

Shembull 1

```
//Shembull qe tregon vendosjen e nje butoni ne nje JFrame

import java.awt.FlowLayout;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

public class ButtonFrame extends JFrame
{
    private JButton jButtonIThjeshte; // buton me tekst
    private JButton jButtonMeIkone; // buton me ikone

    public ButtonFrame()
    {
        super( "Testimi i butonave" );
        setLayout( new FlowLayout() ); // vendosja e JFrame

        jButtonIThjeshte = new JButton( "Buton i thjeshte" );
        add( jButtonIThjeshte );

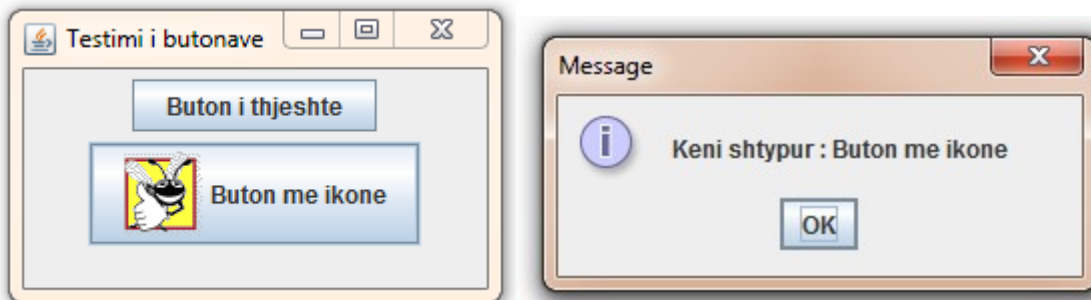
        Icon bug1 = new ImageIcon( getClass().getResource( "bug1.gif" ) );
        Icon bug2 = new ImageIcon( getClass().getResource( "bug2.gif" ) );
        jButtonMeIkone = new JButton( "Buton me ikone", bug1 );
        jButtonMeIkone.setRolloverIcon( bug2 );
        add( jButtonMeIkone );

        // krijimi i mbajtesit te butonit
        ButtonHandler mbajtes = new ButtonHandler();
        jButtonMeIkone.addActionListener( mbajtes );
        jButtonIThjeshte.addActionListener( mbajtes );
    }

    // klasa e brendshme per ngjarjet e butonit
    private class ButtonHandler implements ActionListener
    {
        public void actionPerformed( ActionEvent event )
        {
            JOptionPane.showMessageDialog( ButtonFrame.this, String.format(
                "Keni shtypur : %s", event.getActionCommand() ) );
        }
    }
}
```

```
//Testimi i frame
import javax.swing.JFrame;

public class ButtonTest
{
public static void main( String[] args )
{
    ButtonFrame buttonFrame = new ButtonFrame();
    buttonFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    buttonFrame.setSize( 275, 110 );
    buttonFrame.setVisible( true );
}
}
```



Shembull 2

```
//Shembull qe tregon vendosjen e checkbox ne nje JFrame
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;
import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.JCheckBox;

public class CheckBoxFrame extends JFrame
{
private JTextField textField;
private JCheckBox boldJCheckBox;
private JCheckBox italicJCheckBox;

// konstruktori i CheckBoxFrame qe shton JCheckBoxes ne JFrame
public CheckBoxFrame()
{
    super( "Testimi i JCheckBox" );
    setLayout( new FlowLayout() );

    textField = new JTextField( "Ndryshimi i stilit te shkrimit", 20 );
    textField.setFont( new Font( "Serif", Font.PLAIN, 14 ) );
    add( textField );

    boldJCheckBox = new JCheckBox( "I theksuar" );
```

```
italicJCheckBox = new JCheckBox( "I pjerret" );
add( boldJCheckBox );
add( italicJCheckBox );

// mbajtesit e JCheckBoxe
CheckBoxHandler mbajtes = new CheckBoxHandler();
boldJCheckBox.addItemListener( mbajtes );
italicJCheckBox.addItemListener( mbajtes );
}

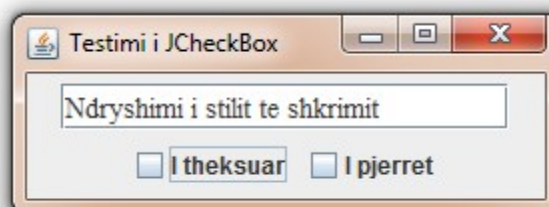
// klasa e brendshme private per ngjarjet e ItemListener
private class CheckBoxHandler implements ItemListener
{
    // pergjigja e eventeve te check box
    public void itemStateChanged( ItemEvent event )
    {
        Font font = null;

        if ( boldJCheckBox.isSelected() && italicJCheckBox.isSelected() )
            font = new Font( "Serif", Font.BOLD + Font.ITALIC, 14 );
        else if ( boldJCheckBox.isSelected() )
            font = new Font( "Serif", Font.BOLD, 14 );
        else if ( italicJCheckBox.isSelected() )
            font = new Font( "Serif", Font.ITALIC, 14 );
        else
            font = new Font( "Serif", Font.PLAIN, 14 );

        textField.setFont( font );
    }
}

//Testimi i CheckBoxFrame.
import javax.swing.JFrame;

public class CheckBoxTest
{
    public static void main( String[] args )
    {
        CheckBoxFrame checkBoxFrame = new CheckBoxFrame();
        checkBoxFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        checkBoxFrame.setSize( 275, 100 );
        checkBoxFrame.setVisible( true );
    }
}
```



