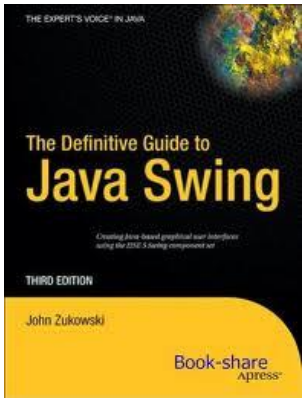
Programação Orientada a Objetos

Unidade 7 - Interface Gráfica

Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP

Bibliografia

- The Definitive Guide to Java Swing – Third edition – John Zukowski – 2005 - Apress
- Beginning Java 2 – Ivor Horton – 2011 WROX
- Java – The Complete Reference – 8th Edition – Herbert Schildt – Oracle Press - 2011
- Core Java Fundamentals – Horstmann / Cornell – PTR- Volumes 1 e 2 – 8th Edition
- Inside the Java 2 – Virtual Machine Venners – McGrawHill
- Understanding Object-Oriented Programming with JAVA – Timothy Budd – Addison Wesley
- Head First Java, 2nd Edition by Kathy Sierra and Bert Bates
- Effective Java, 2nd Edition by Joshua Bloch - 2008
- Thinking in Java (4th Edition) by Bruce Eckel
- Java How to Program - 9th Edition by Paul Deitel and Harvey Deitel



Introdução

- A API Java Swing inclui suporte para diversas coisas que você geralmente vê em interfaces não-web: janelas, botões, menus, etc...
- Swing utiliza a **JFC** – Java Foundation Class, um toolkit que suporta a criação de aplicações que podem ser executadas em diferentes sistemas operacionais.

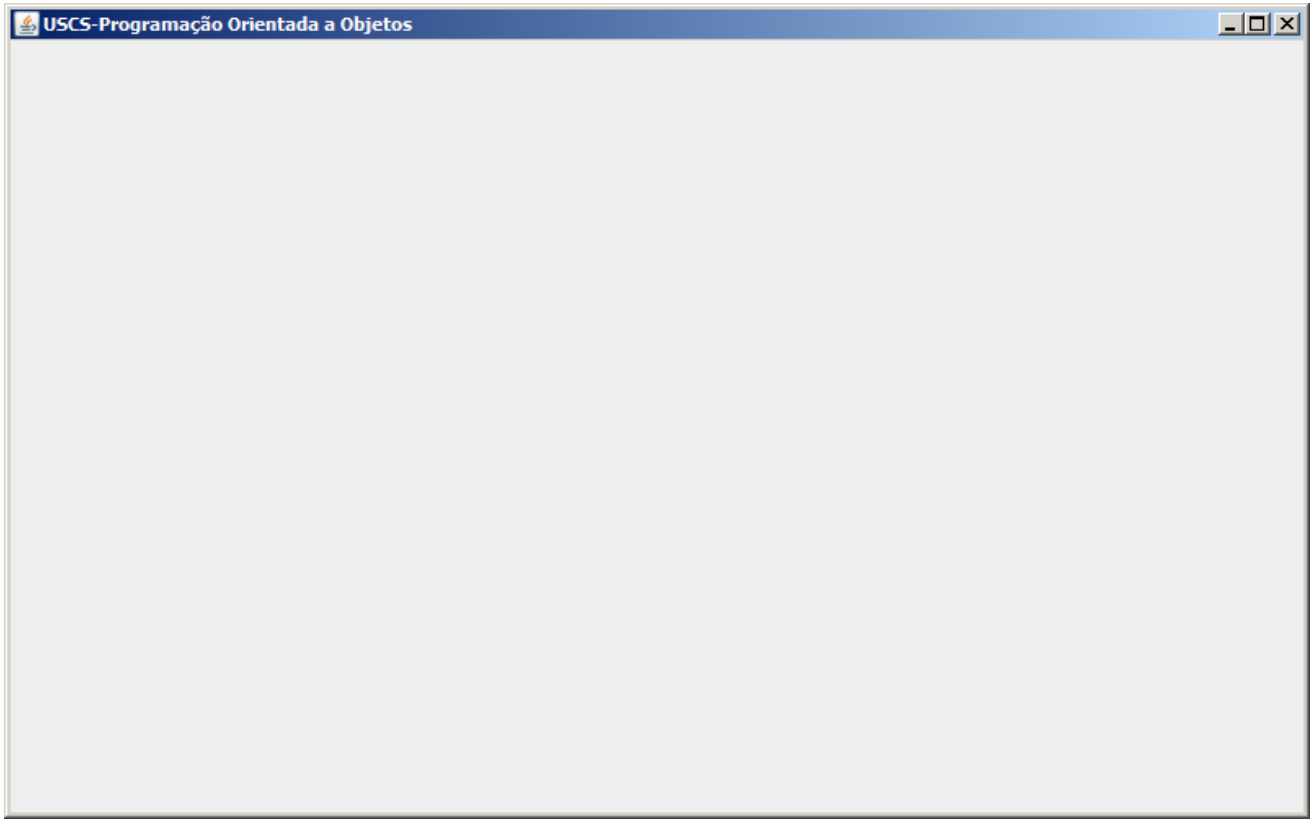
Uma simples aplicação Swing

- Aplicações **Swing** seguem um fluxo básico para o desenvolvimento.
- Primeiro, cria-se um objeto **JFrame** que corresponde à janela principal e dá suporte aos outros componentes da interface (menu, botões, etc.)

