ComboBoxDemo.java

```
package aula.igrafica;

import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class ComboBoxDemo extends JFrame{

   private JComboBox imagensComboBox;
   private JLabel label;

   private String nomes[] = {"smile.gif", "cool.gif", "dead.gif", "oh.gif"};
   private Icon[] icones = {new ImageIcon(nomes[0]), new ImageIcon(nomes[1]), new
ImageIcon(nomes[2]), new ImageIcon(nomes[3])};
   // configura a GUI
   public ComboBoxDemo() {
      super("Testando o JComboBox");
      // configuração do layout
      Container container = getContentPane();
      container.setLayout(new FlowLayout());
      // configura a JComboBox e registra o tratamento de eventos
      imagensComboBox = new JComboBox(nomes);
      imagensComboBox.setMaximumRowCount(3);
      imagensComboBox.addItemListener(
         new ItemListener() { // classe interna para tratamento do evento de JComboBox
            public void itemStateChanged(ItemEvent event){
               if(event.getStateChange() == ItemEvent.SELECTED)
                  label.setIcon( icones[imagensComboBox.getSelectedIndex()] );
            }// fim da classe interna anônima
         }
       ); // fim da chamada addItemListener
       container.add( imagensComboBox );
       // configura o JLabel para apresentar o resultado
       label = new JLabel( icones[0] );
       container.add(label);

       setSize(350,100);
       setVisible(true);
   }
      public static void main(String args[]){
      ComboBoxDemo aplicacao = new ComboBoxDemo();
      aplicacao.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
   }
}
```

JListDemo.java

```
package aula.igrafica;

import java.awt.Color;
import java.awt.Container;
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.ListSelectionModel;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;

public class JListDemo extends JFrame{
   private JList colorList;
   private Container container;

   private String nomeCor[] = {"Preto", "Azul", "Ciano", "Cinza Escuro", "Cinza",
                        "Verde", "Cinza Claro", "Magenta", "Laranja", "Rosa",
                        "Vermelho", "Branco", "Amarelo"};
   private Color cores[] = {Color.black, Color.blue, Color.cyan, Color.darkGray, Color.gray,
                       Color.green, Color.lightGray, Color.magenta, Color.orange, Color.pink,
                       Color.red, Color.white, Color.yellow};
   // configura a GUI
   public JListDemo() {
      super("Teste da JList");
      // configura o layout
      container = getContentPane();
      container.setLayout(new FlowLayout());
      // cria uma lista com itens do array nomeCor
      colorList = new JList(nomeCor);
      colorList.setVisibleRowCount(5);
      // não permite seleção múltipla
      colorList.setSelectionMode( ListSelectionModel.SINGLE_SELECTION );
      container.add(new JScrollPane(colorList));
      // adiciona tratamento de evento ao JLits
      colorList.addListSelectionListener(
         new ListSelectionListener() {
            public void valueChanged(ListSelectionEvent e) {
               container.setBackground( cores[colorList.getSelectedIndex()] );
            }
         }// fim da classe interna anônima
      ); // fim da chamada addListSelectionListener
      setSize(350,150);
      setVisible(true);
   }
      public static void main(String args[]){
      JListDemo aplicacao = new JListDemo();
      aplicacao.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }
}
```

JListMultiDemo.java

```
package aula.igrafica;

import java.awt.Container;
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.JList;
import javax.swing.JScrollPane;
import javax.swing.ListSelectionModel;

public class JListMultiDemo extends JFrame {
   private JList colorList, copyList;
   private JButton copyButton;

   private String nomeCor[] = {"Preto", "Azul", "Ciano", "Cinza Escuro", "Cinza",
                       "Verde", "Cinza Claro", "Magenta", "Laranja", "Rosa",
                       "Vermelho", "Branco", "Amarelo"};
   // configura a GUI
   public JListMultiDemo() {
      super("Múltipla seleção na JList");
      // configura layout
      Container container = getContentPane();
      container.setLayout(new FlowLayout());
      // configura colorList
      colorList = new JList(nomeCor);
      colorList.setVisibleRowCount(5);
      colorList.setFixedCellHeight(15);
      colorList.setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);
      container.add(new JScrollPane(colorList));
      // cria botão de cópia e configura o ouvinte ("listener")
      copyButton = new JButton("Copiar >>");
      copyButton.addActionListener(
         new ActionListener() {// classe interna anônima para evento de botão
            public void actionPerformed(ActionEvent e) {
               // coloca os valores selecionados na copyList
               copyList.setListData(colorList.getSelectedValues());
            }
         }// fim da classe interna anônima
      );//fim da chamada de addActionListener
      container.add(copyButton);

      //configura a copyList
      copyList = new JList();
      copyList.setVisibleRowCount(5);
      copyList.setFixedCellWidth(100);
      copyList.setFixedCellHeight(15);
      copyList.setSelectionMode( ListSelectionModel.SINGLE_INTERVAL_SELECTION );
      container.add(new JScrollPane(copyList));
```

```
      setSize(400,150);
      setVisible(true);
   }

   public static void main(String args[]){
      JListMultiDemo aplicacao = new JListMultiDemo();
      aplicacao.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }
}
```

MouseTeste.java

```
package aula.igrafica;

import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

public class MouseTeste extends JFrame
      implements MouseListener, MouseMotionListener{

   private JLabel statusBar;
   // configura a GUI e registra os tratadores de eventos
   public MouseTeste() {
      super("Demonstrando os eventos do mouse");
      statusBar = new JLabel();
      Font font = new Font("Serif",Font.BOLD,36);
      statusBar.setFont(font);
      getContentPane().add(statusBar,BorderLayout.SOUTH);
      //programa espera seus próprios eventos do mouse
      addMouseListener(this);
      addMouseMotionListener(this);
      setSize(675,320);
      setLocation(100,100);
      setVisible(true);
   }

   //tratadores de eventos do MouseListener

   // trata evento quando o mouse é liberado imediatamente após ser pressionado
   public void mouseClicked( MouseEvent event ){
      statusBar.setText("Clicado em ["+event.getX()+", "+event.getY()+"]");
   }

   // trata evento quando o mouse é pressionado
```

```
   public void mousePressed( MouseEvent event ){
      statusBar.setText("Pressionado em ["+event.getX()+", "+event.getY()+"]");
   }

   // trata evento quando o mouse é liberado após ser arrastado
   public void mouseReleased( MouseEvent event ){
      statusBar.setText("Liberado em ["+event.getX()+", "+event.getY()+"]");
   }

   // trata evento quando o mouse entra na área da janela na tela
   public void mouseEntered( MouseEvent event ){
      //statusBar.setText("Entrou na tela em ["+event.getX()+", "+event.getY()+"]");
      JOptionPane.showMessageDialog(null,"Mouse na tela!");
   }

   // trata evento quando o mouse entra na área da janela na tela
   public void mouseExited( MouseEvent event ){
      statusBar.setText("Mouse fora da tela!");
   }

   //tratadores de eventos do MouseMotionListener

   // trata evento quando o usuário arrasta o mouse com o botão pressionado
   public void mouseDragged( MouseEvent event ){
      statusBar.setText("Arrastado em ["+event.getX()+", "+event.getY()+"]");
   }

   // trata evento quando o usuário movimenta o mouse
   public void mouseMoved( MouseEvent event ){
      statusBar.setText("Movimentado em ["+event.getX()+", "+event.getY()+"]");
   }

   public static void main(String args[]){
      MouseTeste aplicacao = new MouseTeste();
      aplicacao.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }

}
```