Graphical User Interfaces

Mouse Listeners and Key Listeners

COMP2603
Object Oriented Programming 1

Week 7

Outline

- MouseListeners
- MouseEvents
- KeyListeners
- KeyEvents

Reading: Chapter 17 - Intro. to Graphical User Interface Programming: P. Mohan 2013

Event Driven Programming

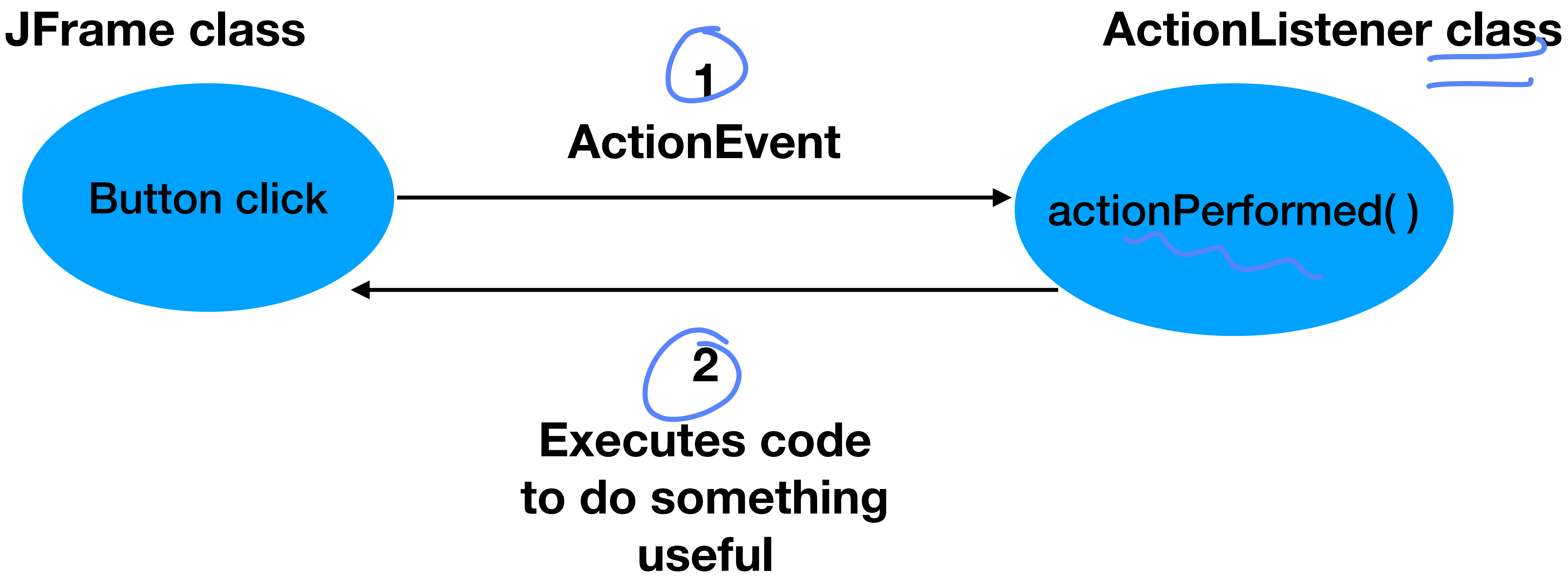
This involves writing application-specific code to take some action when a pre-determined event occurs.

Such code is referred to as an event handler.

Event Handling Methods

Interface	Interface Methods	Event Class
ActionListener	void actionPerformed (ActionEvent e)	ActionEvent
KeyListener	void keyPressed (KeyEvent e) void keyReleased (KeyEvent e) void keyTyped (KeyEvent e)	KeyEvent
MouseListener	void mouseClicked (MouseEvent e) void mouseEntered (MouseEvent e) void mouseExited (MouseEvent e) void mousePressed (MouseEvent e) void mouseReleased (MouseEvent e)	MouseEvent

ActionListener



JFrame class

Button click

1

ActionEvent

ActionListener class

actionPerformed( )

