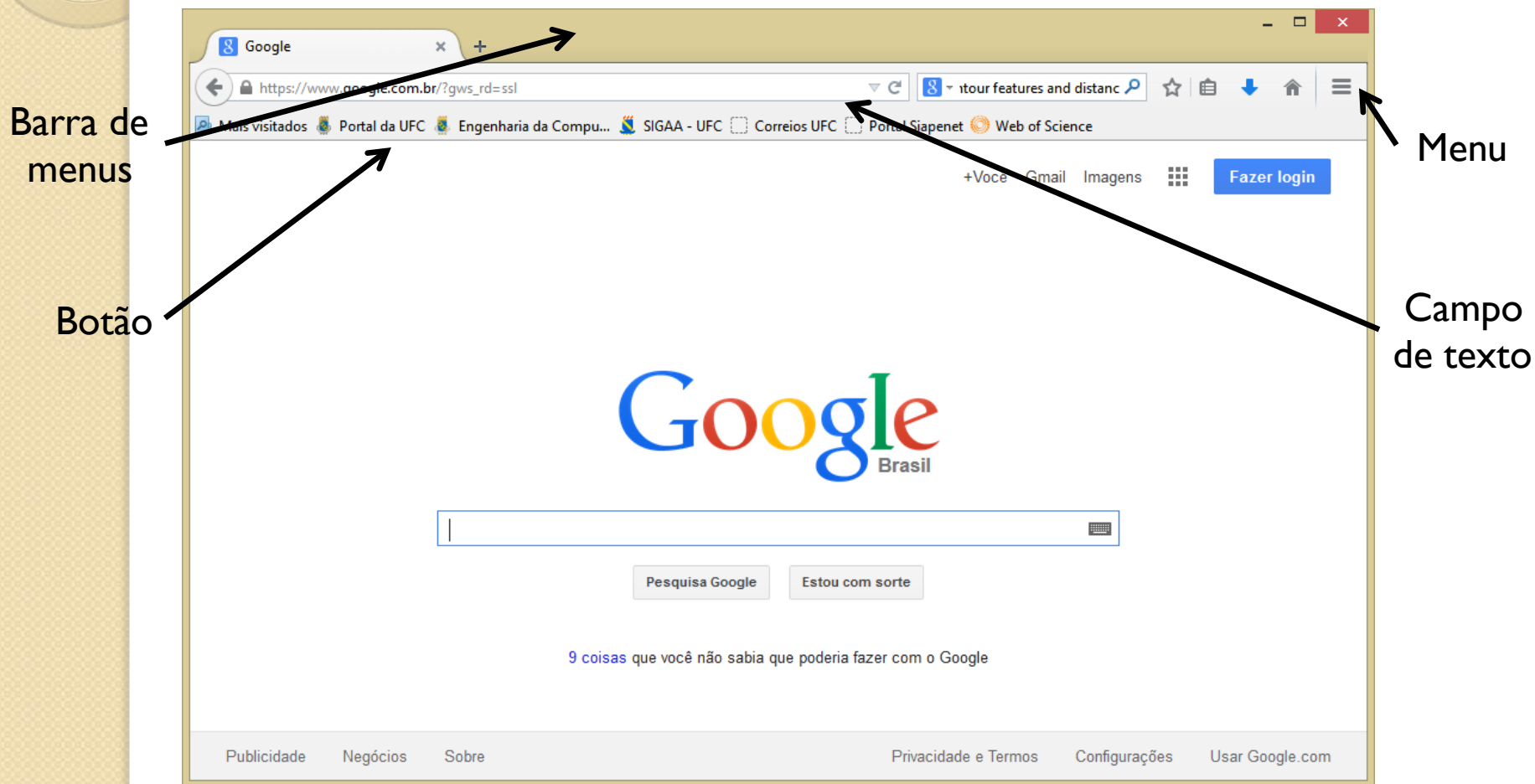
Introdução

- Componentes GUI em um browser:



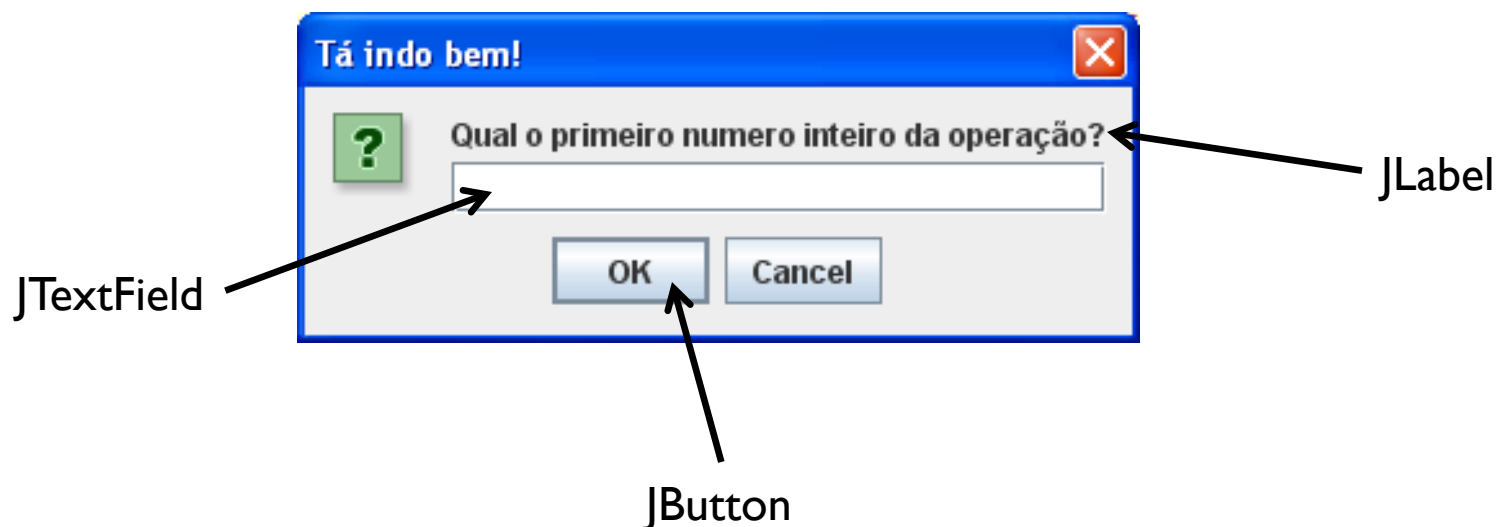
Introdução

- Componentes GUI em um browser:



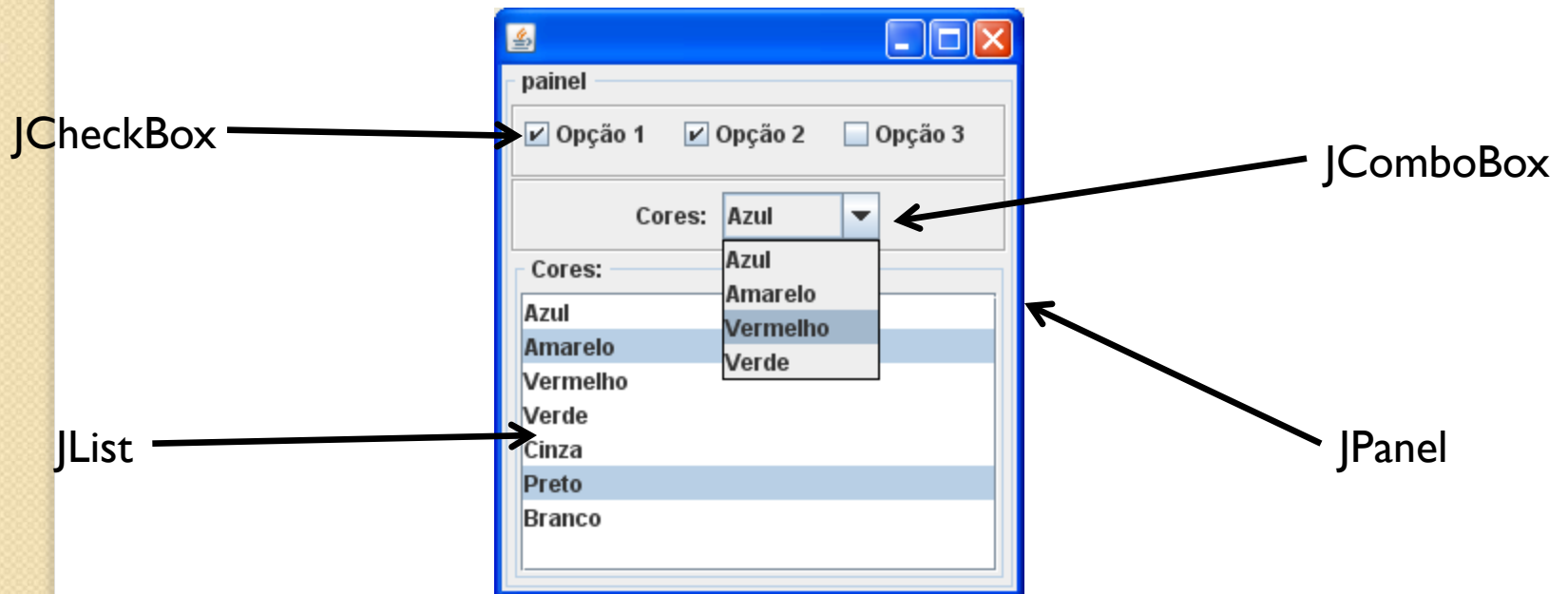
Introdução

- Componentes GUI – interação através do mouse, teclado, etc.
- Exemplos:
 - JLabel – área exibível para textos não-editáveis ou ícones;
 - JTextField – área para exibição e inserção de dados através do teclado;
 - JButton – área que aciona um evento após clique do mouse;



