Graphical User Interfaces is not a topic for examination

Suggested Reading:
Bruce Eckel, Thinking in Java (Fourth Edition)
Graphical User Interfaces


```
import javax.swing.*;

public class HelloSwing {
    public static void main(String[] args){
        JFrame frame = new JFrame("Hello Swing");
        frame.setVisible(true);
        frame.setSize(200, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

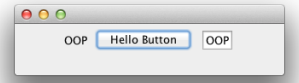
import javax.swing.*;

```
import javax.swing.*;

public class HelloLabel {
    public static void main(String[] args){
        JFrame frame = new JFrame("Hello Swing");

        JLabel label = new JLabel("Hello Label");
        frame.add(label);

        frame.setVisible(true);
        frame.setSize(200, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```


```
import java.awt.FlowLayout;

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

public class HelloButton {
    public static void main(String[] args){
        JFrame frame = new JFrame("Hello Swing");
        //left to right and top to bottom
        //frame.setLayout(new FlowLayout());

        JLabel label = new JLabel("Hello Label");
        frame.add(label);

        JButton button = new JButton("Hello Button");
        frame.add(button);

        frame.setVisible(true);
        frame.setSize(200, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

```
public class HelloButton2 extends JFrame {
    private JLabel label = new JLabel("Hello Label");
    private JButton button = new JButton("Hello Button");

    public HelloButton2() {
        setLayout(new FlowLayout());
        add(label);
        add(button);

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

        button.addActionListener(
                new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        String name = "MyButton was pressed";
                        label.setText(name);
                    }
                });
    }
}
```


```
public class HelloButton3 extends JFrame {
    private JLabel label = new JLabel("Hello Label");
    private JButton button = new JButton("Hello Button");
    private JTextField text = new JTextField("Text");

    public HelloButton3() {
        setLayout(new FlowLayout());
        add(label);
        add(button);
        add(text);
        setVisible(true);
        setSize(200, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        button.addActionListener(
                new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        String name = "MyButton was pressed";
                        label.setText(text.getText());
                    }
                });
    }
}
```



ActionListener

public void actionPerformed(ActionEvent e)

ActionEvent indicates
that a component-defined action occurred.

This high-level event is generated by a component (JButton) when the component-specific action occurs (such as being pressed).

The event is passed to every ActionListener object that registered to receive such events using the component's addActionListener method. (java 1.7 API)

ComponentListener

ComponentAdapter

public void

componentHidden(ComponentEvent e)

componentMoved(ComponentEvent e)

componentResized(ComponentEvent e)

componentShown(ComponentEvent e)

ComponentEvent

A low-level event which indicates that a component moved, changed size, or changed visibility (also, the root class for the other component-level events). (java 1.7 API)

ContainerListener

ContainerAdapter

public void

componentAdded(ContainerEvent e)

componentRemoved(ContainerEvent e)

ContainerEvent

A low-level event which indicates that a container's contents changed because a component was added or removed.(java 1.7 API)

FocusListener

FocusAdapter

public void

focusGained(FocusEvent e)

focusLost(FocusEvent e)

FocusEvent

A low-level event which indicates that a Component has gained or lost the input focus.

This low-level event is generated by a Component (such as a TextField).

The event is passed to every FocusListener or FocusAdapter object which registered to receive such events using the Component's addFocusListener method.(java 1.7 API)

KeyListener

KeyAdapter

public void

keyPressed(KeyEvent e)

keyReleased(KeyEvent e)

keyTyped(KeyEvent e)

KeyEvent

This low-level event is generated by a component object (such as a text field) when a key is pressed, released, or typed.

The event is passed to every KeyListener or KeyAdapter object which registered to receive such events using the component's addKeyListener method. (KeyAdapter objects implement the KeyListener interface.)

Each such listener object gets this KeyEvent when the event occurs.(java 1.7 API)

MouseListener

MouseAdapter

public void

mouseClicked(MouseEvent e)

mouseEntered(MouseEvent e)

mouseExited(MouseEvent e)

mousePressed(MouseEvent e)

mouseReleased(MouseEvent e)

MouseMotionListener

MouseMotionAdapter

public void

mouseDragged(MouseEvent e)

mouseMoved(MouseEvent e)

WindowListener

WindowAdapter

public void

windowOpened(WindowEvent e)

windowClosing(WindowEvent e)

windowClosed(WindowEvent e)

windowActivated(WindowEvent e)

windowDeactivated(WindowEvent e)

windowIconified(WindowEvent e)

windowDeiconified(WindowEvent e)

WindowEvent

A low-level event that indicates that a window has changed its status.

This low-level event is generated by a Window object when it is opened, closed, activated, deactivated, iconified, or deiconified, or when focus is transferred into or out of the Window.(java 1.7 API)

ItemListener

public void

itemStateChanged(ItemEvent e)

ItemEvent

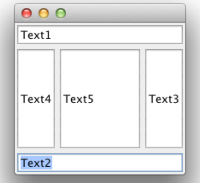
A semantic event which indicates that an item was selected or deselected.

This high-level event is generated by an ItemSelectable object (such as a List) when an item is selected or deselected by the user.

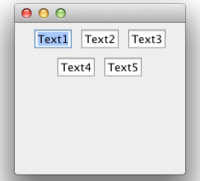
The event is passed to every ItemListener object which registered to receive such events using the component's addItemListener method.(java 1.7 API)