2

Executes code
to do something
useful

ActionListener Code

```
import java.awt.event.ActionListener;
public class RegistrationGUI extends javax.swing.JFrame {
  private javax.swing.JButton clearButton; //declare JButton
  private void initComponents() {
    clearButton = new javax.swing.JButton(); //initialise JButton
    //create ActionListener object – requires another class
    ClearButtonListener al = new ClearButtonListener();
    //add ActionListener object to clear button
    clearButton.addActionListener(cButtonListener);
     …}
  }
```

ActionListener Code

```
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class ClearButtonListener implements
ActionListener{
   public void actionPerformed(ActionEvent evt){
       /*clear data from GUI components,
         which requires access to these components*/
   }
  }
```

Netbeans Code

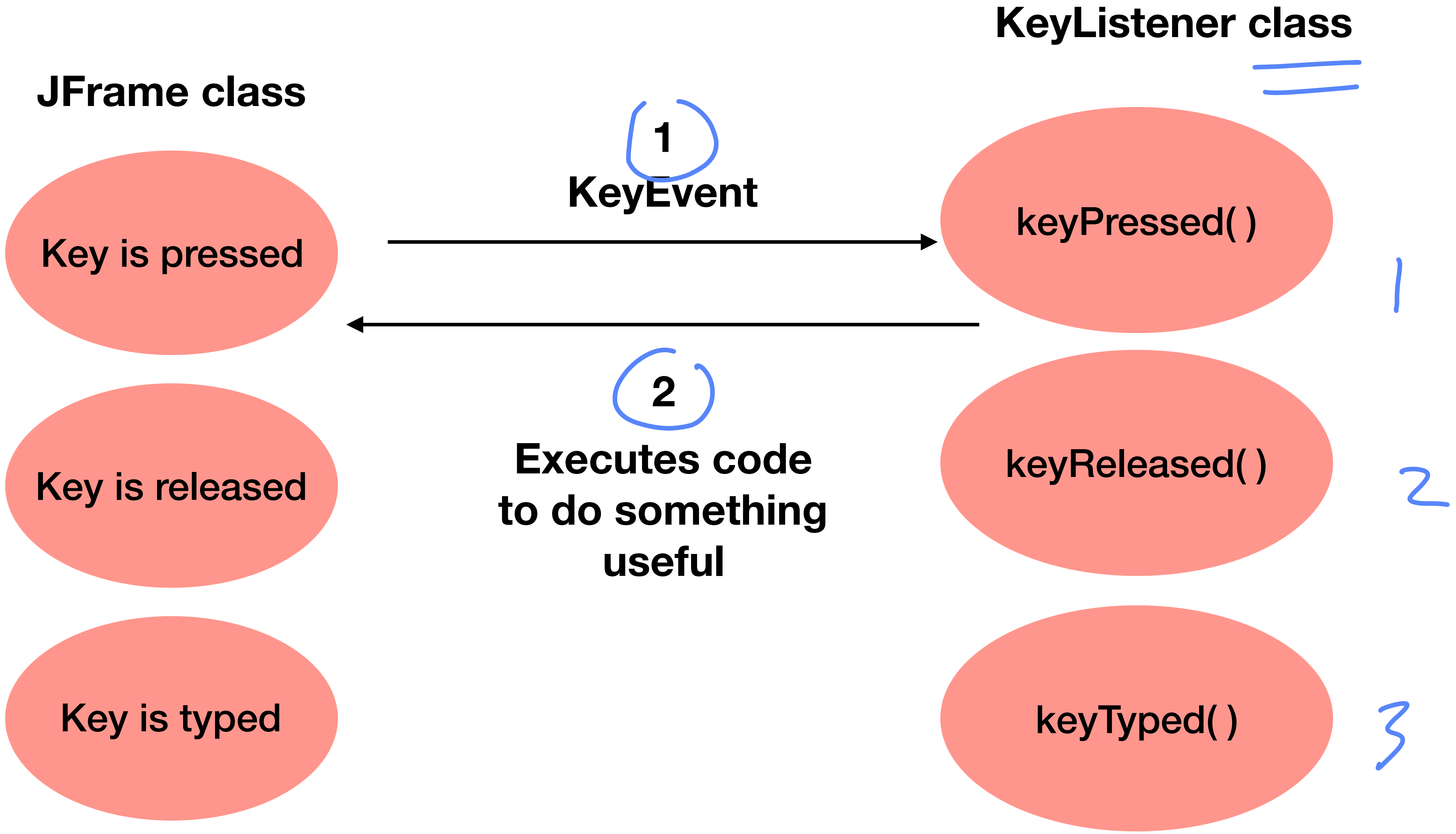
```
private void initComponents() {
  //initialise JButton
  clearButton = new javax.swing.JButton();
  //creating the ActionListener class AND adding it at once
  clearButton.addActionListener(
       new java.awt.event.ActionListener() {
          public void actionPerformed(
          java.awt.event.ActionEvent evt{
                clearButtonActionPerformed(evt);
          }
        });
…
}
```