Uma simples aplicação Swing

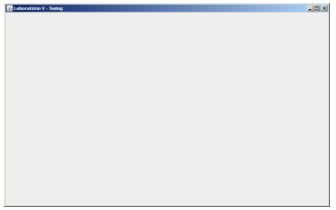
```
package swing;
import javax.swing.*;
import java.awt.*;

public class JFrame_01 {

    private void CriaJanela() {

        JFrame frame = new JFrame("USCS-Programação Orientada a Objetos");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(new Dimension(800,500));
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        JFrame_01 app = new JFrame_01();
        app.CriaJanela();
    }
}
```

Uma simples aplicação Swing



Uma simples aplicação Swing

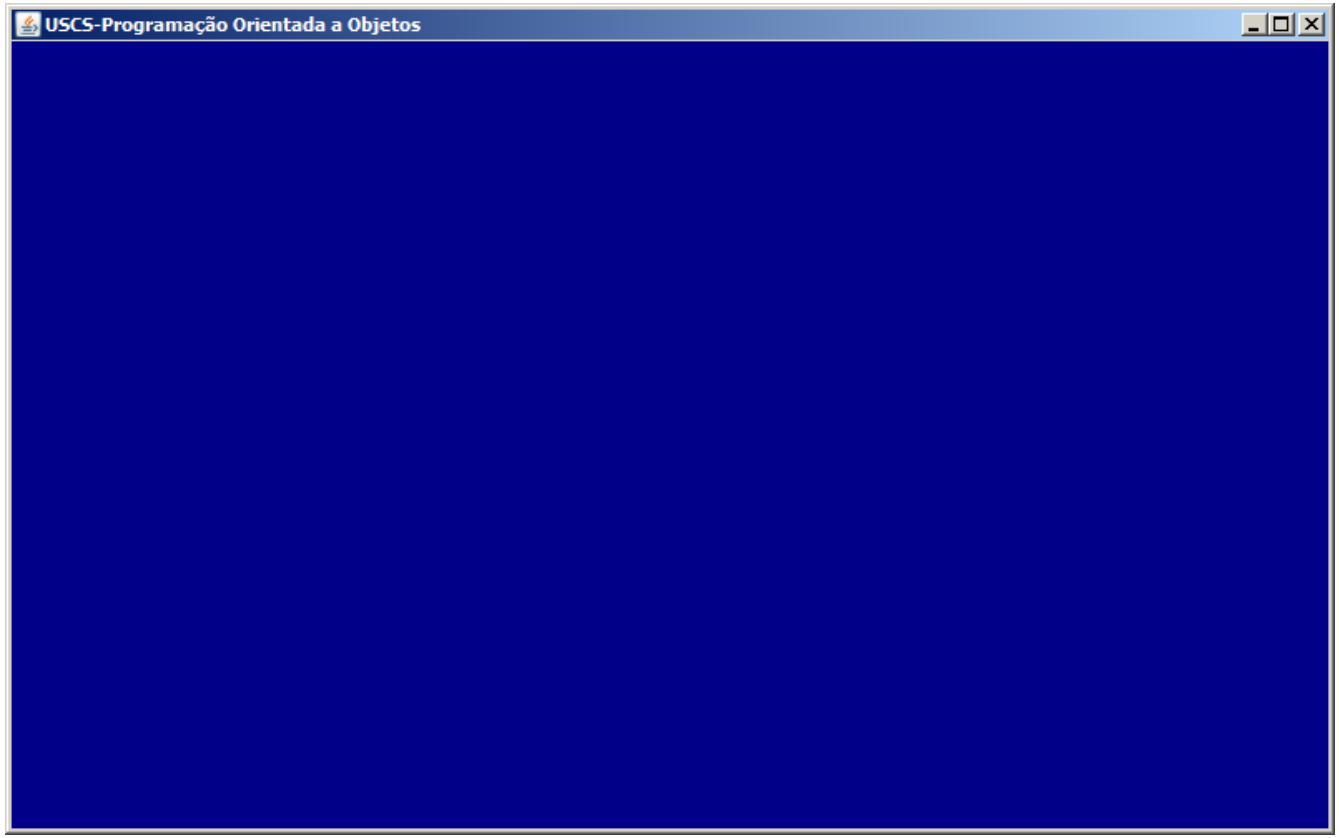
- O que o programa anterior faz é simplemente criar e exibir uma janela vazia.
- O objeto frame define a janela.
- Em seguida, definimos alguns atributos da janela (tamanho e forma de fechar a janela).
- O método setVisible define se o componente será visível ou não.



Modificando a cor da janela

```
package swing;
import javax.swing.*;
import java.awt.*;

public class JFrame_02 {

    private void CriaJanela() {

        JFrame frame = new JFrame("USCS-Programação Orientada a Objetos");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(new Dimension(800,500));
        Color color = new Color(0x000088);
        frame.getContentPane().setBackground(color);

            frame.setVisible(true);
    }

    public static void main(String[] args) {
        JFrame_02 app = new JFrame_02();
        app.CriaJanela();
    }
}
```

