Object Oriented Programming with Java 18IS34

Unit-5

Swing Fundamentals

- Swing is a collection of classes and interfaces that offer rich set of visual components such as push butons, textfields ,scrollbars,checkboxes, trees, tables and menus.
- Components and containers
- A Component is an independent visual control such as push button.
- A Swing defines two types of containers: Top level containers JFrame, JApplet, JWindow,JDialog.
- Second type of container is lightweight container.

Top level container panes

- These define set of panes.
- JRootPane: glasspanes, content pane and layered pane.
- Layout Managers:

FlowLayout

BorderLayout

GridLayout

GridBagLayout

First Simple Swing Program

```
import javax.swing.*;
class swingDemo {
        swingDemo() {
        JFrame jfrm= new JFrame("A simple swing Application");
        jfrm.setSize(275,100);
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel("Swing defines the modern java GUI");
        jfrm.add(jlab);
        jfrm.setVisible(true);
}
public static void main(String [] args) {
        SwingUtilities.invokeLater(new Runnable() {
        public void run() {
                new swingDemo();
                }
        });

    }
}
```

Event Handling

- JLabel does not take input from the user and it does not generate events.
- An event is an object that describes a state change in a source.
- Event source is an object that generates an event.
- A listener is an object that is notified when an event occurs
- Listener has two requirements :

  It must have registered with one or more sources

  It must implement a method to recieve and process that event.

Adapter classes

- Java offers a set of adapter classes that provide an empty implementation of event listener interface methods.
- Some are ActionEvent, FocusEvent, KeyEvent, MouseEvent, WindowEvent

Using a push button

- push button is an instance of JButton
- Jbutton supplies several contructors

```
import java.awt.*;
import java.awt.event.*
import java.swing.*;
class ButtonDemo implements ActionListener {
    JLabel jlab;
    ButtonDemo(){
    JFrame jfrm = new JFrame(“A Button Example”);
    jfrm.setLayout(new FLowLayout());
    jfrm.setSize(220,90);
    jfrm.setDefaultCloseOperation(JFrame,EXIT_ON_CLOSE);
    JButton jbtnFirst = new JButton(“First”);
    JButton jbtSecond = new JButton(“Second”);
```

```
jbtnFirst.addActionListener(this);
jbtnSecond.addActionListener(this);
jfrm.add(jbtnFirst):
jfrm.add(jbtnSecond);
jlab = new JLabel(“Press a button”);
jfrm.SetVisible(true);
public void actionperformed(ActionEvent ae) {
    if(ae.getActionCommand().equals(“First”))
        jlab.setText(“First button was pressed”):
    else
        jlab.setText(“Second button was pressed”);
    }
```

```
public static void main(String [] args) {
    Swingutilities.invokeLater(new Runnable() {
        public void run() {
            new ButtonDemo();
            }
        });
    }
}
```

Jlabel and Imageicon

```
import java.swing.*;
import java.awt.*;
class JLabelDemo {
    JLabelDemo() {
    JFrame jfrm =new JFrame(“JLabel and ImageIcon Example”);
    jrm.setSize(320, 280);
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    ImageIcon goIcon = new ImageIcon(Go.gif”);
    JLabel jLabGo = new JLabel(“Go”, goIcon, Swingconstants.LEFT);
    ImageIcon cautionIcon = new ImageIcon(Caution.gif”);
    JLabel jLabCaution = new JLabel(“Caution”, cautionIcon, Swingconstants.CENTER);
    ImageIcon stopIcon = new ImageIcon(Stop.gif”);
    JLabel jlabStop = new JLabel(“Stop”, stopIcon, Swingconstants.RIGHT);
```

```
jfrm.add(jlabGo, BorderLayout.NORTH);
jfrm.add(jlabCaution, BorderLayout.CENTER);
jfrm.add(jlablabStop, BorderLayout.SOUTH);
jfrm.setVisible(true);
}
public static void main(String [] args) {
    swingUtilities.invokeLater(new Runnable() {
        public void run() {
            new JLabelDemo();
        }
    });
  }
}
```

Swing Buttons, Trees

- JButton A standard push button
- JToggleButton A two state (on/off) button
- JCHeckBox A standard check box
- JRadioButton A mutually exclusive check box

JTable