

MÓDULO 4 – SWING – Parte 3

// Exemplo14 - JTable

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

public class Ex14_JTable extends JFrame implements ActionListener{

    private static final long serialVersionUID = 1L;
    private JButton btnExcluir, btnIncluir, btnMostrar;
    private JTextField texto;
    private JTable tabela;
    private int incCod = 0;

    public Ex14_JTable(){
        setTitle("Tabela");
        setSize(441,300);
        setLocation(100,50);
        Container P = getContentPane();
        P.setLayout(new BorderLayout());
        tabela = new JTable();
        tabela.setModel(new DefaultTableModel(    new Object [][] { },
        new String [] { "ID", "Nome" }));

        //define largura das colunas -- permite ou não, ajustar a largura
        tabela.getColumnModel().getColumn(0).setPreferredWidth(20);
        tabela.getColumnModel().getColumn(0).setResizable(false);
        tabela.getColumnModel().getColumn(1).setPreferredWidth(150);
        tabela.getColumnModel().getColumn(1).setResizable(true);

        JScrollPane rolagemTabela = new JScrollPane(tabela);
        P.add(rolagemTabela, "Center");

        // criação do Painel de baixo
        JPanel PTabSul = new JPanel();
        btnIncluir = new JButton("Incluir");
        PTabSul.add(btnIncluir);
        btnExcluir = new JButton("Excluir");
        PTabSul.add(btnExcluir);
        btnMostrar = new JButton("Mostrar");
        PTabSul.add(btnMostrar);
        texto = new JTextField("Nome Selecionado");    PTabSul.add(texto);
        P.add(PTabSul, "South");
        btnExcluir.addActionListener(this);
        btnIncluir.addActionListener(this);
        btnMostrar.addActionListener(this);
    }

    public void actionPerformed(ActionEvent evt){
        Object origem = evt.getSource();
        if (origem == btnIncluir){
            DefaultTableModel dtm = (DefaultTableModel)tabela.getModel();
            dtm.addRow(new Object[]{
                new Integer(++incCod), "Cliente " + incCod});}

        if (origem == btnExcluir){
```

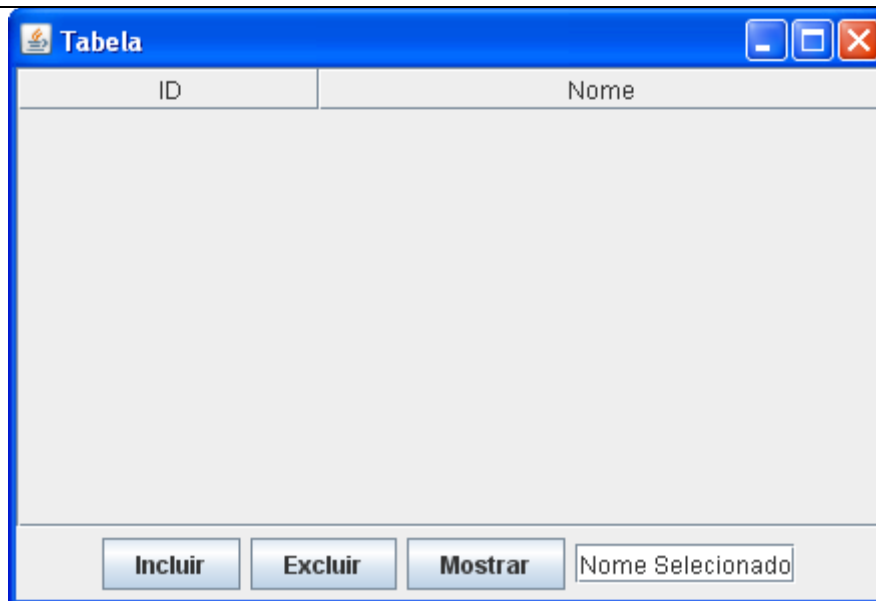
```

        int linhas[] = tabela.getSelectedRows();
        DefaultTableModel dtm = (DefaultTableModel)tabela.getModel();
        for(int i = (linhas.length - 1); i >= 0; i--)
            dtm.removeRow(linhas[i]);
    }

    if (origem == btnMostrar){
        if (tabela.getSelectedRow()>=0)
            texto.setText(tabela.getValueAt(tabela.getSelectedRow(),
            tabela.getSelectedColumn()).toString());
    }
}

public static void main(String args[]) {
    Ex14_JTable fr = new Ex14_JTable();
    fr.setVisible(true);
}
}

```



// Exemplo15 - JTabbedPane