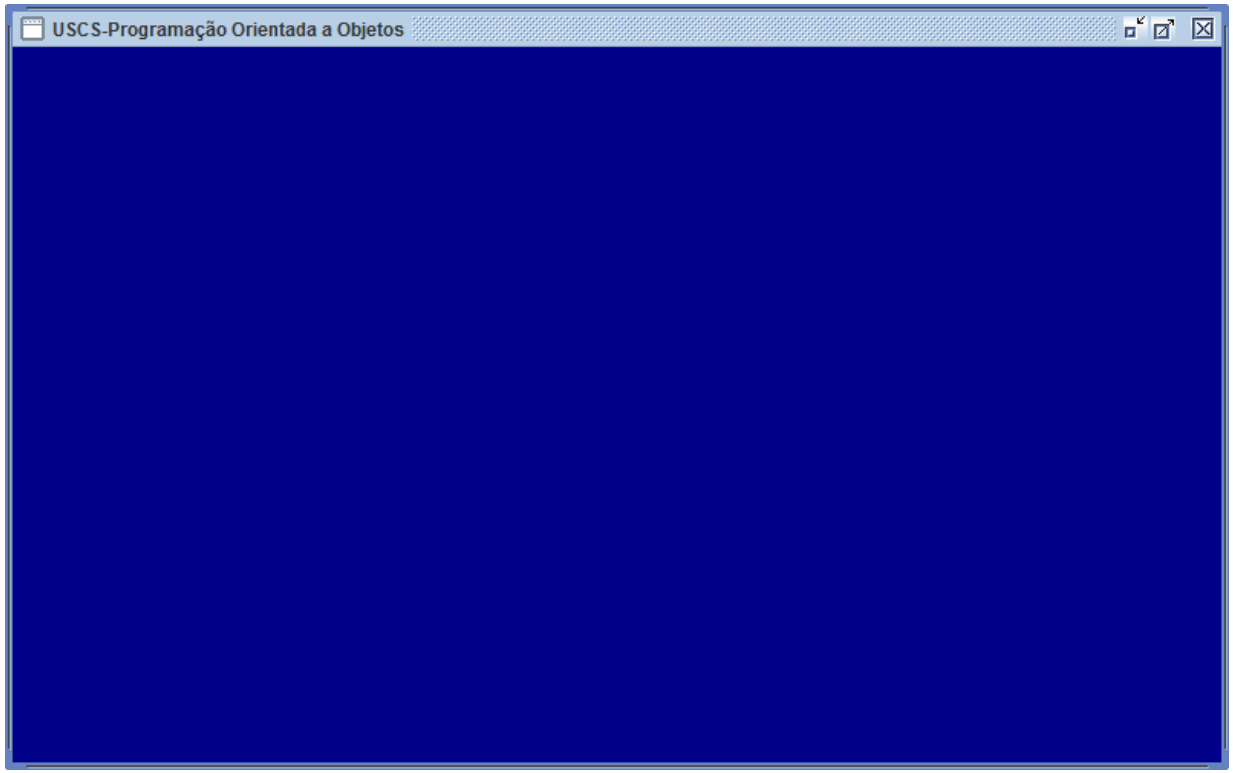
Modificando a cor da janela



Uma janela mais bonita

```
package swing;
import javax.swing.*;
import java.awt.*;

public class JFrame_03 {

    private void CriaJanela() {

        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new JFrame("USCS-Programação Orientada a Objetos");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Color color = new Color(0x000088);
        frame.getContentPane().setBackground(color);
        frame.setSize(new Dimension(800,500));
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        JFrame_03 app = new JFrame_03();
        app.CriaJanela();
    }
}
```

Uma janela mais bonita

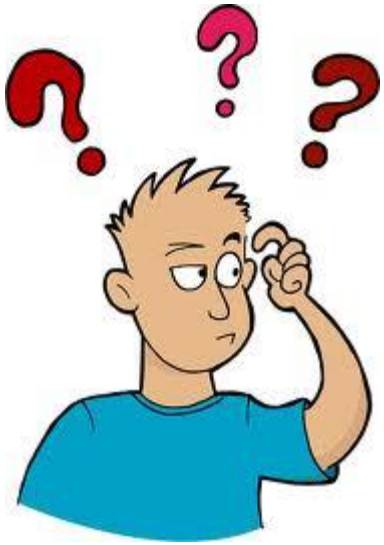


Como centralizo a janela no computador ?

Centralizando a janela

```
package swing;
import javax.swing.*;
import java.awt.*;

public class JFrame_04 {

    private void CriaJanela() {

        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new JFrame("USCS-Programação Orientada a Objetos");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Color color = new Color(0x000088);
        frame.getContentPane().setBackground(color);
        frame.setSize(new Dimension(800,500));
        frame.setVisible(true);
        frame.setLocationRelativeTo(null);

    }

    public static void main(String[] args) {
        JFrame_04 app = new JFrame_04();
        app.CriaJanela();
    }
}
```

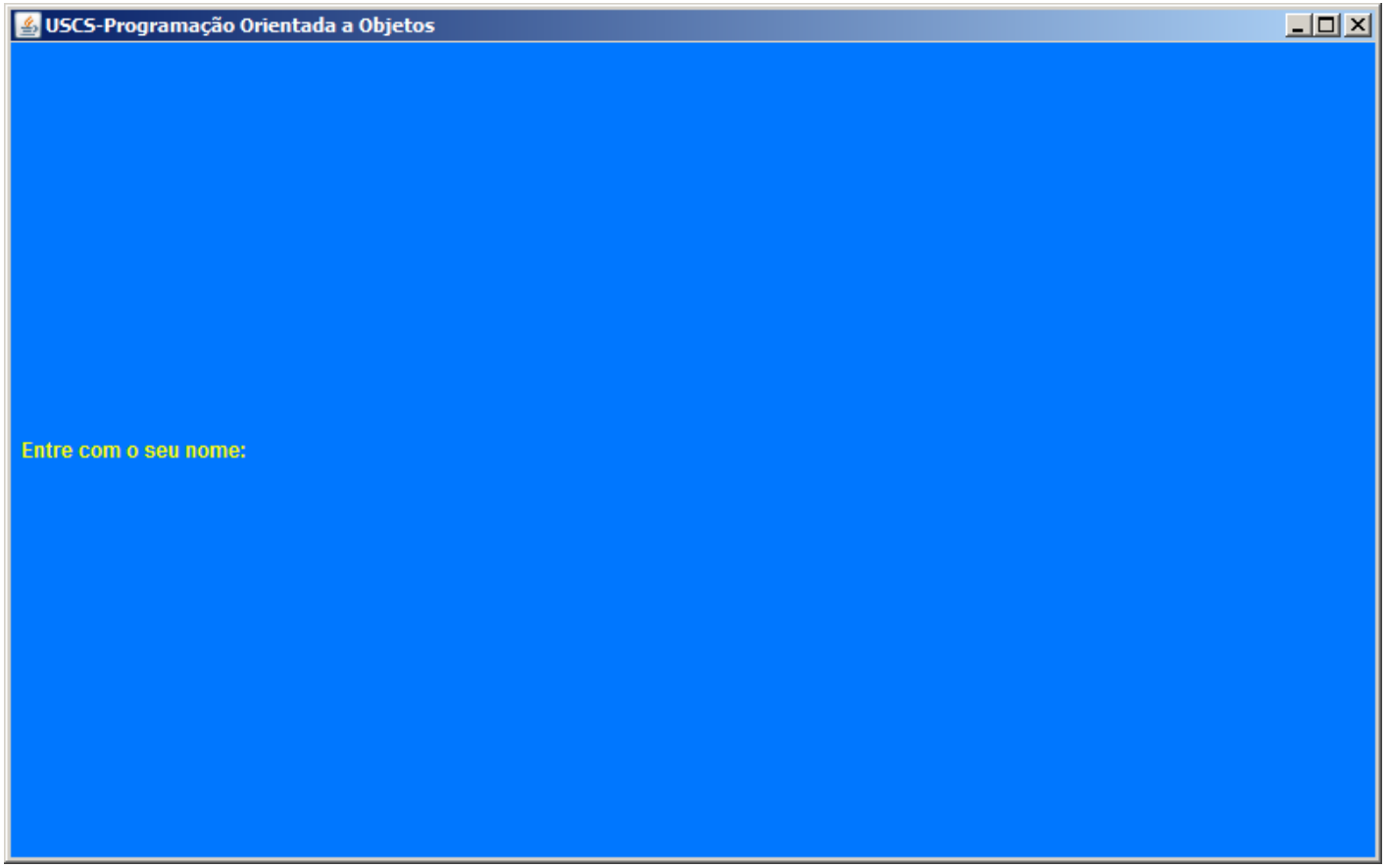
Como acrescento um label à janela ?



Acrescentando label

```
package swing;

import java.awt.Color;

import javax.swing.JFrame;
import javax.swing.JLabel;

public class JFrame_05 extends JFrame {
        public JFrame_05() {
                super("USCS-Programação Orientada a Objetos");
                JLabel labelNome;
                labelNome = new JLabel("  Entre com o seu nome: ");
                labelNome.setForeground(Color.yellow);

                this.add(labelNome);

                this.setSize(800,500);
                this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                Color color = new Color(0x0077FF);
                this.getContentPane().setBackground(color);

                this.setLocationRelativeTo(null);

                this.setVisible(true);
        }
  public static void main(String[] args) {
        JFrame_05 app = new JFrame_05();

  }
}
```

Acrescentando label

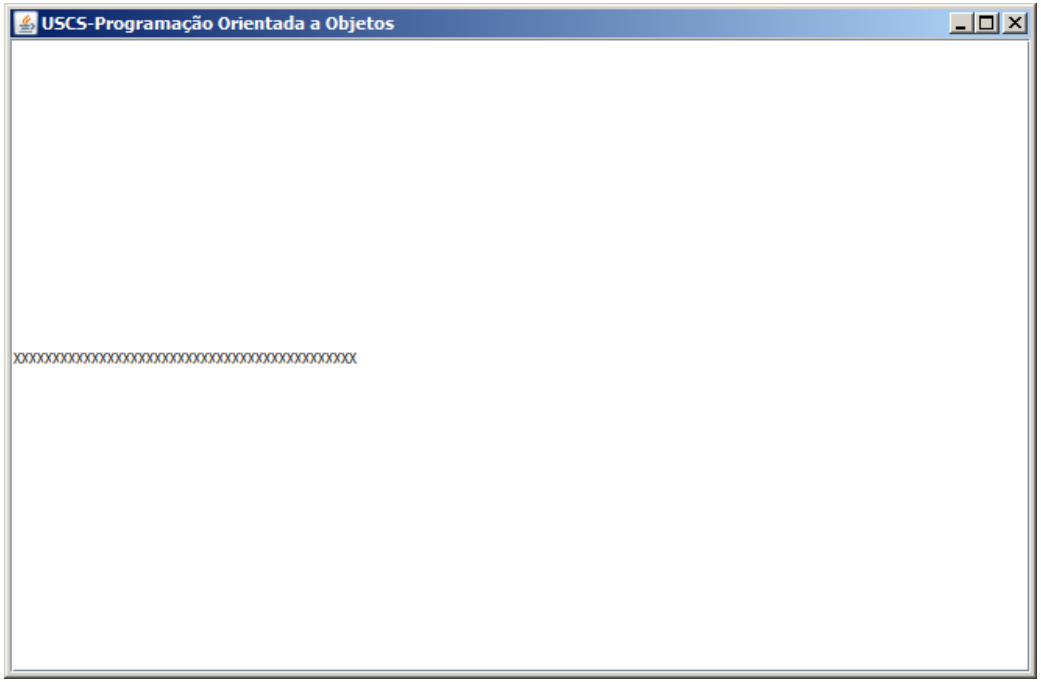


Como acrescento um campo de texto à janela ?

Acrescentando textField

```
package swing;

import java.awt.*;
import javax.swing.*;
public class JFrame_06 extends JFrame {
          public JFrame_06() {
                    super("USCS-Programação Orientada a Objetos");
                    JLabel labelNome;
                    labelNome = new JLabel("  Entre com o seu nome: ");
                    labelNome.setForeground(Color.yellow);
                    this.add(labelNome);

                    JTextField textFieldNome = new JTextField(30);
                    this.add(textFieldNome);

                    this.setSize(800,500);
                    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                    Color color = new Color(0x0077FF);
                    this.getContentPane().setBackground(color);

                    this.setLocationRelativeTo(null);

                    this.setVisible(true);
          }
  public static void main(String[] args) {
          JFrame_06 app = new JFrame_06();

  }
}
```

Acrescentando textField



Xi ????

Mas o label que eu havia feito antes, desapareceu ! Por que?