Shembull 3

```
//Shembull me JComboBox qe afishon nje liste me emrat e imazheve
import java.awt.FlowLayout;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JComboBox;
import javax.swing.Icon;
import javax.swing.ImageIcon;

public class ComboBoxFrame extends JFrame
{
    private JComboBox imgJComboBox; // combobox qe ruan emrat e ikonave
    private JLabel etikete; // etikete qe tregon ikonën e zgjedhur

    private static final String[] emri =
        { "bug1.gif", "bug2.gif", "travelbug.gif", "buganim.gif" };
    private Icon[] icons = {
        new ImageIcon( getClass().getResource( emri[ 0 ] ) ),
        new ImageIcon( getClass().getResource( emri[ 1 ] ) ),
        new ImageIcon( getClass().getResource( emri[ 2 ] ) ),
        new ImageIcon( getClass().getResource( emri[ 3 ] ) ) };

    // konstruktori i ComboBoxFrame i cili shton JComboBox ne JFrame
    public ComboBoxFrame()
    {
        super( "Testimi i JComboBox" );
        setLayout( new FlowLayout() ); // caktimi i nderfaqes

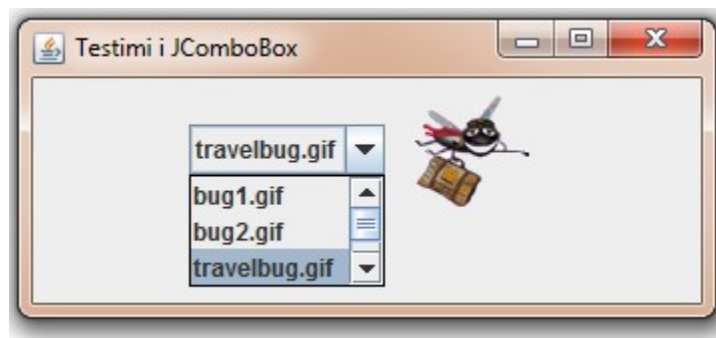
        imgJComboBox = new JComboBox( emri );
        imgJComboBox.setMaximumRowCount( 3 ); // afishohen tre rreshta ne combo

        imgJComboBox.addItemListener(
            new ItemListener() //klasa e brendshme anonime, pa emer
            {
                // mbajtesi i JComboBox
                public void itemStateChanged( ItemEvent event )
                {
                    // percaktimi kur zgjidhet nje element ne combo
                    if ( event.getStateChange() == ItemEvent.SELECTED )
                        etikete.setIcon( icons[
                            imgJComboBox.getSelectedIndex() ] );
                }
            }
        );

        add( imgJComboBox );
        etikete = new JLabel( icons[ 0 ] );
        add( etikete );
    }
}
```

```
//Testimi i JComboBoxFrame
import javax.swing.JFrame;

public class JComboBoxTest
{
public static void main( String[] args )
{
    JComboBoxFrame comboBoxFrame = new JComboBoxFrame();
    comboBoxFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    comboBoxFrame.setSize( 350, 150 );
    comboBoxFrame.setVisible( true );
}
}
```