```

// Demonstrando JTabbedPane.
import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JTabbedPane;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JButton;
import javax.swing.SwingConstants;

public class Ex15_JTabbedPane extends JFrame{

    private static final long serialVersionUID = 1L;

    public Ex15_JTabbedPane(){
        super( "JTabbedPane Demo " );

        JTabbedPane tabbedPane = new JTabbedPane(); // cria JTabbedPane

        // configura o painel e o adiciona ao JTabbedPane
        JLabel label1 = new JLabel( "panel one", SwingConstants.CENTER );
        JPanel panel1 = new JPanel(); // cria o primeiro painel
        panel1.add( label1 ); // adiciona o rótulo ao painel
        tabbedPane.addTab( "Tab One" , null, panel1, "Primeiro Panel" );
    }
}

```

```

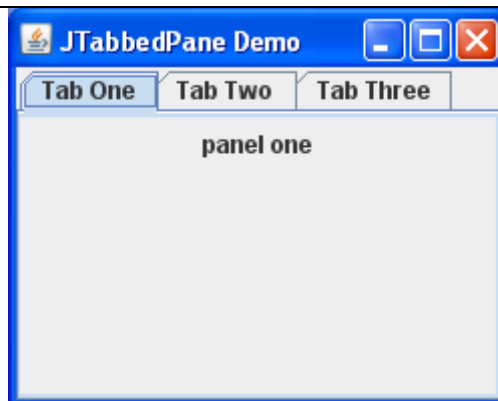
//configura panel2 e o adiciona ao JTabbedPane
JLabel label2 = new JLabel( "panel two", SwingConstants.CENTER );
JPanel panel2 = new JPanel(); // cria o segundo painel
panel2.setBackground( Color.YELLOW ); // configura fundo como amarelo
panel2.add( label2 ); // adiciona o rótulo ao painel
tabbedPane.addTab( "Tab Two", null, panel2, "Segundo Panel" );

// configura o panel3 e o adiciona ao JTabbedPane
JLabel label3 = new JLabel( "panel three" );
JPanel panel3 = new JPanel(); // cria o terceiro painel
panel3.setLayout( new BorderLayout() ); // utiliza o borderlayout
panel3.add( new JButton( "North" ), BorderLayout.NORTH );
panel3.add( new JButton( "West" ), BorderLayout.WEST );
panel3.add( new JButton( "East" ), BorderLayout.EAST );
panel3.add( new JButton( "South" ), BorderLayout.SOUTH );
panel3.add( label3, BorderLayout.CENTER );
tabbedPane.addTab( "Tab Three", null, panel3, "Terceiro Panel" );

add( tabbedPane ); // adiciona JTabbedPane ao frame
} // fim do construtor JTabbedPaneFrame

public static void main( String args[] ){
    Ex15_JTabbedPane tabbedPaneFrame = new Ex15_JTabbedPane();
    tabbedPaneFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    tabbedPaneFrame.setSize( 250, 200 ); // configura o tamanho do frame
    tabbedPaneFrame.setVisible( true ); // exhibe o frame
} // fim do main
} // fim da classe JTabbedPaneFrame

```



// Exemplo16 - JSlider

```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JSlider;
import javax.swing.SwingConstants;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

public class Ex16_JSlider extends JFrame{

    private static final long serialVersionUID = 1L;
    private JSlider diameterJSlider; // controle deslizante p/ selecionar o
diâmetro
    private Ex16_OvalPanel myPanel; // pa painel nel para desenhar o círculo

    // construtor sem argumentos
    public Ex16_JSlider () {

```

```

        super( "Slider Demo" );
        myPanel = new Ex16_OvalPanel(); // cr cria ia painel para desenhar o
círculo
        myPanel.setBackground(Color.YELLOW ); // configura o fundo como
amarelo

        // configura o JSlider para controlar o valor diâmetro
        diameterJSlider = new JSlider( SwingConstants.HORIZONTAL, 0, 200, 10
);
        diameterJSlider.setMajorTickSpacing( 10 ); // cria uma marca de medida
a cada 10
        diameterJSlider.setPaintTicks( true ); // pinta marcas no controle
deslizante

        // registra o ouvinte de evento do JSlider
        diameterJSlider.addChangeListener(

                new ChangeListener() // classe interna anônima
                {

                        // trata da alteração de valor do controle deslizante

                        public void stateChanged( ChangeEvent e ){
                                myPanel.setDiameter( diameterJSlider.getValue());
                                System.out.println(diameterJSlider.getValue());
                        } // fim do método stateChanged
                } // fim da classe interna anônima
        ); // fim da chamada para addChangeListener

        add( diameterJSlider, BorderLayout.SOUTH ); // ad adiciona iciona o
controle deslizante ao frame
        add( myPanel, BorderLayout.CENTER ); // ad adiciona iciona o painel ao
frame
    } // fim do construtor SliderFrame

    public static void main( String args[] ){
        Ex16_JSlider sliderFrame = new Ex16_JSlider ();
        sliderFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        sliderFrame.setSize( 220, 270 ); // configura o tamanho do frame
        sliderFrame.setVisible( true ); // exhibe o frame
    }
} // fim da class classe SliderFrame

class Ex16_OvalPanel extends JPanel {

    private static final long serialVersionUID = 1L;
    // private static final long serialVersionUID = 1L;
    private int diameter = 10; // diâmetro padrão de 10