Gerenciamento de Layout

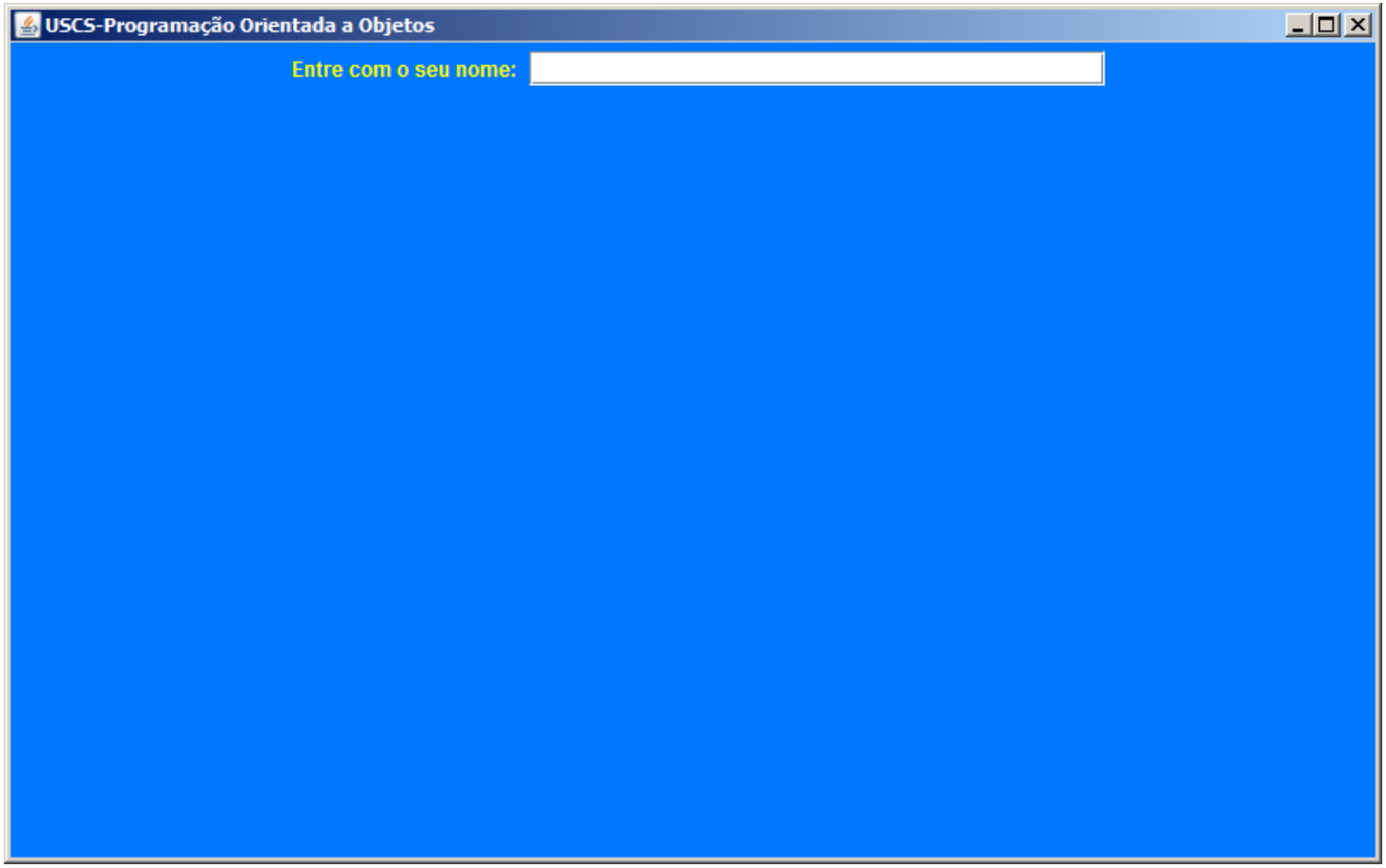
- À medida em que vamos inserindo componentes no Frame, será necessário um modo de gerenciar estes componentes na tela e assim darmos funcionalidade à interface.
- Com isso, iremos organizar os componentes no layout, posicionando-os de forma mais adequada na interface.

Como implementar um gerenciamento de Layout ?

Gerenciamento de Layout

```
package swing;

import java.awt.*;
import javax.swing.*;
public class JFrame_07 extends JFrame {
        public JFrame_07() {
                super("USCS-Programação Orientada a Objetos");
                FlowLayout layout = new FlowLayout();
                this.setLayout(layout);

                JLabel labelNome;
                labelNome = new JLabel("  Entre com o seu nome: ");
                labelNome.setForeground(Color.yellow);
                this.add(labelNome);

                JTextField textFieldNome = new JTextField(30);
                this.add(textFieldNome);

                this.setSize(800,500);
                this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                Color color = new Color(0x0077FF);
                this.getContentPane().setBackground(color);

                this.setLocationRelativeTo(null);

                this.setVisible(true);
        }
  public static void main(String[] args) {
        JFrame_07 app = new JFrame_07();

  }
}
```

Gerenciamento de Layout



Como acrescento um botão à janela ?

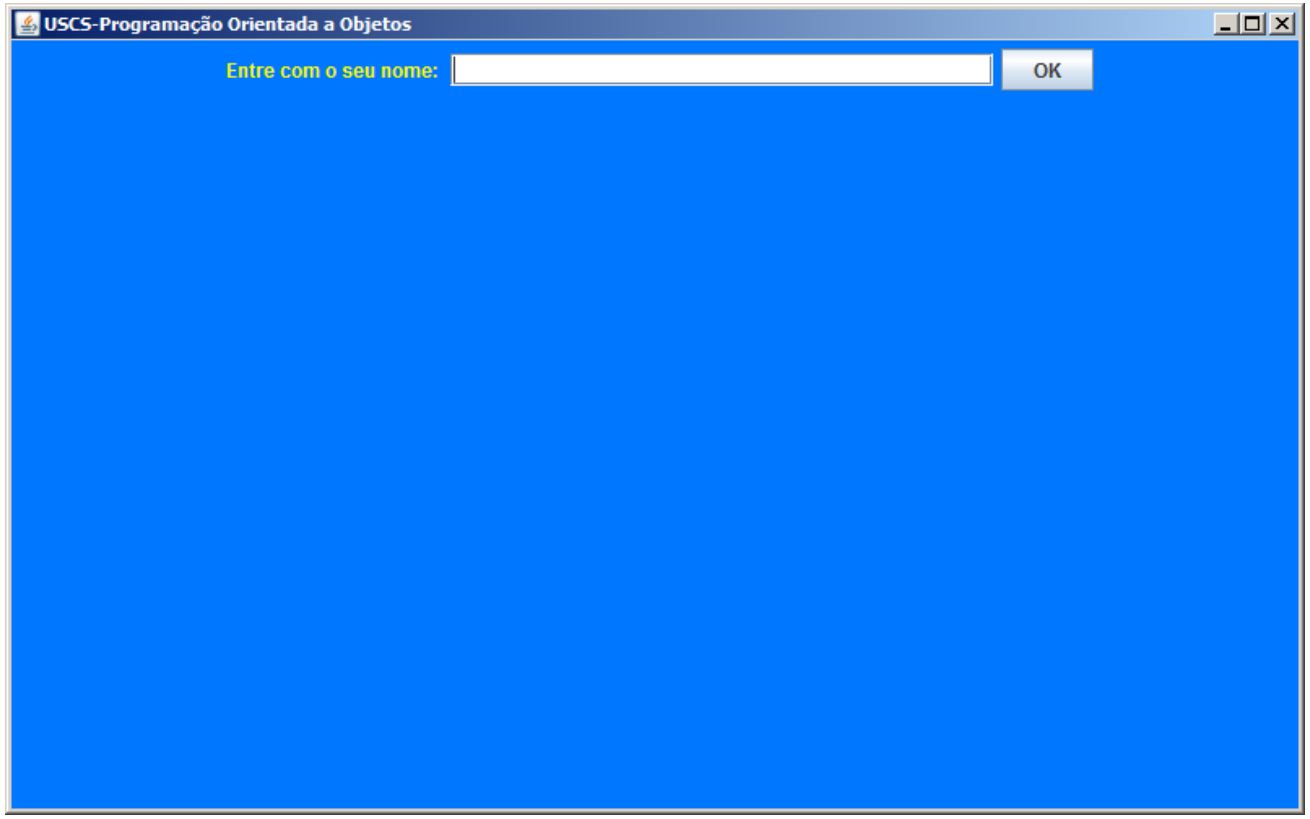
Incluindo um botão

```
package swing;

import java.awt.*;
import javax.swing.*;

public class JFrame_08 extends JFrame {
        public JFrame_08() {

                super("USCS-Programação Orientada a Objetos");
                FlowLayout layout = new FlowLayout();
                this.setLayout(layout);

                JLabel labelNome;
                labelNome = new JLabel("  Entre com o seu nome: ");
                labelNome.setForeground(Color.yellow);
                this.add(labelNome);

                JTextField textFieldNome = new JTextField(30);
                this.add(textFieldNome);
```

Incluindo um botão

```
        JButton botao_OK = new JButton(" OK ");
        this.add(botao_OK);

        this.setSize(800,500);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Color color = new Color(0x0077FF);
        this.getContentPane().setBackground(color);

        this.setLocationRelativeTo(null);

        this.setVisible(true);

    }
        public static void main(String[] args) {
                JFrame_08 app = new JFrame_08();

  }
}
```

Incluindo um botão



Porque ao clicar no botão nada acontece ?

Tratamento de Eventos

- Para implementarmos uma funcionalidade no botão precisamos gerar o código que trate o evento de clique do usuário no componente.
- Iremos implementar uma classe responsável pelo tratamento dos eventos no botão.
- Esta classe implementa funções da interface ActionListener.

Tratamento de Eventos

```
package swing;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;

public class JFrame_09 extends JFrame {

          private JLabel labelNome;
          private JTextField textFieldNome;
          private FlowLayout layout;
          private JButton botao_OK;

          public JFrame_09() {
                    super("USCS - Lab V - Frame centralizado");
                    this.layout = new FlowLayout();
                    this.setLayout(layout);
```

Tratamento de Eventos

```
labelNome = new JLabel("  Entre com o seu nome: ");
labelNome.setForeground(Color.yellow);
this.add(labelNome);

this.textFieldNome = new JTextField(30);
this.add(textFieldNome);
this.botao_OK = new JButton(" OK ");
this.add(botao_OK);

ButtonHandler handler = new ButtonHandler();
botao_OK.addActionListener(handler);
this.setSize(800,500);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
Color color = new Color(0x0077FF);
this.getContentPane().setBackground(color);

this.setLocationRelativeTo(null);

this.setVisible(true);

}
```