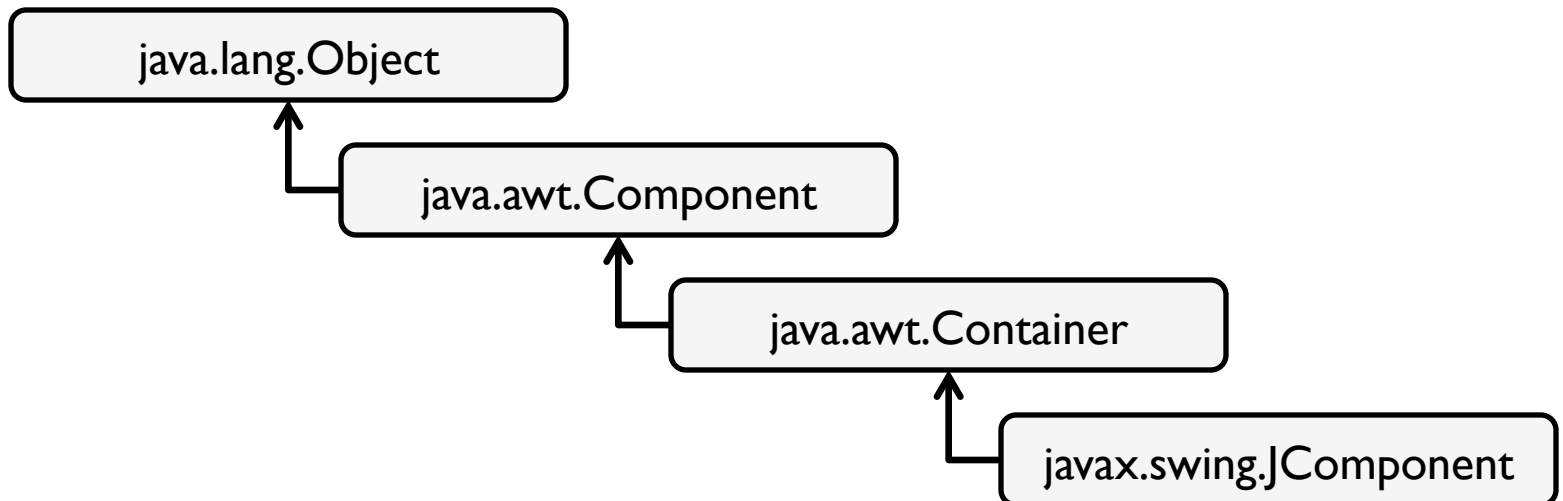
Introdução

- JCheckBox – componente com dois estados: selecionado ou não-selecionado;
- JComboBox – lista de itens em que apenas um é selecionado, por mouse ou teclado;
- JList – lista de itens em que mais de um podem ser selecionados;
- JPanel – contêiner em que os componentes podem ser colocados.



Visão geral do Swing

- Pacote ***javax.swing***
 - Escritos, manipulados e exibidos completamente em Java (componentes *Java puro*);
- Pacote ***java.awt***
 - Estão diretamente associados à GUI da plataforma local;
 - Exibidos de forma diferente, dependendo da plataforma;



Caixas de Diálogo

- No código AloMundo.java:
 - JOptionPane.showMessageDialog(...) – return **void**
 - Parâmetros:
 - **Component** parentComponent
 - **Object** message
 - **String** title (*sobrecarga 1*)
 - **int** messageType (*sobrecarga 1*)
 - **Icon** icon (*sobrecarga 2*)
 - Exemplo:
 - JOptionPane.showMessageDialog(null,"Alô, mundo!","Primeiro programa com interface gráfica",JOptionPane.DEFAULT_OPTION);