Shembull 4

```
//Shembull qe tregon kopjimin e nje elementi liste tek nje tjetër
import java.awt.FlowLayout;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JButton;
import javax.swing.JScrollPane;
import javax.swing.ListSelectionModel;

public class MultipleSelectionFrame extends JFrame
{
private JList colorJList; // lista qe mban emrat e ngjyrave
private JList copyJList; // list ku do kopjohen emrat
private JButton copyJButton; // butoni qe kopjon emrat
private static final String[] ngjyra = { "E zeze", "Blu", "E kalter",
    "Gri e Erret", "Gri", "E gjelber", "Gri e celet", "Mavi", "Portokalli",
    "Roze", "E kuqe", "E bardhe", "E verdhe" };

// Konstruktori i MultipleSelectionFrame
public MultipleSelectionFrame()
{
    super( "Liste me disa zgjedhje" );
    setLayout( new FlowLayout() );

    colorJList = new JList( ngjyra );
```

```

colorJList.setVisibleRowCount( 5 );
colorJList.setSelectionMode(
    ListSelectionMode.MULTIPLE_INTERVAL_SELECTION );
add( new JScrollPane( colorJList ) );

copyJButton = new JButton( "Kopjo >>>" );
copyJButton.addActionListener(

    new ActionListener() // klasa e brendshme anonime
    {
        // mbajtja e butonit
        public void actionPerformed( ActionEvent event )
        {
            // shtimi i ngjyres ne copyJList
            copyJList.setListData( colorJList.getSelectedValuesList().toArray() );
        }
    }
);

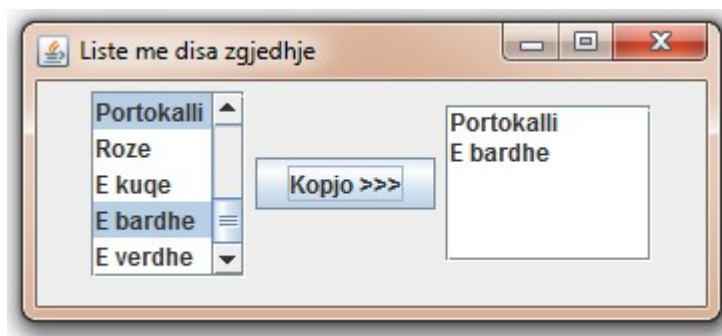
add( copyJButton );

copyJList = new JList();
copyJList.setVisibleRowCount( 5 );
copyJList.setFixedCellWidth( 100 );
copyJList.setFixedCellHeight( 15 );
copyJList.setSelectionMode(
    ListSelectionMode.SINGLE_INTERVAL_SELECTION );
add( new JScrollPane( copyJList ) );
}
}

//Testimi i MultipleSelectionFrame
import javax.swing.JFrame;

public class MultipleSelectionTest
{
    public static void main( String[] args )
    {
        MultipleSelectionFrame multipleSelectionFrame =
            new MultipleSelectionFrame();
        multipleSelectionFrame.setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE );
        multipleSelectionFrame.setSize( 350, 150 );
        multipleSelectionFrame.setVisible( true );
    }
}

```



Shembull 5

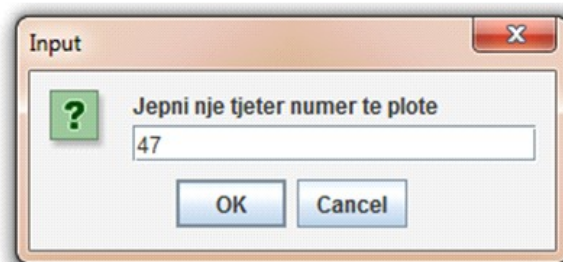
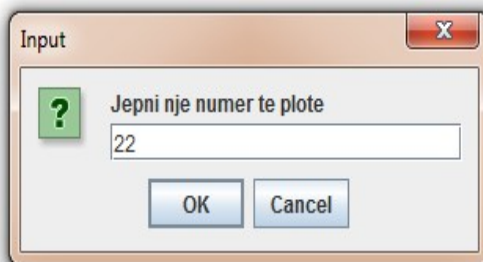
```
//Shembull qe tregon implementimin e JOptionPane per input/ output.
import javax.swing.JOptionPane;

public class MbledhjeNr
{
    public static void main( String[] args )
    {
        // lejon perdoruesin te jape dy numra te plote
        String numriIPare =
            JOptionPane.showInputDialog( "Jepni nje numer te plote" );
        String numriIDyte =
            JOptionPane.showInputDialog( "Jepni nje tjetere numer te plote" );

        // kthen inputin nga String inputs ne int
        int nr1 = Integer.parseInt( numriIPare );
        int nr2 = Integer.parseInt( numriIDyte );

        int shuma = nr1 + nr2;

        // afishohet rezultati ne nje dritare mesazhi JOptionPane
        JOptionPane.showMessageDialog( null, "Shuma e dy numrave te dhene eshte " +
            shuma,
            "Shuma e dy numrave", JOptionPane.PLAIN_MESSAGE );
    }
}
```



