WEEK 13

AIM: To develop Swing application which uses JTabbedPane and JScrollPane

ALGORITHM: JTabbedPane

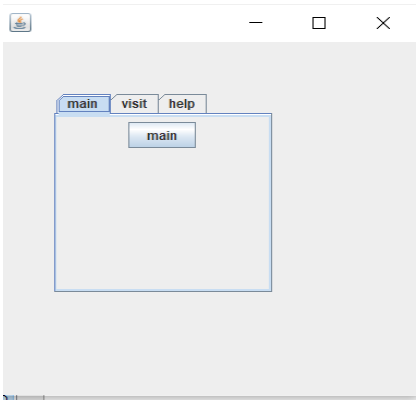
STEP1: START
STEP2: Create a TextArea with required bounds and add it to a panel.
STEP3: Create other two panels with some information to distinguish from one another.
STEP4: Create a JTabbedPane with required bounds and add the three panels to JTabbedPane object.
STEP5: Create a JFrame and add JTabbedPane to the JFrame.
STEP6: Set the JFrame size and set setVisible(true) to make JTabbedPane visible.
STEP7: END

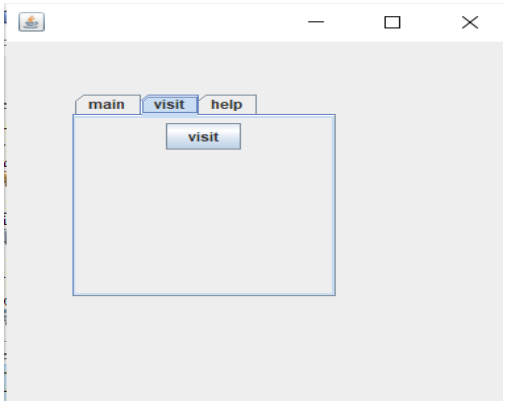
SOURCE CODE: JTabbedPane

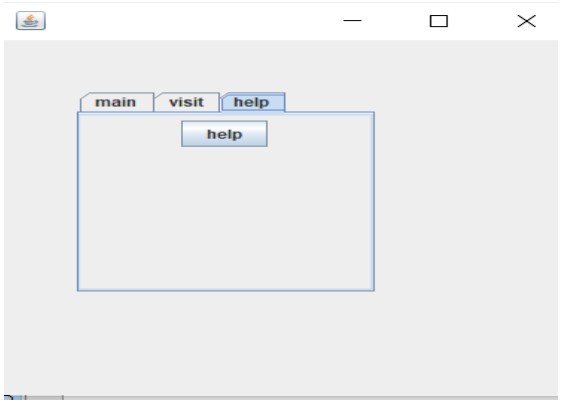
```
import javax.swing.*;

 public class TabbedPaneExample {

   JFrame f;

   TabbedPaneExample(){

      f=new JFrame();

      JPanel p1=new JPanel();

      JPanel p2=new JPanel();

      JPanel p3=new JPanel();

      JButton b1=new JButton("main");

      JButton b2=new JButton("visit");

      JButton b3=new JButton("help");

        p1.add(b1);

        p2.add(b2);

        p3.add(b3);

      JTabbedPane tp=new JTabbedPane();

      tp.setBounds(50,50,200,200);

      tp.add("main",p1);

      tp.add("visit",p2);

      tp.add("help",p3);

      f.add(tp);
```

```
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
  public static void main(String[] args) {
    new TabbedPaneExample();
  } }
```

OUTPUT: JTabbedPane







ALGORITHM: JScrollPane

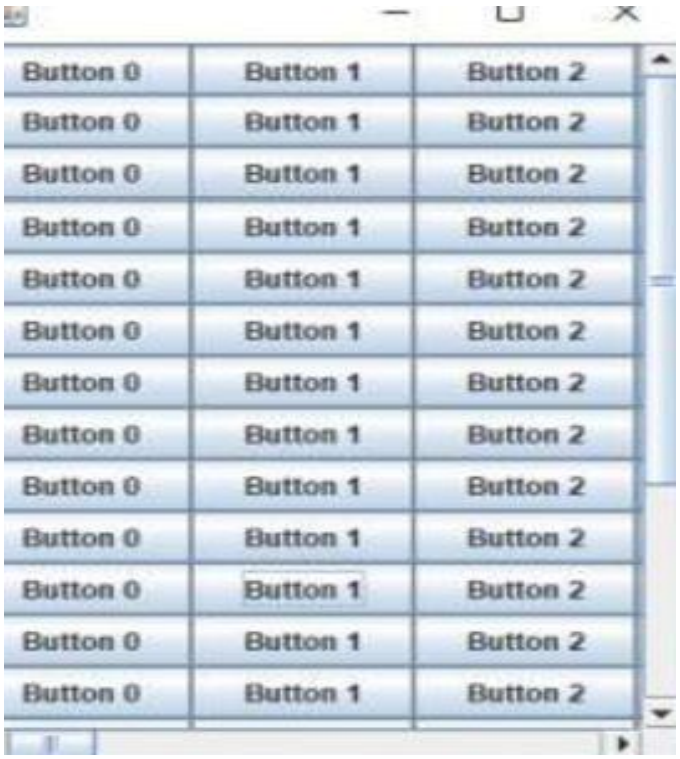
STEP1: START
STEP2: Create a Panel and set panel Layout to the GridLayout.
STEP3: Create array of buttons and add them to the Panel.
STEP4: Create horizontal and vertical ScrollPaneConstants.
STEP5: Create a JScrollPane and set panel, horizontal and vertical ScrollPaneConstants.
STEP6: Get the contentPane object and add the JScrollPane object to it with required layout.
STEP7: END

SOURCE CODE: JScrollPane

```
import java.awt.*;
  import javax.swing.*;
  public class ScrollEx
 {
    JFrame f;
    ScrollEx()
   {
    f=new JFrame();
    JPanel jp=new JPanel() ;
    jp.setLayout(new GridLayout(20,20)) ;
    for(int i=0;i<20;i++)
    for(int j=0;j<20;j++)
    {
     jp.add(new JButton("Button "+j));
    }
    JScrollPane js=new JScrollPane(jp);
    f.add(js);
    f.setSize(300,400);
    f.setVisible(true);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
```

```
    public static void main(String[] args)
    {
     new ScrollEx();
    }
}
```

OUTPUT: JScrollPane



VIVA VOCE:

1. What is JFC?

JFC is short for Java Foundation Classes, which encompass a group of features for building graphical user interfaces (GUIs) and adding rich graphics functionality and interactivity to Java applications. It is defined as containing the features shown in the table below. Feature. Description. Swing GUI Components.

2. What is Light weight component?

A lightweight component has no native screen resource of its own, so it is "lighter." A lightweight component relies on the screen resource from an ancestor in the containment hierarchy, possibly the underlying Frame object. Components from the javax.

3. What is the purpose of JTabbedPane class?

The JTabbedPane class is used to switch between a group of components by clicking on a tab with a given title or icon. It inherits JComponent class.

4. What is the purpose of JScrollPane class?

A JscrollPane is used to make scrollable view of a component. When screen size is limited, we use a scroll pane to display a large component or a component whose size can change dynamically.

5. What are the differences between a JScrollBar and a JScrollPane in Java?

A JScrollBar is a component and it doesn't handle its own events whereas a JScrollPane is a Container and it handles its own events and performs its own scrolling.