This strategy grants the ActionListener class direct access to the GUI components

Event Handling Methods

Interface	**Interface Methods**	**Event Class**
ActionListener	void actionPerformed (ActionEvent e)	ActionEvent
KeyListener	void keyPressed (KeyEvent e) void keyReleased (KeyEvent e) void keyTyped (KeyEvent e)	KeyEvent
MouseListener	void mouseClicked (MouseEvent e) void mouseEntered (MouseEvent e) void mouseExited (MouseEvent e) void mousePressed (MouseEvent e) void mouseReleased (MouseEvent e)	MouseEvent

KeyListener



JFrame class

Key is pressed

Key is released

Key is typed

1

KeyEvent

2

Executes code to do something useful

KeyListener class

keyPressed( )

keyReleased( )

keyTyped( )

1

2

3

KeyListener Code

If we wanted to know which keys the user pressed, then the KeyListener interface needs to be implemented. !

5

public class KeyHandler implements KeyListener{

2 1 3

```
public void keyPressed(java.awt.event.KeyEvent e){
    int keyCode = e.getKeyCode();
    String keyText = e.getKeyText(keyCode);
    System.out.println(“You pressed: “ + keyText);
}
```

"a"

KeyListener Code

If we wanted to know which keys the user pressed, then the KeyListener interface needs to be implemented

public class KeyHandler2 implements KeyListener }

```
public void keyReleased(java.awt.event.KeyEvent e){
    int keyCode = e.getKeyCode();
    String keyText = e.getKeyText(keyCode);
    System.out.println(“You released: “ + keyText);
}
```

"a"

KeyListener Code

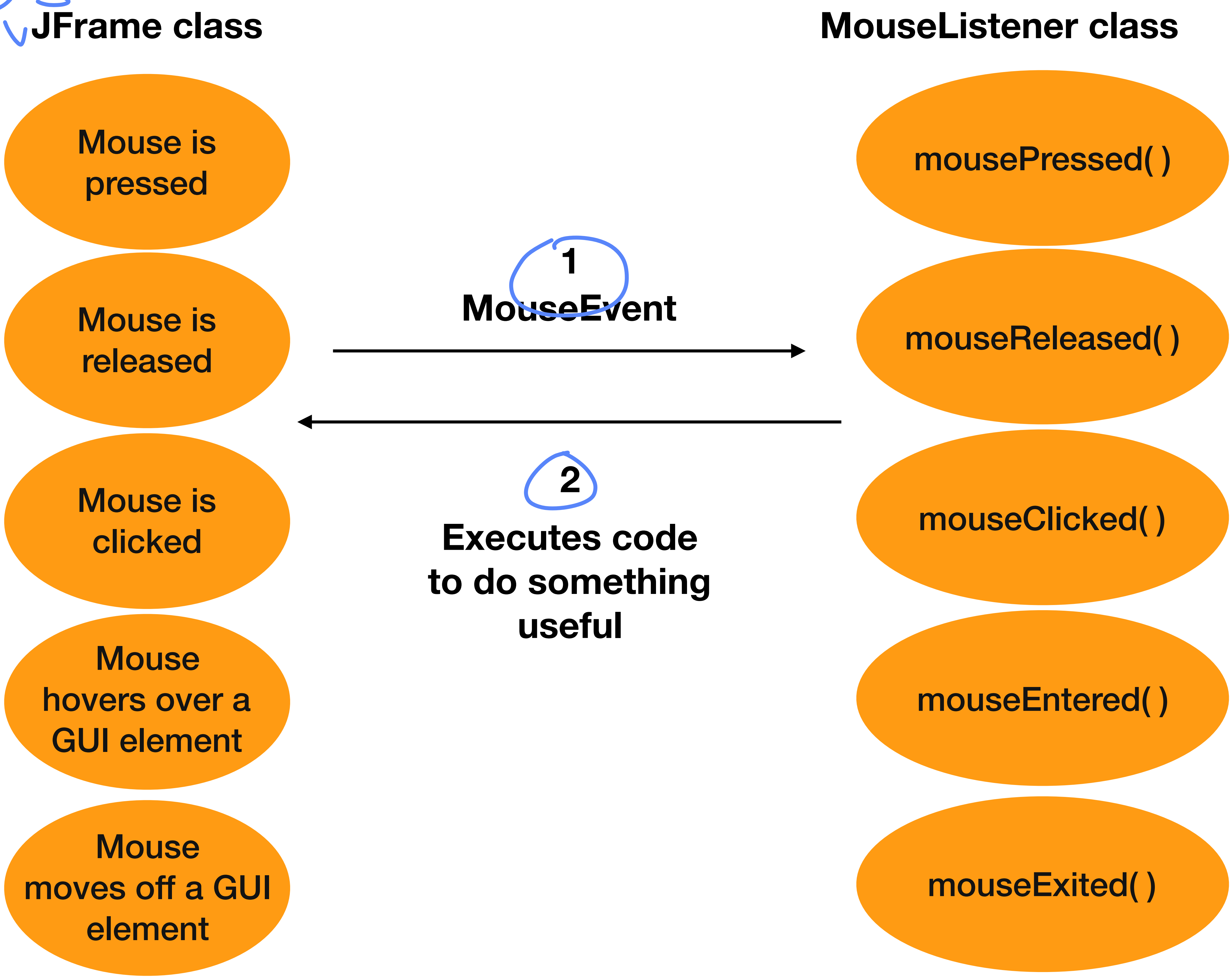
The KeyEvent for key typed is only generated if a valid Unicode character could be generated, e.g. for the Shift key

```
public void keyTyped(java.awt.event.KeyEvent e){
    int keyCode = e.getKeyCode();
    String keyText = e.getKeyText(keyCode);
    System.out.println(“You typed: “ + keyText);
}
```

Event Handling Methods

Interface	Interface Methods	Event Class
ActionListener	void actionPerformed (ActionEvent e)	ActionEvent
KeyListener	void keyPressed (KeyEvent e) void keyReleased (KeyEvent e) void keyTyped (KeyEvent e)	KeyEvent
MouseListener	void mouseClicked (MouseEvent e) void mouseEntered (MouseEvent e) void mouseExited (MouseEvent e) void mousePressed (MouseEvent e) void mouseReleased (MouseEvent e)	MouseEvent

MouseListener



JFrame class

MouseListener class

Mouse is pressed

Mouse is released

Mouse is clicked

Mouse hovers over a GUI element

Mouse moves off a GUI element

1 MouseEvent

2 Executes code to do something useful

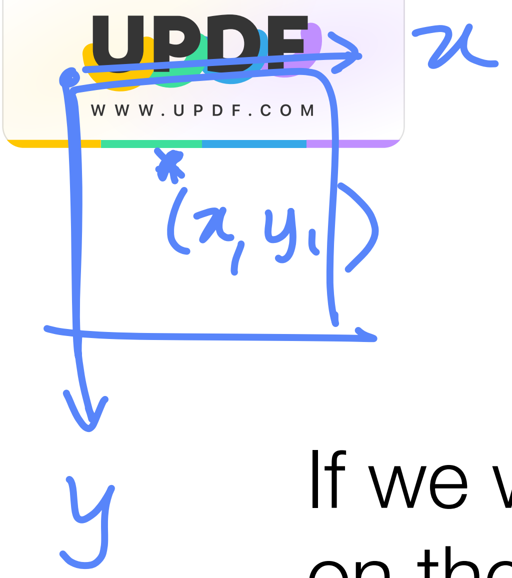
mousePressed( )

mouseReleased( )

mouseClicked( )

mouseEntered( )

mouseExited( )