Shembull 6

```
//Shembull qe tregon perdorimin e ngjarjeve te tastjeres

import java.awt.Color;
import java.awt.event.KeyListener;
import java.awt.event.KeyEvent;
import javax.swing.JFrame;
import javax.swing.JTextArea;

public class KeyDemoFrame extends JFrame implements KeyListener
{
    private String rreshti1 = ""; // rreshti i pare ne hapsiren tekst
    private String rreshti2 = ""; // rreshti i dyte ne hapsiren tekst
    private String rreshti3 = ""; // rreshti i trete ne hapsiren tekst
    private JTextArea teksti; // hapsira tekst

    // kontruktori i KeyDemoFrame
    public KeyDemoFrame()
    {
        super( "Testimi i ngjarjeve te tastjeres" );

        teksti = new JTextArea( 10, 15 );
        teksti.setText( "Shtyp cilindo buton ne tastjere..." );
        teksti.setEnabled( false ); // fshehja e hapsires tekst
        teksti.setDisabledTextColor( Color.BLACK );
        add( teksti );

        addKeyListener( this ); // frame reagon ndaj ngjarjeve te tastjeres
    }

    // mbajtja e ngjarjes te shtypjes se butonit te tastjeres
    public void keyPressed( KeyEvent event )
    {
        rreshti1 = String.format( "Butoni i shtypur eshte : %s",
            KeyEvent.getKeyText( event.getKeyCode() ) ); // tregohet butoni i shtypur
        setLines2and3( event ); // caktohen rreshtat 2 dhe 3
    }

    // mbajtja e ngjarjes te leshimit te butonit te tastjeres
    public void keyReleased( KeyEvent event )
    {
        rreshti1 = String.format( "Butoni i leshuar eshte : %s",
            KeyEvent.getKeyText( event.getKeyCode() ) ); // tregohet butoni i leshuar
        setLines2and3( event );
    }

    // mbajtja e ngjarjes te shtypjes se nje butoni funksion te tastjeres
    public void keyTyped( KeyEvent event )
    {
        rreshti1 = String.format( "Butoni i shtypur eshte : %s", event.getKeyChar() );
        setLines2and3( event );
    }
}
```



```
// caktimi i rreshtave te output
private void setLines2and3( KeyEvent event )
{
    rreshti2 = String.format( "Ky buton %seshte nje buton funksioni",
        ( event.isActionKey() ? "" : "nuk " ) );

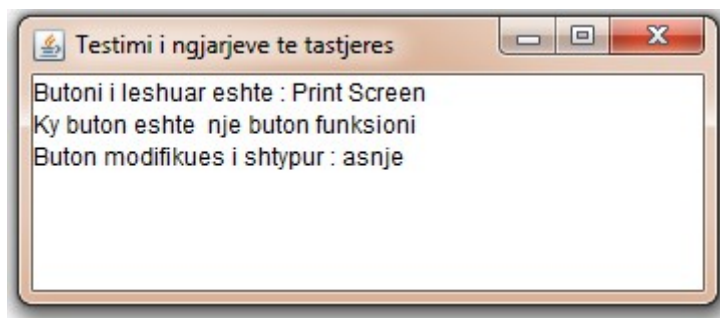
    String temp = KeyEvent.getKeyModifiersText( event.getModifiers() );

    rreshti3 = String.format( "Buton modifikues i shtypur : %s",
        ( temp.equals( "" ) ? "asnje" : temp ) );

    teksti.setText( String.format( "%s\n%s\n%s\n",
        rreshti1, rreshti2, rreshti3 ) );
}
}

//Testimi i KeyDemoFrame
import javax.swing.JFrame;

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



Shembull 7

```
//Shembull qe tregon perdorimin e etiketave

import java.awt.FlowLayout; // percakton si jane te vendosur komponentet grafike
import javax.swing.JFrame; // ofron cilesi te dritares
import javax.swing.JLabel; // etikete me tekst ose me ikone
import javax.swing.SwingConstants; // konstante ndihmese te Swing
import javax.swing.Icon; // nderfaqe qe perdoret per ikonat
import javax.swing.ImageIcon; // ngarkimi i ikonave

public class LabelFrame extends JFrame
{
    private JLabel etiketa1; // etikete vetem me tekst
    private JLabel etiketa2; // etikete me tekst dhe ikone
    private JLabel etiketa3; // etikete me tekst dhe ikone
}
```

```
// konstruktori i klases JLabelFrame
public JLabelFrame()
{
    super( "Testimi i JLabel" );
    setLayout( new FlowLayout() );

    // konstruktori i JLabel kerkon nje parameter tekst
    etiketa1 = new JLabel( "Etikete me tekst" );
    etiketa1.setToolTipText( "Kjo eshte etiketa1" );
    add( etiketa1 );

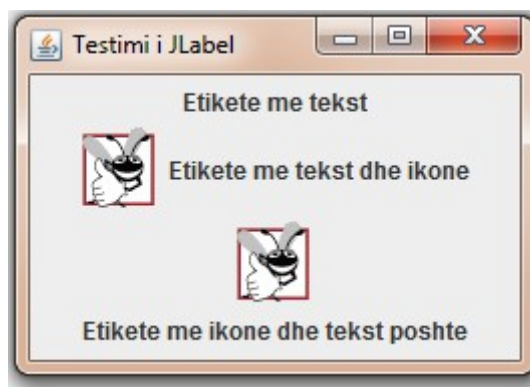
    // konstruktori i JLabel kerkon nje string, nje Icon dhe argument per
    pozicionimin
    Icon bug = new ImageIcon( getClass().getResource( "bug1.png" ) );
    etiketa2 = new JLabel( "Etikete me tekst dhe ikone", bug,
        SwingConstants.LEFT );
    etiketa2.setToolTipText( "Kjo eshte etiketa2" );
    add( etiketa2 );

    etiketa3 = new JLabel();
    etiketa3.setText( "Etikete me ikone dhe tekst poshte" );
    etiketa3.setIcon( bug );
    etiketa3.setHorizontalTextPosition( SwingConstants.CENTER );
    etiketa3.setVerticalTextPosition( SwingConstants.BOTTOM );
    etiketa3.setToolTipText( "This is label3" );
    add( etiketa3 );
}
}