    // desenha uma oval do diâmetro especificado
    public void paintComponent( Graphics g ){
        super.paintComponent( g );
        g.fillOval( 10, 10, diameter, diameter ); // desenha um círculo
    } // fim do método paintComponent

    // valida e configura o diâmetro, então pinta novamente
    public void setDiameter( int newDiameter ){
        // se o diâmetro for inválido , assume o padrão de 10
        diameter = ( newDiameter >= 0 ? newDiameter : 10 );
        repaint(); // pinta o painel novamente
    } // fim do método setDiameter
    // utilizado pelo gerenciador de layout para determinar o tamanho
preferido

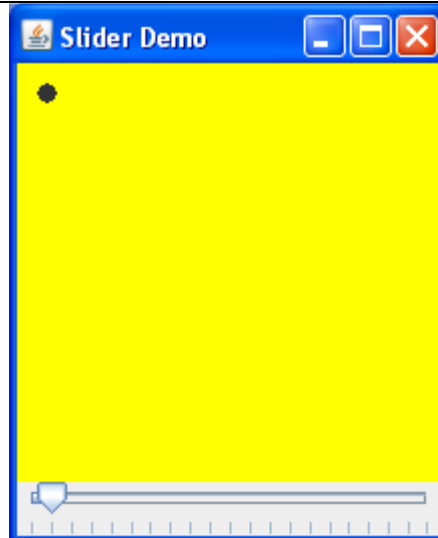
```

```

public Dimension getPreferredSize(){
    return new Dimension( 200, 200 );
} // fim do método getPreferredSize

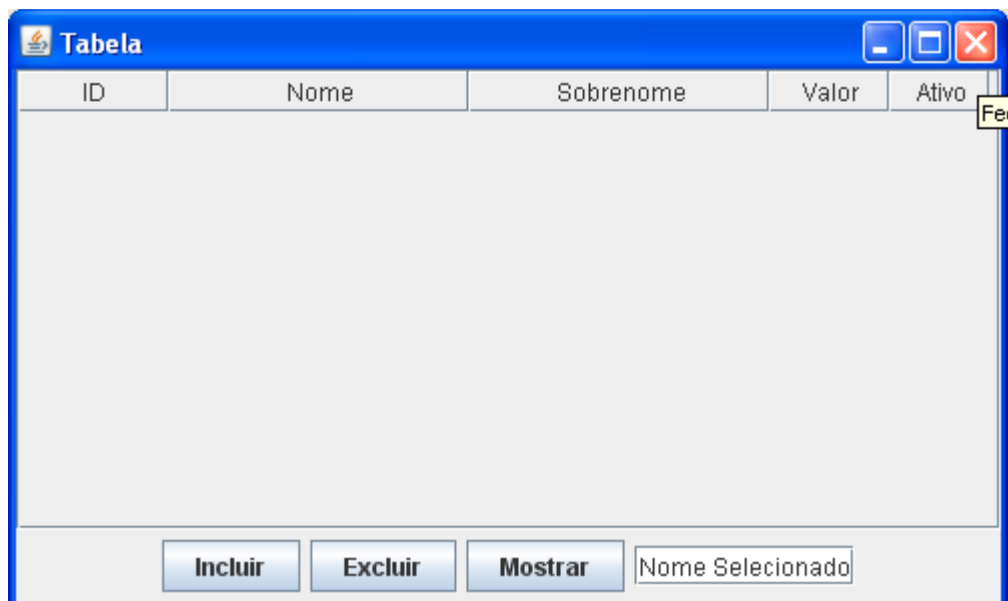
// utilizado pelo gerenciador de layout para determinar o tamanho
mínimo
public Dimension getMinimumSize(){
    return getPreferredSize();
} // fim do método getMinimumSize
} // fim da classe OvalPanel

```



EXERCÍCIOS PARA ENTREGA

EE_06) Implemente um programa em Java que resulte na seguinte janela e suas funcionalidades alterando a cor do fundo conforme a cor do botão clicado:



EE_07) Implemente um programa em Java que resulte na seguinte janela e suas funcionalidades:

