WEEK 7

AIM: To write a Java program to demonstrate MouseListener, MouseMotionListener and KeyListener.

THEORY: A listener is an object that is notified when an event occurs. It has two major requirements.First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications. The methods that receive and process events are defined in a set of interfaces found in java.awt.event.

ALGORITHM:

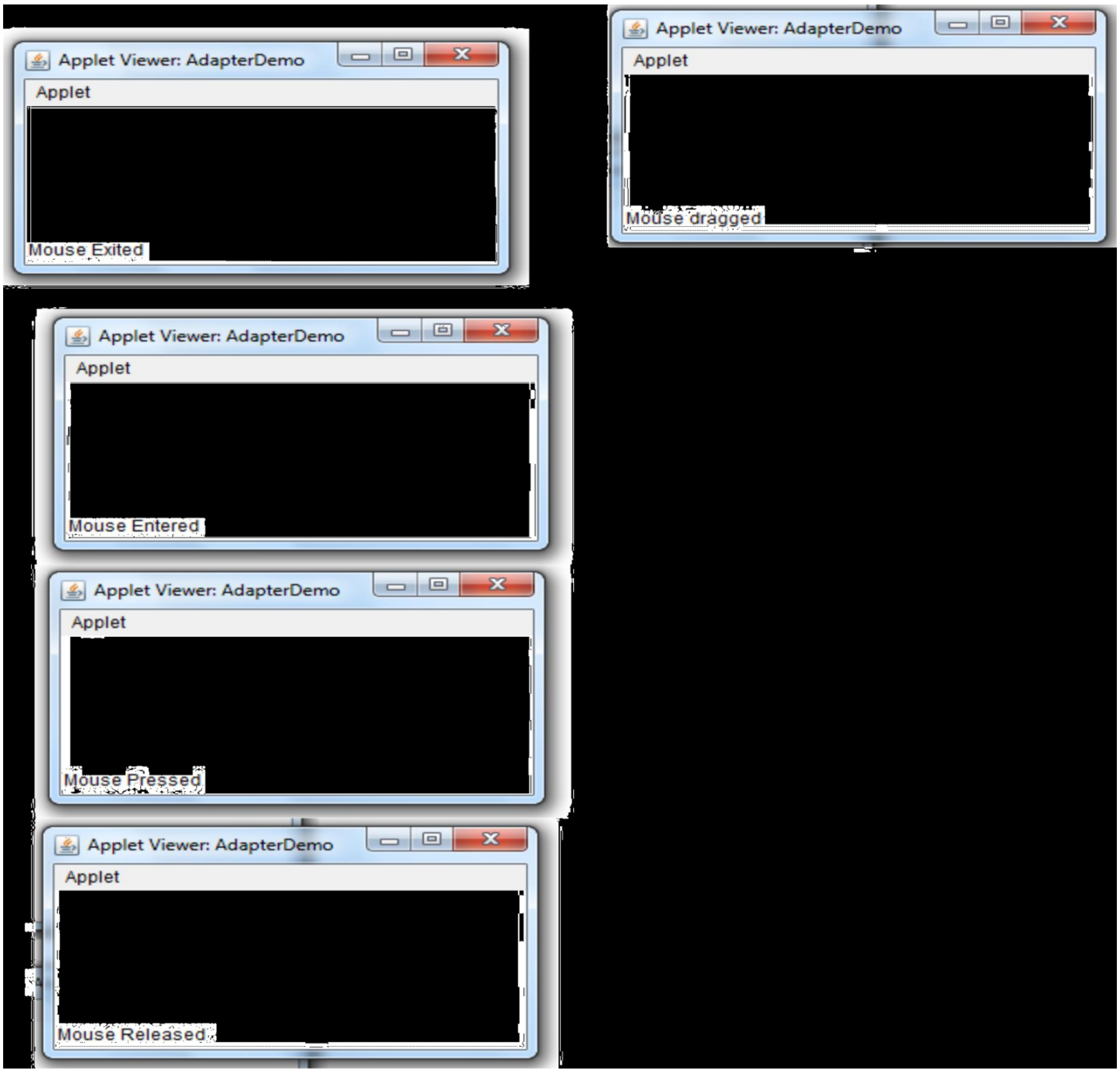
```
STEP1: START
STEP2: Create an applet extending from an Applet class.
STEP3: In the init()method add both MouseListener and MouseMotionListener.
STEP4: Extend the MouseAdapter and MouseMotionAdapter classes.
STEP5: Override the following methods
    a) mouseClicked()
    b) mouseEntered()
    c) mouseExited()
    d) mousePressed()
    e) mouseReleased()
    f) mouseDragged ()
    g) mouseMoved()
STEP6: Using showStatus() method display the message.
STEP7: Display the necessary information on the screen.
STEP8: END
```

SOURCE CODE:

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

/*<applet code="MouseEvents" width=300 height=300>

</applet>*/

public class MouseEvents extends Applet implements MouseListener, MouseMotionListener

{

        String msg = "";

        int mouseX = 0, mouseY = 0; // coordinates of mouse

        public void init()

        {

                addMouseListener(this);

                addMouseMotionListener(this);

        }

        public void mouseClicked(MouseEvent me)

        {

                mouseX = 0;
```

```
            mouseY = 10;
            msg = "Mouse clicked.";
            repaint();
    }
    public void mouseEntered(MouseEvent me)
    {
            mouseX = 0;
            mouseY = 10;
            msg = "Mouse entered.";
            repaint();
    }
    public void mouseExited(MouseEvent me)
    {
            mouseX = 0;
            mouseY = 10;
            msg = "Mouse exited.";
            repaint();
    }
    public void mousePressed(MouseEvent me)
    {
            mouseX = me.getX();
            mouseY = me.getY();
            msg = "Pressed";
            repaint();
    }
    public void mouseReleased(MouseEvent me)
    {
            mouseX = me.getX();
            mouseY = me.getY();
            msg = "Released";
            repaint();
    }
    public void mouseDragged(MouseEvent me)
    {
            mouseX = me.getX();
            mouseY = me.getY();
```

```
                msg = "Dragged";
                showStatus("Dragging mouse at " + mouseX + ", " + mouseY);
                repaint();
        }
        public void mouseMoved(MouseEvent me)
        {
                showStatus("Moving mouse at " + me.getX() + ", " + me.getY());
        }
        public void paint(Graphics g)
        {
                g.drawString(msg, mouseX, mouseY);
        }
}
```
OUTPUT:



```
Mouse Exited
```
```
Mouse dragged
```
```
Mouse Entered
```
```
Mouse Pressed
```
```
Mouse Released
```

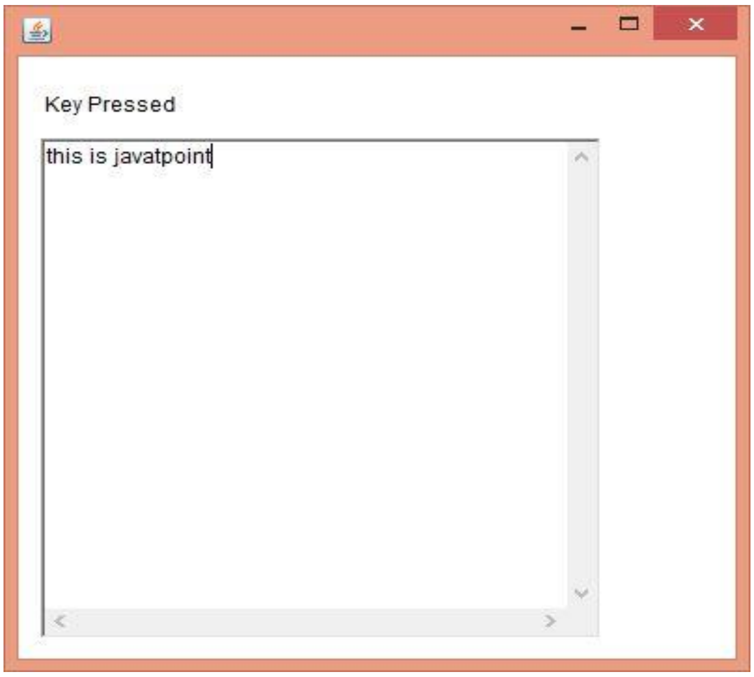
ALGORITHM:

```
STEP1: START
STEP2: Create a class extending from a Frame and implementing KeyListener.
STEP3: Create a TextArea and add KeyListener to it.
STEP4: Set bounds of the text area using setBounds() method.
STEP5: Set size to the frame using setSize() method.
STEP6: Override the following methods
        a) keyPressed()
        b) keyReleased()
        c) keyTyped()
STEP7: Using setText() method display the message in the text area.
STEP8: Display the necessary information on the screen.
STEP9: END
```

SOURCE CODE:

```
import java.awt.*;
    import java.awt.event.*;
    public class KeyListenerExample extends Frame implements KeyListener{
        Label l;
        TextArea area;
        KeyListenerExample(){
            l=new Label();
            l.setBounds(20,50,100,20);
            area=new TextArea();
            area.setBounds(20,80,300, 300);
            area.addKeyListener(this);
            add(l);
             add(area);
            setSize(400,400);
            setLayout(null);
            setVisible(true);
        }
        public void keyPressed(KeyEvent e) {
            l.setText("Key Pressed");
        }
        public void keyReleased(KeyEvent e) {
            l.setText("Key Released");
        }
        public void keyTyped(KeyEvent e) {
            l.setText("Key Typed");
        }

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

OUTPUT:


Key Pressed

```
this is javatpoint
```

VIVA - VOCE:

1. What is the purpose of adapter classes?

    Java adapter classes provide the default implementation of listener interfaces. If you inherit the adapter class, you will not be forced to provide the implementation of all the methods of listener interfaces. The adapter classes are found in java.awt.event and javax.swing.event packages.

2. What is the purpose of MouseListener?

    The Java MouseListener is notified whenever you change the state of mouse. It is notified against MouseEvent. The MouseListener interface is found in java.awt.event package. It has five methods. Methods of MouseListener interface are given below:

```
 public abstract void mouseClicked(MouseEvent e);
public abstract void mouseEntered(MouseEvent e);
public abstract void mouseExited(MouseEvent e);
public abstract void mousePressed(MouseEvent e);
 public abstract void mouseReleased(MouseEvent e);.
```

3. What is a the purpose of MouseMotionListener?

The Java MouseMotionListener is notified whenever you move or drag mouse. It is notified against MouseEvent. The MouseMotionListener interface is found in java.awt.event package. It has two methods.Methods of MouseMotionListener interface are given below:

public abstract void mouseDragged(MouseEvent e);

public abstract void mouseMoved(MouseEvent e);

4. What is a the purpose of KeyListener?

The Java KeyListener is notified whenever you change the state of key. It is notified against KeyEvent. The KeyListener interface is found in java.awt.event package. It has three methods.

Methods of KeyListener interface are given below:

public abstract void keyPressed(KeyEvent e);

public abstract void keyReleased(KeyEvent e);

public abstract void keyTyped(KeyEvent e);

5. What is the purpose of showStatus() method and repaint() method?

The **showStatus( )** method is used to display the information in applet windows status bar of the browser or Appletviewer. This is used for debugging, because it gives you an easy way to output message.

The **repaint( )** method causes the AWT runtime system to execute the update () method of the Component class which clears the window with the background color of the applet and then calls the paint () method.

6. What is the difference between Frame and JFrame?

A) Frame is part of java.awt package and exists since JDK1.0. JFrame is part of javax.swing package and exists since JDK1.1.3 or something. Frame extends Window. JFrame extends Frame. You can directly add components to Frame. You add components to JFrame.getContentPane(). JFrame implements javax.swing.RootPane Container. Frame does not.