//Testimi i etiketave

import javax.swing.JFrame;

public class LabelTest
{
    public static void main( String[] args )
    {
        JLabelFrame labelFrame = new JLabelFrame();
        labelFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        labelFrame.setSize( 260, 180 );
        labelFrame.setVisible( true );
    }
}
```



Shembull 8

//Shembull qe tregon perdorimin e nje JList qe permban emrat e ngjyrave.

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

public class ListFrame extends JFrame
{
    private JList colorJList; // lista qe permban ngjyrat
    private static final String[] emrat = { "E zeze", "Blu", "E kalter",
        "Gri e erret", "Gri", "E gjelber", "Gri e celet", "Mavi",
        "Portokalli", "Roze", "E kuqe", "E bardhe", "E verdhe" };
    private static final Color[] ngjyrat = { Color.BLACK, Color.BLUE,
        Color.CYAN, Color.DARK_GRAY, Color.GRAY, Color.GREEN,
        Color.LIGHT_GRAY, Color.MAGENTA, Color.ORANGE, Color.PINK,
        Color.RED, Color.WHITE, Color.YELLOW };

    // Konstruktori i ListFrame i cili shton JScrollPane qe permban listen
    public ListFrame()
    {
        super( "List Test" );
        setLayout( new FlowLayout() );

        colorJList = new JList( emrat );
        colorJList.setVisibleRowCount( 5 );

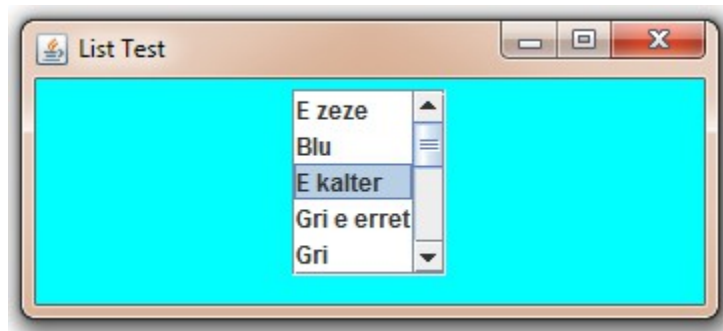
        // ky instruksion nuk lejon zgjedhjen e disa ngjyrave njeheresh
        colorJList.setSelectionMode( ListSelectionModel.SINGLE_SELECTION );

        add( new JScrollPane( colorJList ) );

        colorJList.addListSelectionListener(
            new ListSelectionListener() // klasa e brendshme anonime
            {
                // mbajtja e ngjarjeve te listes
                public void valueChanged( ListSelectionEvent event )
                {
                    getContentPane().setBackground(
                        ngjyrat[ colorJList.getSelectedIndex() ] );
                }
            }
        );
    }
}
```

```
//Testimi i listes se ngjyrave
import javax.swing.JFrame;

public class ListTest
{
public static void main( String[] args )
{
    ListFrame listFrame = new ListFrame();
    listFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    listFrame.setSize( 350, 150 );
    listFrame.setVisible( true );
}
}
```



Shembull 9

```
//Shembull qe tregon mbajtjet e ngjarjeve te mousit
import java.awt.Color;
import java.awt.BorderLayout;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;
import java.awt.event.MouseEvent;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class MouseTrackerFrame extends JFrame
{
private JPanel mousePanel; // panel ne te cilin ndodhin ngjarjet e mouse
private JLabel statusBar; // etikete qe afishon informacion mbi ngjarjet

// konstruktori i MouseTrackerFrame nderton nderfaqen grafike dhe regjistron
mbajtesit e ngjarjeve

public MouseTrackerFrame()
{
    super( "Shembull me ngjarjet e mouse" );

    mousePanel = new JPanel();
    mousePanel.setBackground( Color.WHITE );
    add( mousePanel, BorderLayout.CENTER );
}
```

```
statusBar = new JLabel( "Mouse ndodhet jashte JPanel" );
add( statusBar, BorderLayout.SOUTH );

// krijimi dhe regjistrimi i mbajtesve te ngjarjeve te mouse
MouseListener handler = new MouseHandler();
mousePanel.addMouseListener( handler );
mousePanel.addMouseMotionListener( handler );
}

private class MouseHandler implements MouseListener,
MouseListener
{
    // MouseListener si mbajtes te ngjarjeve
    // ngjarja e klikimit
    public void mouseClicked( MouseEvent event )
    {
        statusBar.setText( String.format( "Klikuar ne koordinatat [%d, %d]",
            event.getX(), event.getY() ) );
    }

    // ngjarja e shtypjes se butonit te majte
    public void mousePressed( MouseEvent event )
    {
        statusBar.setText( String.format( "Shtypur ne koordinatat [%d, %d]",
            event.getX(), event.getY() ) );
    }

    // ngjarja e lshimit te butonit
    public void mouseReleased( MouseEvent event )
    {
        statusBar.setText( String.format( "Leshuar ne koordinatat [%d, %d]",
            event.getX(), event.getY() ) );
    }

    // ngjarja e pozicionimit te mouse mbi panel
    public void mouseEntered( MouseEvent event )
    {
        statusBar.setText( String.format( "Mouse eshte ne koordinatat [%d, %d]",
            event.getX(), event.getY() ) );
        mousePanel.setBackground( Color.GREEN );
    }

    // ngjarja e daljes se mouse nga paneli
    public void mouseExited( MouseEvent event )
    {
        statusBar.setText( "Mouse jashte nga JPanel" );
        mousePanel.setBackground( Color.WHITE );
    }

    // ngjarja e levizjes se mouse me buton shtypur (zvarritje)
    public void mouseDragged( MouseEvent event )
    {
        statusBar.setText( String.format( "Zvarritje ne koordinatat [%d, %d]",
            event.getX(), event.getY() ) );
    }

    // ngjarja levizjes se mouse
    public void mouseMoved( MouseEvent event )
    {

```

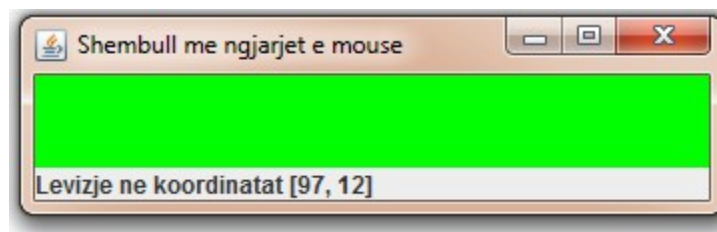
```

        statusBar.setText( String.format( "Levizje ne koordinatat [%d, %d]",
            event.getX(), event.getY() ) );
    }
}
}

//Testimi i MouseTrackerFrame.
import javax.swing.JFrame;

public class MouseTracker
{
    public static void main( String[] args )
    {
        MouseTrackerFrame mouseTrackerFrame = new MouseTrackerFrame();
        mouseTrackerFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        mouseTrackerFrame.setSize( 300, 100 );
        mouseTrackerFrame.setVisible( true );
    }
}

```



Shembull 10

```

//Shembull qe tregon perdorimin e ngjarjeve te mouse

import java.awt.BorderLayout;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class MouseDetailsFrame extends JFrame
{
    private String detaje; // String qe afishohet ne statusbar
    private JLabel statusBar; // JLabel qe afishohet ne fund te dritares

    // konstruktori qe cakton titullin dhe regjistron mbajtesin e ngjarjeve te mouse
    public MouseDetailsFrame()
    {
        super( "Klikime" );

        statusBar = new JLabel( "Kliko" );
        add( statusBar, BorderLayout.SOUTH );
        addMouseListener( new MouseClickListener() ); // regjistrohet mbajtesi i klik
    }

    // klasa e brendshme qe mban ngjarjet e mouse
    private class MouseClickListener extends MouseAdapter

```

```

{
    public void mouseClicked( MouseEvent event )
    {
        int xPos = event.getX();
        int yPos = event.getY();

        detaje = String.format( "Klikuar %d here", event.getClickCount() );

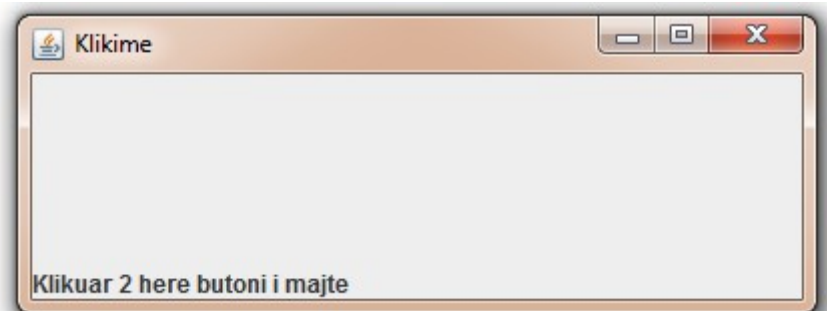
        if ( event.isMetaDown() ) // butoni i djathte
            detaje += " butoni i djathte";
        else if ( event.isAltDown() ) // butoni i mesit (rrotulla)
            detaje += " rrotulla";
        else // butoni i majte
            detaje += " butoni i majte";

        statusBar.setText( detaje ); // detajet afishohen ne status
    }
}

//Testimi i MouseDetailsFrame.
import javax.swing.JFrame;

public class MouseDetails
{
    public static void main( String[] args )
    {
        MouseDetailsFrame mouseDetailsFrame = new MouseDetailsFrame();
        mouseDetailsFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        mouseDetailsFrame.setSize( 400, 150 );
        mouseDetailsFrame.setVisible( true );
    }
}