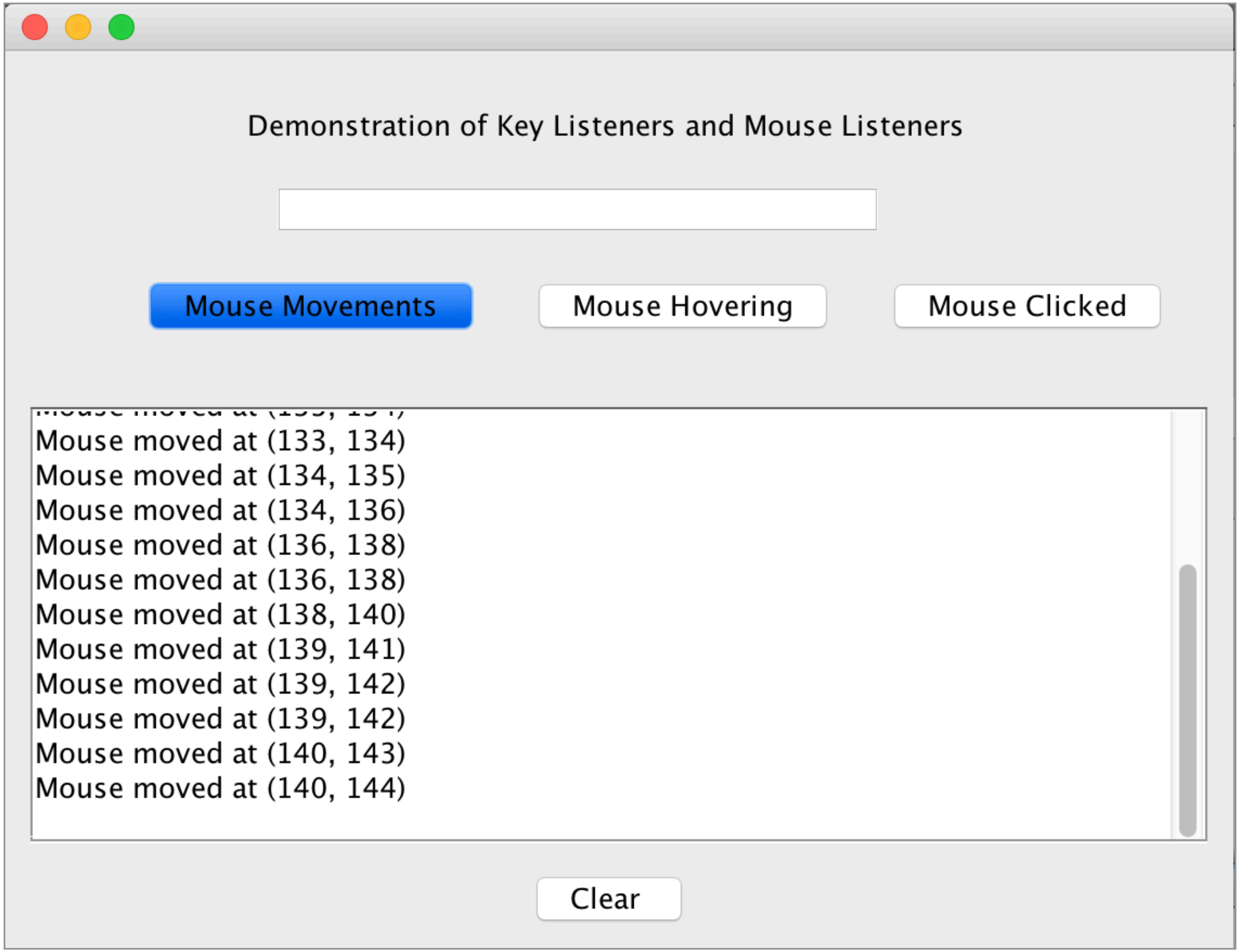


MouseListener Code

If we wanted to know if the user clicked the mouse somewhere on the window in an area not occupied by a GUI component.

```
public void mouseClicked(java.awt.event.MouseEvent e){
    int x = e.getX();
    int y = e.getY();
    System.out.println(“Mouse click at: (“ + x + “, “ + y + “)”);
}
```

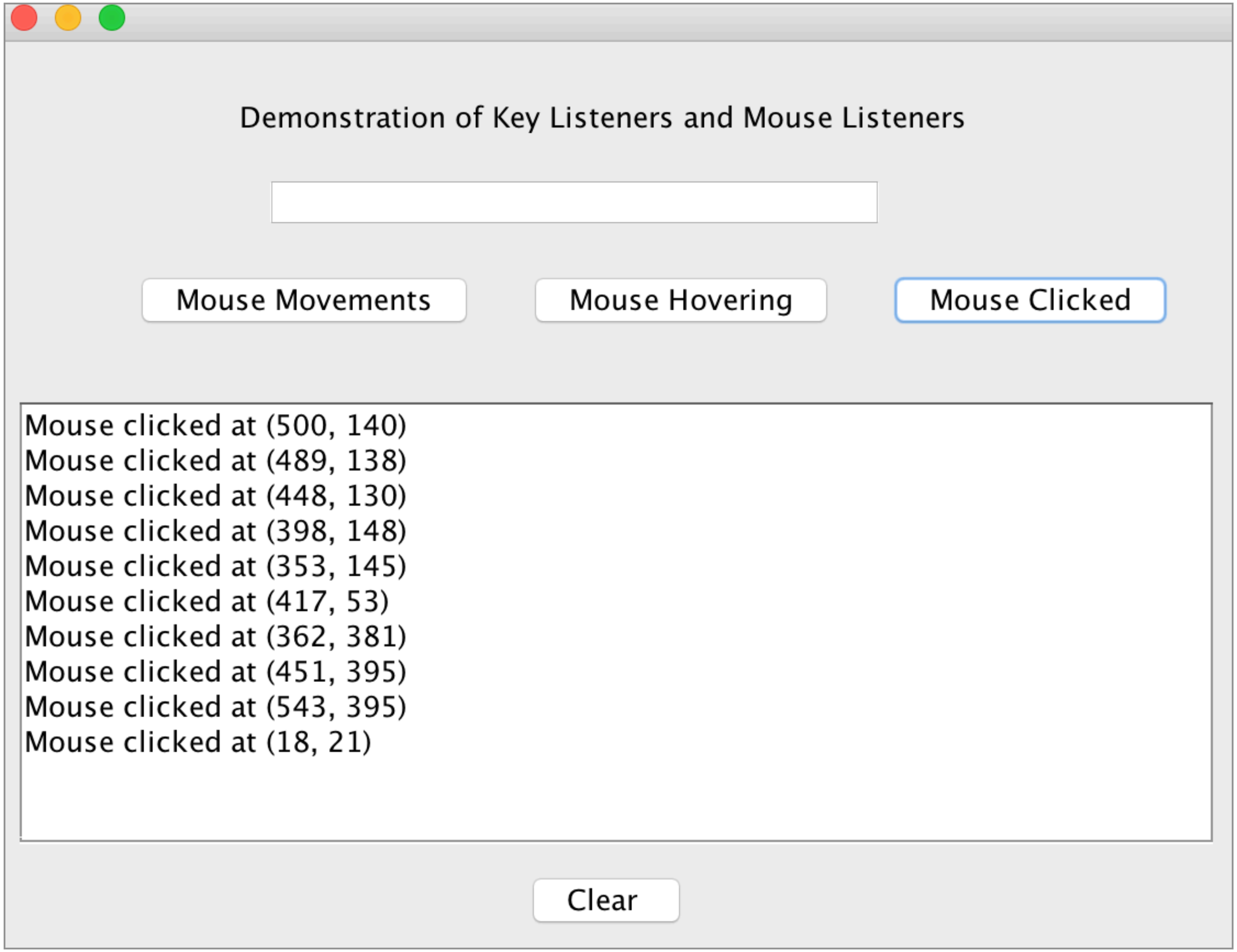
Example - Mouse movements



Demonstration of Key Listeners and Mouse Listeners

```
Mouse moved at (133, 134)
Mouse moved at (134, 135)
Mouse moved at (134, 136)
Mouse moved at (136, 138)
Mouse moved at (136, 138)
Mouse moved at (138, 140)
Mouse moved at (139, 141)
Mouse moved at (139, 142)
Mouse moved at (139, 142)
Mouse moved at (140, 143)
Mouse moved at (140, 144)
```