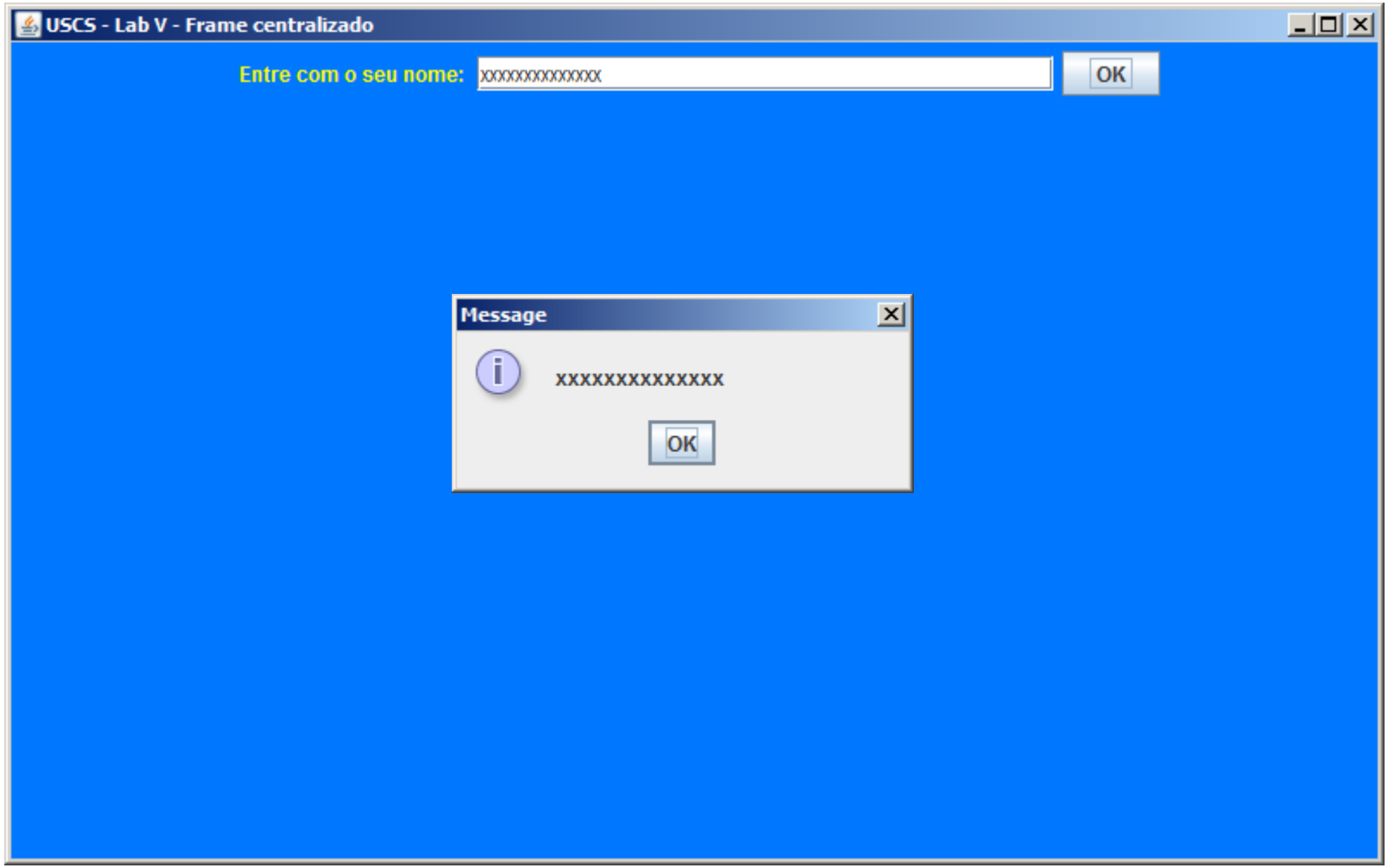
Tratamento de Eventos

```
class ButtonHandler implements ActionListener {
@Override
        public void actionPerformed(ActionEvent event) {
                if (event.getSource() == botao_OK) {
                        String nome = textFieldNome.getText();
                        JOptionPane.showMessageDialog(null, nome);
                }
        }

}

  public static void main(String[] args) {
        JFrame_09 app = new JFrame_09();
  }
}
```

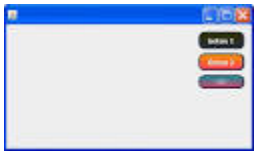
Tratamento de Eventos

```
    public void center() {

            Toolkit tk = Toolkit.getDefaultToolkit();
            Dimension screensize = tk.getScreenSize();
            int screenWith = screensize.width - this.getWidth();
            int screenHeigth = screensize.height - this.getHeight();
            setLocation(screenWith/2,screenHeigth/2);
    }

    public static void main(String[] args) {
                    JFrame_07 app = new JFrame_07();
    }
}
```

Tratando o evento do botão



Exemplo com 2 botões



Exemplo com 2 botões


```
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;

public class JFrame_10 extends JFrame {

        private JLabel labelNome;
        private JTextField textFieldNome;
        private FlowLayout layout;
        private JButton botao_OK;
        private JButton botao_FIM;

        public JFrame_10() {
                super("USCS - Lab V - Frame centralizado");
                this.layout = new FlowLayout();
                this.setLayout(layout);


                labelNome = new JLabel("  Entre com o seu nome: ");
                labelNome.setForeground(Color.yellow);
                this.add(labelNome);
```

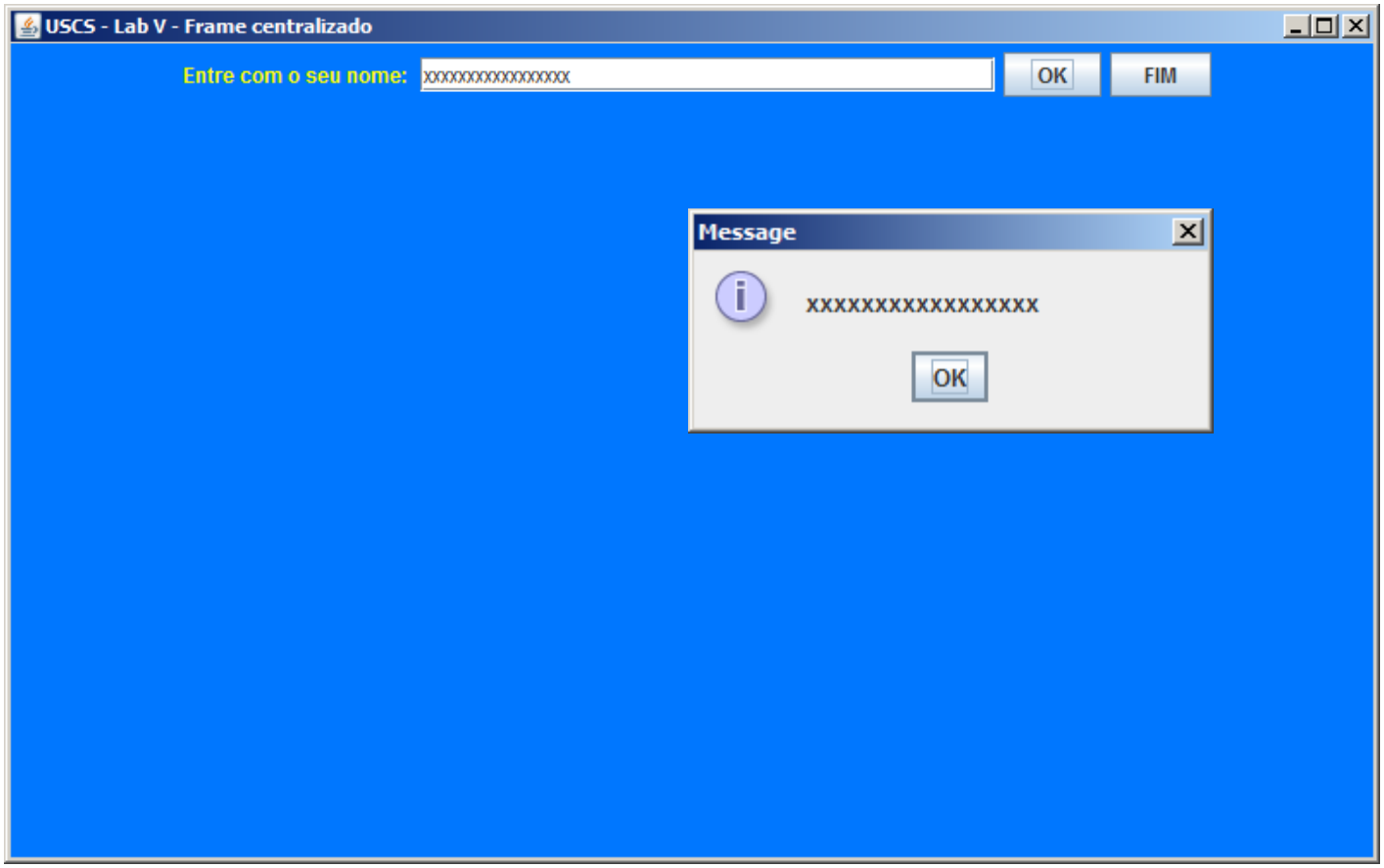
Exemplo com 2 botões

```
this.textFieldNome = new JTextField(30);
this.add(textFieldNome);

this.botao_OK = new JButton(" OK ");
this.add(botao_OK);


this.botao_FIM = new JButton(" FIM ");
this.add(botao_FIM);

ButtonHandler handler = new ButtonHandler();

botao_OK.addActionListener(handler);
botao_FIM.addActionListener(handler);

this.setSize(800,500);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
Color color = new Color(0x0077FF);
this.getContentPane().setBackground(color);

this.setLocationRelativeTo(null);

this.setVisible(true);

}
```