```
public class MyBorderLayout1 extends JFrame {
    private JTextField text1 = new JTextField("Text1");
    private JTextField text2 = new JTextField("Text2");
    private JTextField text3 = new JTextField("Text3");
    private JTextField text4 = new JTextField("Text4");
    private JTextField text5 = new JTextField("Text5");

    public MyBorderLayout1() {
        setLayout(new BorderLayout());
        add(text1);
        add(text2);
        add(text3);
        add(text4);
        add(text5);

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

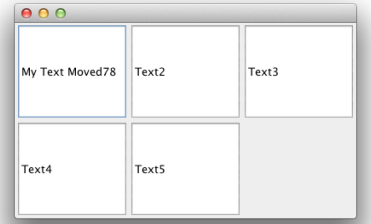

```
public class MyBorderLayout2 extends JFrame {
    private JTextField text1 = new JTextField("Text1");
    private JTextField text2 = new JTextField("Text2");
    private JTextField text3 = new JTextField("Text3");
    private JTextField text4 = new JTextField("Text4");
    private JTextField text5 = new JTextField("Text5");

    public MyBorderLayout2() {
        setLayout(new BorderLayout());
        add(BorderLayout.NORTH, text1);
        add(BorderLayout.SOUTH, text2);
        add(BorderLayout.EAST, text3);
        add(BorderLayout.WEST, text4);
        add(text5);

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



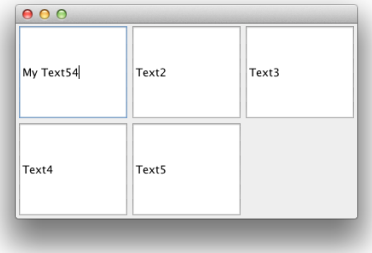

```
public class MyFlowLayout1 extends JFrame {
    private JTextField text1 = new JTextField("Text1");
    private JTextField text2 = new JTextField("Text2");
    private JTextField text3 = new JTextField("Text3");
    private JTextField text4 = new JTextField("Text4");
    private JTextField text5 = new JTextField("Text5");

    public MyFlowLayout1() {
        setLayout(new FlowLayout());
        add(text1);
        add(text2);
        add(text3);
        add(text4);
        add(text5);

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



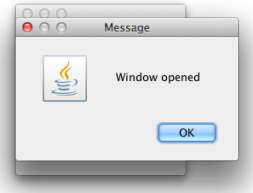
```
public class MyGridLayout1 extends JFrame {
    private JTextField text1 = new JTextField("Text1");
    private JTextField text2 = new JTextField("Text2");
    private JTextField text3 = new JTextField("Text3");
    private JTextField text4 = new JTextField("Text4");
    private JTextField text5 = new JTextField("Text5");

    public MyGridLayout1() {
        setLayout(new GridLayout(2,2));
        add(text1);
        add(text2);
        add(text3);
        add(text4);
        add(text5);

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

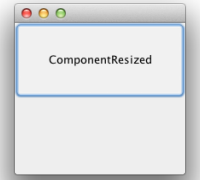


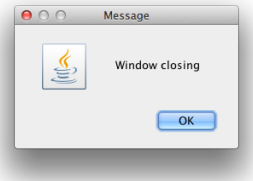

```
public class MyFrameListener1 extends JFrame {
    private JTextField text1 = new JTextField("Text1");
    private JTextField text2 = new JTextField("Text2");
    private JTextField text3 = new JTextField("Text3");
    private JTextField text4 = new JTextField("Text4");
    private JTextField text5 = new JTextField("Text5");

    public MyFrameListener1() {
        setLayout(new GridLayout(2,2));
        add(text1);
        add(text2);
        add(text3);
        add(text4);
        add(text5);

        setVisible(true);
        setSize(200, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