```



Shembull 11

```

//Shembull qe tregon implementimin e nje paneli vizatimi
import java.awt.Point;
import java.awt.Graphics;
import java.awt.event.MouseEvent;
import java.awt.event.MouseMotionAdapter;

```

```
import javax.swing.JPanel;

public class PanelVizatimi extends JPanel
{
    private int nrPikash = 0; // numeron pikat ne panel

    // tabele me 10000 java.awt.Point
    private Point[] pikat = new Point[ 10000 ];

    // konstruktori i panelit dhe mbajtja e ngjarjeve te mouse
    public PanelVizatimi()
    {
        // mban ngjarjen e levizjes se mouse
        addMouseListener(

            new MouseMotionAdapter() // klasa e brendshme anonime
            {

                public void mouseDragged( MouseEvent event )
                {
                    if ( nrPikash < pikat.length )
                    {
                        pikat[ nrPikash ] = event.getPoint();
                        ++nrPikash;
                        repaint();
                    }
                }
            }
        );
    }

    public void paintComponent( Graphics g )
    {
        super.paintComponent( g );

        for ( int i = 0; i < nrPikash; i++ )
            g.fillOval( pikat[ i ].x, pikat[ i ].y, 4, 4 );
    }
}

//Testimi i panelit

import java.awt.BorderLayout;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class Painter
{
    public static void main( String[] args )
    {

        JFrame application = new JFrame( "Panel vizatimi" );

        PanelVizatimi panel = new PanelVizatimi();
        application.add( panel, BorderLayout.CENTER );

        application.add( new JLabel( "Zvarrit mouse per te vizatuar" ),
```



```

        BorderLayout.SOUTH );

    application.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    application.setSize( 400, 200 );
    application.setVisible( true );
}
}

```



Shembull 12

//Shembull qe tregon implementimin e nje paneli me butona

```

import java.awt.GridLayout;
import java.awt.BorderLayout;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JButton;

public class PanelFrame extends JFrame
{
    private JPanel buttonJPanel;
    private JButton[] buttonat;

    public PanelFrame()
    {
        super( "Panel Demo" );
        buttonat = new JButton[ 5 ];
        buttonJPanel = new JPanel();
        buttonJPanel.setLayout( new GridLayout( 1, buttonat.length ) );

        for ( int i = 0; i < buttonat.length; i++ )
        {
            buttonat[i] = new JButton( "Butoni " + ( i + 1 ) );
            buttonJPanel.add( buttonat[ i ] );
        }
    }
}

```

```
    add( buttonJPanel, BorderLayout.SOUTH );
}
}

//Testimi i PanelFrame

import javax.swing.JFrame;

public class PanelDemo extends JFrame
{
    public static void main( String[] args )
    {
        PanelFrame panelFrame = new PanelFrame();
        panelFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        panelFrame.setSize( 450, 200 );
        panelFrame.setVisible( true );
    }
}
```



Shembull 13

```
//Shembull qe tregon perdorimin e radio-butonave
```

```
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;
import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.JRadioButton;
import javax.swing.ButtonGroup;

public class RadioButtonFrame extends JFrame
{
    private JTextField textField;
    private Font iThjeshte;
    private Font iTheksuar;
    private Font iPjerret;
    private Font iTheksuarDhePjerret;
    private JRadioButton radioButtonITheshte;
```

```

private JRadioButton radioButtonITheksuar;
private JRadioButton radioButtonIPjerret;
private JRadioButton radioButtonITheksuarPjerret;
private ButtonGroup radioButtonGroup;
// konstruktori i RadioButtonFrame
public RadioButtonFrame()
{
    super( "RadioButton Test" );
    setLayout( new FlowLayout() );

    textField = new JTextField( "Vini re ndryshimin e fontit", 25 );
    add( textField );

    radioButtonITheshte = new JRadioButton( "I thjeshte", true );
    radioButtonITheksuar = new JRadioButton( "I theksuar", false );
    radioButtonIPjerret = new JRadioButton( "I pjerret", false );
    radioButtonITheksuarPjerret = new JRadioButton( "I theksuar/I pjerret", false );
    add( radioButtonITheshte );
    add( radioButtonITheksuar );
    add( radioButtonIPjerret );
    add( radioButtonITheksuarPjerret );

    // krijimi i lidhjeve logjike mes JRadioButtons
    radioButtonGroup = new ButtonGroup();
    radioButtonGroup.add( radioButtonITheshte );
    radioButtonGroup.add( radioButtonITheksuar );
    radioButtonGroup.add( radioButtonIPjerret );
    radioButtonGroup.add( radioButtonITheksuarPjerret );

    iPjerret = new Font( "Serif", Font.PLAIN, 14 );
    iTheksuar = new Font( "Serif", Font.BOLD, 14 );
    iPjerret = new Font( "Serif", Font.ITALIC, 14 );
    iTheksuarDhePjerret = new Font( "Serif", Font.BOLD + Font.ITALIC, 14 );
    textField.setFont( iPjerret );

    radioButtonITheshte.addItemListener(
        new RadioButtonHandler( iThjeshte ) );
    radioButtonITheksuar.addItemListener(
        new RadioButtonHandler( iTheksuar ) );
    radioButtonIPjerret.addItemListener(
        new RadioButtonHandler( iPjerret ) );
    radioButtonITheksuarPjerret.addItemListener(
        new RadioButtonHandler( iTheksuarDhePjerret ) );
}

// klase e brendshme private qe mban
private class RadioButtonHandler implements ItemListener
{
    private Font font;

    public RadioButtonHandler( Font f )
    {
        font = f;
    }

    public void itemStateChanged( ItemEvent event )
    {
        textField.setFont( font );
    }
}