Exemplo com 2 botões

```
class ButtonHandler implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent event) {
                if (event.getSource() == botao_OK) {
                        String nome = textFieldNome.getText();
                        JOptionPane.showMessageDialog(null, nome);
                }
                else if (event.getSource() == botao_FIM)  {
                        System.out.println("Botao FIM pressionado....");
                        System.out.println("Fim de programa....");
                        System.exit(NORMAL);
                }
        }

}
        public static void main(String[] args) {
                JFrame_10 app = new JFrame_10();
        }
}
```

Exemplo com 2 botões



Como acrescento um menu à janela ?

Programa Swing com Menu


```
package swing;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Menu_01 implements ActionListener {

        private JFrame frame = new JFrame(" Laboratório V - MENUS ");

        private void addMenu(JFrame frame) {
                JMenu file = new JMenu("File");
                file.setMnemonic('F');
                JMenuItem exitItem = new JMenuItem("Exit");
                exitItem.setMnemonic('x');
                exitItem.addActionListener(this);
                file.add(exitItem);
                JMenuBar menuBar = new JMenuBar();
                menuBar.add(file);
                frame.setJMenuBar(menuBar);
}
```

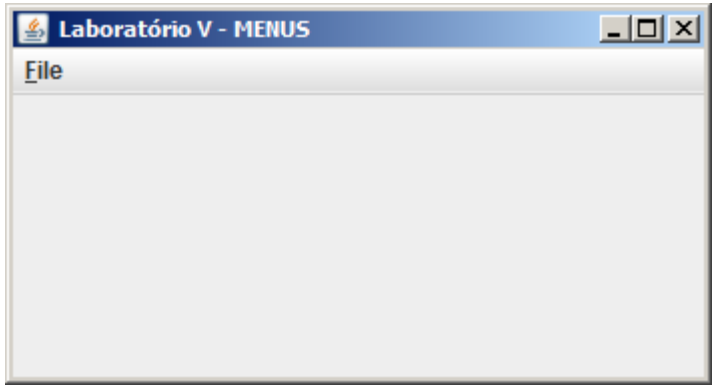
Programa Swing com Menu

```
    private void Exibe_Window() {
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setPreferredSize(new Dimension(800, 500));
            addMenu(frame);
            frame.pack();
            frame.setVisible(true);
    }

    public static void main(String[] args) {
            Menu_01 swing_Menu = new Menu_01();
            swing_Menu.Exibe_Window();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
            if (e.getActionCommand().equals("Exit")) {
            System.exit(0);
    }
   }
}
```

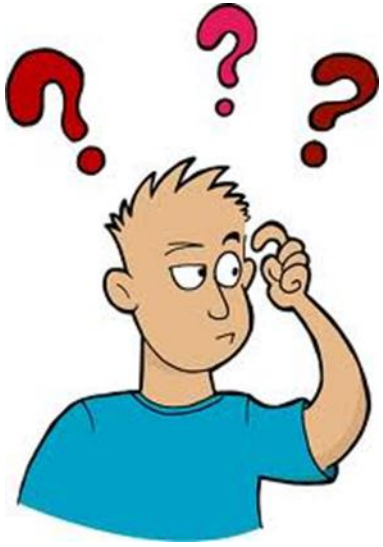
Programa Swing com Menu

- A adição de menu está encapsulada no método **addMenu**.
- O método **addMenu** cria um menu chamado "**File**" e em seguida um item de menu chamado "**Exit**".
- Finalmente, o método **addMenu** cria uma barra de menu e adiciona-a ao item de menu e o liga ao objeto frame (window).
- O método **Exibe_Window** invoca o método **addMenu** para incorporar o menu na janela.

Programa Swing com Menu



Porque o código implementa ActionListener ?



ActionListener

- A classe escrita implementa a interface **ActionListener**.
- Um **listener**, como o nome indica, monitora os eventos no programa e permite que seja especificada alguma ação a ser tomada quando o evento ocorrer.
- No nosso caso, queremos que o programa encerre quando o usuário escolher Exit no menu.
- Java oferece diversos listeners tais como: mouse, teclado, etc., todos eles estendendo a interface **EventListener**.