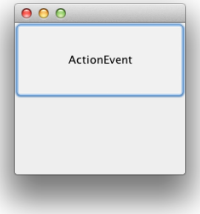




```
        addMouseListener(
            new MouseAdapter() {
                public void mouseClicked(MouseEvent e) {
                    Random r = new Random();
                    text1.setText("My Text" + r.nextInt(101));
                }
            });

        addMouseMotionListener(
                new MouseAdapter() {
                    public void mouseMoved(MouseEvent e) {
                        Random r = new Random();
                        text1.setText("My Text Moved" + r.nextInt(101));
                    }
                });
    }
}
```







```
public class MyButtonListener1 extends JFrame {
    private JButton myButton = new JButton("MyButton");
    private JFrame crtFrame;

    public MyButtonListener1() {
        crtFrame = this;
        setWindowEvents();

        add(myButton);
        setLayout(new GridLayout(2,2));

        setButtonEvents();

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

    }
    ...
}
```



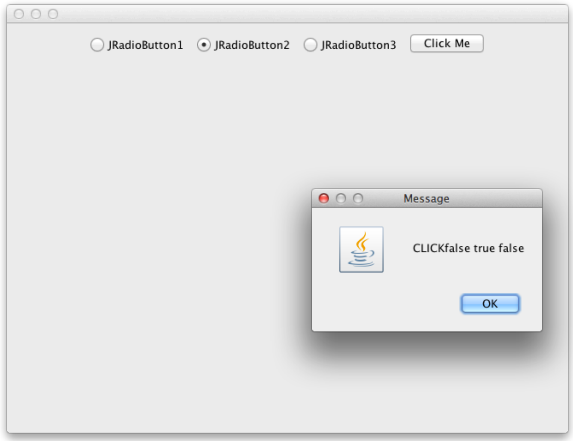
```
private void setButtonEvents() {
    myButton.addActionListener(
        new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                myButton.setText("ActionEvent");
            }
        });

    myButton.addComponentListener(
            new ComponentAdapter() {
                public void componentResized(ComponentEvent e) {
                    myButton.setText("ComponentResized");
                }
            });
}
```



```
    private void setWindowEvents() {
        addWindowListener(
            new WindowAdapter() {
                public void windowOpened(WindowEvent e) {
                    JOptionPane.showMessageDialog(crtFrame, "Window opened");

                }

                public void windowClosing(WindowEvent e) {
                    JOptionPane.showMessageDialog(null, "Window closing");
                }

                public void windowClosed(WindowEvent e) {
                    //only id dispose is called()
                    JOptionPane.showMessageDialog(null, "Window closed");
                }
            });

    }
```



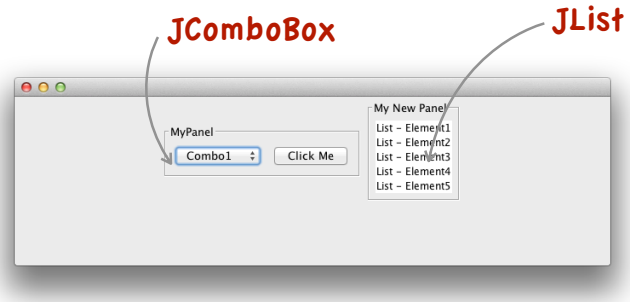
```
public class MyButtons1 extends JFrame {
    private BasicArrowButton up, down, right, left;

    public MyButtons1() {
        setComponents();
        setLayout(new FlowLayout());
        setVisible(true);
        setSize(200, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public void setComponents() {
        up = new BasicArrowButton(BasicArrowButton.NORTH); add(up);
        down = new BasicArrowButton(BasicArrowButton.SOUTH); add(down);
        right = new BasicArrowButton(BasicArrowButton.EAST); add(right);
        left = new BasicArrowButton(BasicArrowButton.WEST); add(left);

        add(new JCheckBox("JCheckBox"));
        add(new JRadioButton("JRadioButton1"));
        add(new JRadioButton("JRadioButton2"));
        add(new JRadioButton("JRadioButton3"));
    }
}
```