```

```

}
}
//Klase testuese te RadioButtonFrame.
import javax.swing.JFrame;

public class RadioButtonTest
{
public static void main( String[] args )
{
    RadioButtonFrame radioButtonFrame = new RadioButtonFrame();
    radioButtonFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    radioButtonFrame.setSize( 300, 100 );
    radioButtonFrame.setVisible( true );
}
}

```



Shembull 14

```

//Shembull qe tregon kopjimin e tekstit nga njeri tekstarea tek tjetri
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.Box;
import javax.swing.JFrame;
import javax.swing.JTextArea;
import javax.swing.JButton;
import javax.swing.JScrollPane;

public class TextAreaFrame extends JFrame
{
private JTextArea textArea1; // demo string
private JTextArea textArea2; // stringa kopjohet ketu
private JButton copyJButton; // inicializon kopjimin e string

public TextAreaFrame()
{
    super( "TextArea Demo" );
    Box box = Box.createHorizontalBox();
    String demo = "Kjo eshte nje string demo\n" + "tregon kopjimin e tekstit\nnga
nje tekstarea tek \n" + "nje tjetere tekstarea\n\n";

    textArea1 = new JTextArea( demo, 10, 15 );
    box.add( new JScrollPane( textArea1 ) );

    copyJButton = new JButton( "Kopjo >>>" );
    box.add( copyJButton );
    copyJButton.addActionListener(

```

```

new ActionListener()
{
    public void actionPerformed((ActionEvent event) )
    {
        textArea2.setText( textArea1.getSelectedText() );
    }
}
);
textArea2 = new JTextArea( 10, 15 );
textArea2.setEditable( false );
box.add( new JScrollPane( textArea2 ) );
add( box );
}
}

//Klase testuese
import javax.swing.JFrame;

public class TextAreaDemo
{
    public static void main( String[] args )
    {
        TextAreaFrame textAreaFrame = new TextAreaFrame();
        textAreaFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        textAreaFrame.setSize( 425, 200 );
        textAreaFrame.setVisible( true );
    }
}

```



Shembull 15

```

//Shembull qe tregon implementimin e textfield
import java.awt.FlowLayout;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.JOptionPane;

```

```
public class TextFieldFrame extends JFrame
{
private JTextField textField1; // textfield per madhesine
private JTextField textField2; // textfield me tekst te gatshme te paracaktuar
private JTextField textField3; // textfield me madhesi dhe tekst
private JPasswordField passwordField; // passwordfield

public TextFieldFrame()
{
    super( "Testimi i JTextField dhe JPasswordField" );
    setLayout( new FlowLayout() );

    // textfield me 10 shkronja
    textField1 = new JTextField( 10 );
    add( textField1 );
    // textfield me tekst te paracaktuar
    textField2 = new JTextField( "Shkruani tekstin ketu" );
    add( textField2 );
    // textfield me 21 shkronja dhe me tekst te paracaktuar
    textField3 = new JTextField( "Tekst i paeditueshem", 21 );
    textField3.setEditable( false ); // i paeditueshem
    add( textField3 );
    passwordField = new JPasswordField( "FjalekalimiSekret" );
    add( passwordField );

    TextFieldHandler handler = new TextFieldHandler();
    textField1.addActionListener( handler );
    textField2.addActionListener( handler );
    textField3.addActionListener( handler );
    passwordField.addActionListener( handler );
}

private class TextFieldHandler implements ActionListener
{
    public void actionPerformed( ActionEvent event )
    {
        String string = ""; // stringa qe paraqitet

        if ( event.getSource() == textField1 )
            string = String.format( "textField1: %s",
                event.getActionCommand() );

        else if ( event.getSource() == textField2 )
            string = String.format( "textField2: %s",
                event.getActionCommand() );

        else if ( event.getSource() == textField3 )
            string = String.format( "textField3: %s",
                event.getActionCommand() );

        else if ( event.getSource() == passwordField )
            string = String.format( "passwordField: %s",
                event.getActionCommand() );

        JOptionPane.showMessageDialog( null, string );
    }
}
}
```