Caixas de Diálogo

```
package aula.igrafica;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
public class AloMundo extends JFrame {
    public AloMundo(){
        JOptionPane.showMessageDialog(null,"Alô, mundo!", "Primeiro
        programa com interface
        gráfica",JOptionPane.DEFAULT_OPTION);
    }
    public static void main(String args[]){
        AloMundo aplicacao = new AloMundo();
        aplicacao.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Caixas de Diálogo

- `JOptionPane.showConfirmDialog(...)` – return **int**
- Parâmetros:
 - **Component** parentComponent
 - **Object** message
 - **String** title (*sobrecarga 1*)
 - **int** optionType (*sobrecarga 1*)
 - **int** messageType (*sobrecarga 2*)
 - **Icon** icon (*sobrecarga 3*)
- Exemplo:
 - `JOptionPane.showConfirmDialog(null, "Alô, mundo!", "Primeiro programa com interface gráfica", JOptionPane.CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE);`

Caixas de Diálogo

- optionType
 - YES_NO_OPTION,
YES_NO_CANCEL_OPTION,
OK_CANCEL_OPTION
- messageType
 - ERROR_MESSAGE,
INFORMATION_MESSAGE,
WARNING_MESSAGE,
QUESTION_MESSAGE, PLAIN_MESSAGE

Caixas de Diálogo

- No código Calculadora.java:
- JOptionPane.showInputDialog(...) – return **String**
 - Parâmetros:

| Inicial | Sobrecarga 1 | Sobrecarga 2 |
|------------------------------------|----------------------------------|------------------------------------|
| Component parentComponent | Component parentComponent | Component parentComponent |
| Object message | Object message | Object message |
| Object initialSelectedValue | String title | String title |
| | int messageType | int messageType |
| | | Icon icon |
| | | Object[] selectionValue |
| | | Object initialSelectedValue |

- Exemplo:
 - JOptionPane.showInputDialog(null, "Qual o primeiro numero inteiro da operação?", "Tá indo bem!", JOptionPane.QUESTION_MESSAGE);

Caixas de Diálogo

- `JOptionPane.showOptionDialog(...)` – return **int**
- Parâmetros:
 - **Component** parentComponent
 - **Object** message
 - **String** title
 - **int** optionType
 - **int** messageType
 - **Icon** icon
 - **Object[]** options
 - **Object** initialValue
- Exemplo:
 - `String[] opcoes = {"Confirmar", "Cancelar", "Fechar"};`
 - `JOptionPane.showOptionDialog(null, "mensagem", "titulo",
JOptionPane.OK_CANCEL_OPTION,
JOptionPane.INFORMATION_MESSAGE, icon, opcoes,
opcoes[0]);`

Layout do Componente

- Aplicado em componentes containeres
- FlowLayout – objetos seguidos no container
- BorderLayout – objetos dispostos nas bordas e centro do container
 - CENTER, NORTH, SOUTH, EAST, WEST
- GridLayout – objetos dispostos como uma tabela (colunas e linhas)

Layout do Componente

- Qual o resultado do código?

```
FlowLayout layout = new FlowLayout();  
Container container = getContentPane();  
container.setLayout( layout );  
container.add(leftButton);  
container.add(centerButton);  
container.add(leftButton);  
setSize(300,75);  
setVisible(true);
```


Layout do Componente

- Qual o resultado do código?

```
BorderLayout layout = new BorderLayout();  
Container container = getContentPane();  
container.setLayout( layout );  
container.add(leftButton, BorderLayout.WEST);  
container.add(myList, BorderLayout.CENTER);  
container.add(otherButton,  
BorderLayout.SOUTH);  
setSize(300,75);  
setVisible(true);
```

Layout do Componente

- Qual o resultado do código?

```
GridLayout layout = new GridLayout(2,2);  
Container container = getContentPane();  
container.setLayout( layout );  
container.add(leftButton);  
container.add(myButton);  
container.add(otherButton);  
setSize(300,150);  
setVisible(true);
```

JFrame no NetBeans

- Novo -> JFrame
- `public Construtor() {
 initComponents();
}`
- `public void initComponents() { (...) }`
- Área de **Desenho** e de **Código**.

Tratamento de Eventos

- Mecanismo:
 - Origem do evento
 - Objeto evento
 - “Ouvinte” (listener) do evento
- Origem: componente GUI que interage (botão)
- Objeto: encapsula as informações sobre o evento.
- Ouvinte: recebe as informações do evento e responde este mesmo evento.

Tratamento de Eventos

- Tarefas do programador:
 - Registrar o ouvinte para o componente GUI;
 - Implementar o método de tratamento de eventos.
- Pacotes:
 - `java.awt.event`
 - `javax.swing.event`
- Teclar *Enter* irá chamar o método ***actionPerformed***.

Tratamento de Eventos

```
TextFieldHandler handler = new TextFieldHandler;  
textFieldI.addActionListener ( handler );
```

origem

objeto listener

```
private class TextFieldHandler implements ActionListener {
```

```
    // processa eventos
```

```
    public void actionPerformed ( ActionEvent event ) {
```

```
        if (event.getSource() == textFieldI)
```

```
            String saida = "TextFieldI clicado!";
```

```
    }
```

```
}
```

objeto evento