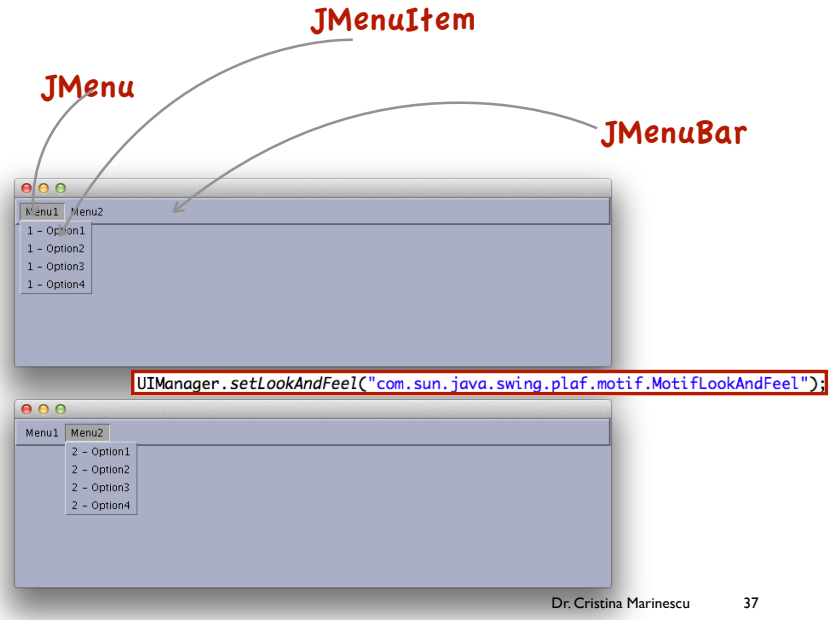


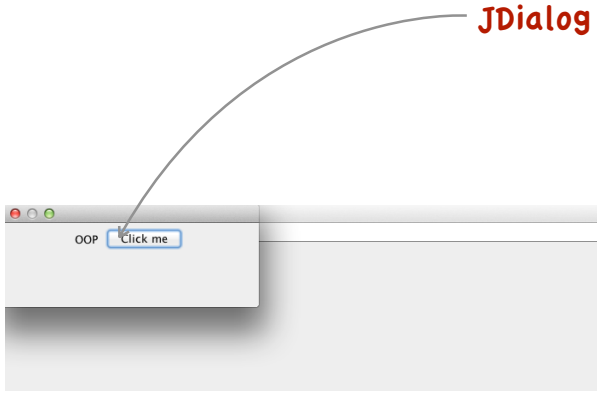

```
public void setComponents() {
    ButtonGroup buttonGroup = new ButtonGroup();
    JPanel panel = new JPanel();
    JButton button = new JButton("Click Me");
    button.setToolTipText("Get the selected radio button");

    final JRadioButton r1 = new JRadioButton("JRadioButton1");
    final JRadioButton r2 = new JRadioButton("JRadioButton2");
    final JRadioButton r3 = new JRadioButton("JRadioButton3");

    buttonGroup.add(r1);
    buttonGroup.add(r2);
    buttonGroup.add(r3);

    panel.add(r1);
    panel.add(r2);
    panel.add(r3);
    panel.add(button);

    add(panel);

  private void showMessage(String message, JRadioButton r1,
                                           JRadioButton r2,
                                           JRadioButton r3) {
     JOptionPane.showMessageDialog(null, message + r1.isSelected() + " " +
                              r2.isSelected() + " " + r3.isSelected());
  }
```

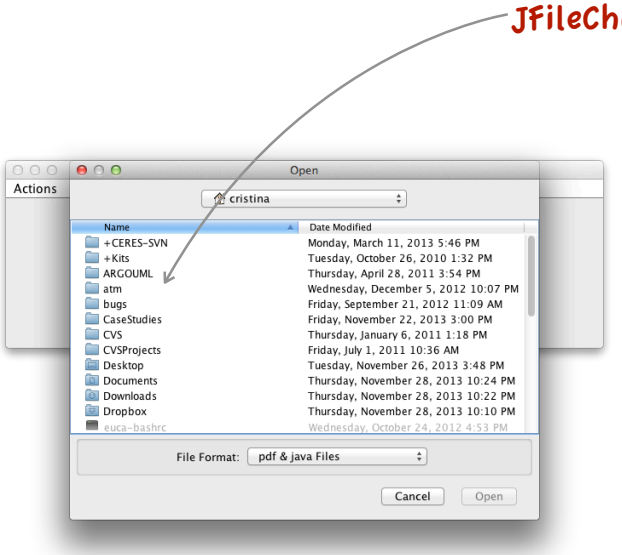


```
                        r1.addActionListener(
                                new ActionListener() {
                                    public void actionPerformed(ActionEvent e) {
                                        showMessage("Radio1", r1, r2, r3);
                                    }
                                });

                        r2.addActionListener(
                                new ActionListener() {
                                    public void actionPerformed(ActionEvent e) {
                                        showMessage("Radio2", r1, r2, r3);
                                    }
                                });

                        r3.addActionListener(
                                new ActionListener() {
                                    public void actionPerformed(ActionEvent e) {
                                        showMessage("Radio3", r1, r2, r3);
                                    }
                                });
    button.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    showMessage("CLICK", r1, r2, r3);
                }
            });

}
```







fore more details please consult

http://docs.oracle.com/javase/7/docs/api/