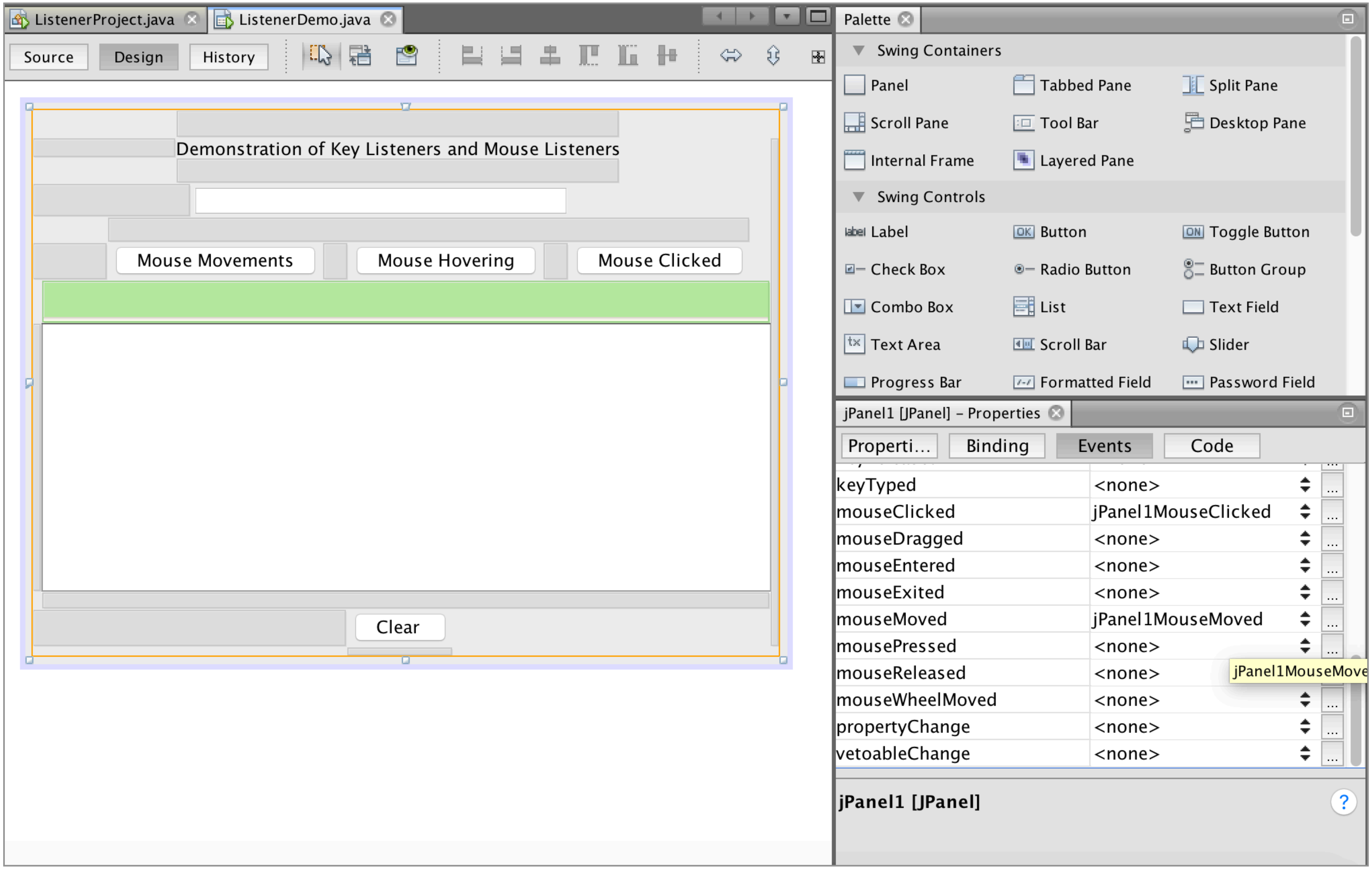
Example - Mouse clicks



Demonstration of Key Listeners and Mouse Listeners

```
Mouse clicked at (500, 140)
Mouse clicked at (489, 138)
Mouse clicked at (448, 130)
Mouse clicked at (398, 148)
Mouse clicked at (353, 145)
Mouse clicked at (417, 53)
Mouse clicked at (362, 381)
Mouse clicked at (451, 395)
Mouse clicked at (543, 395)
Mouse clicked at (18, 21)
```

Example - Mouse movements and clicks

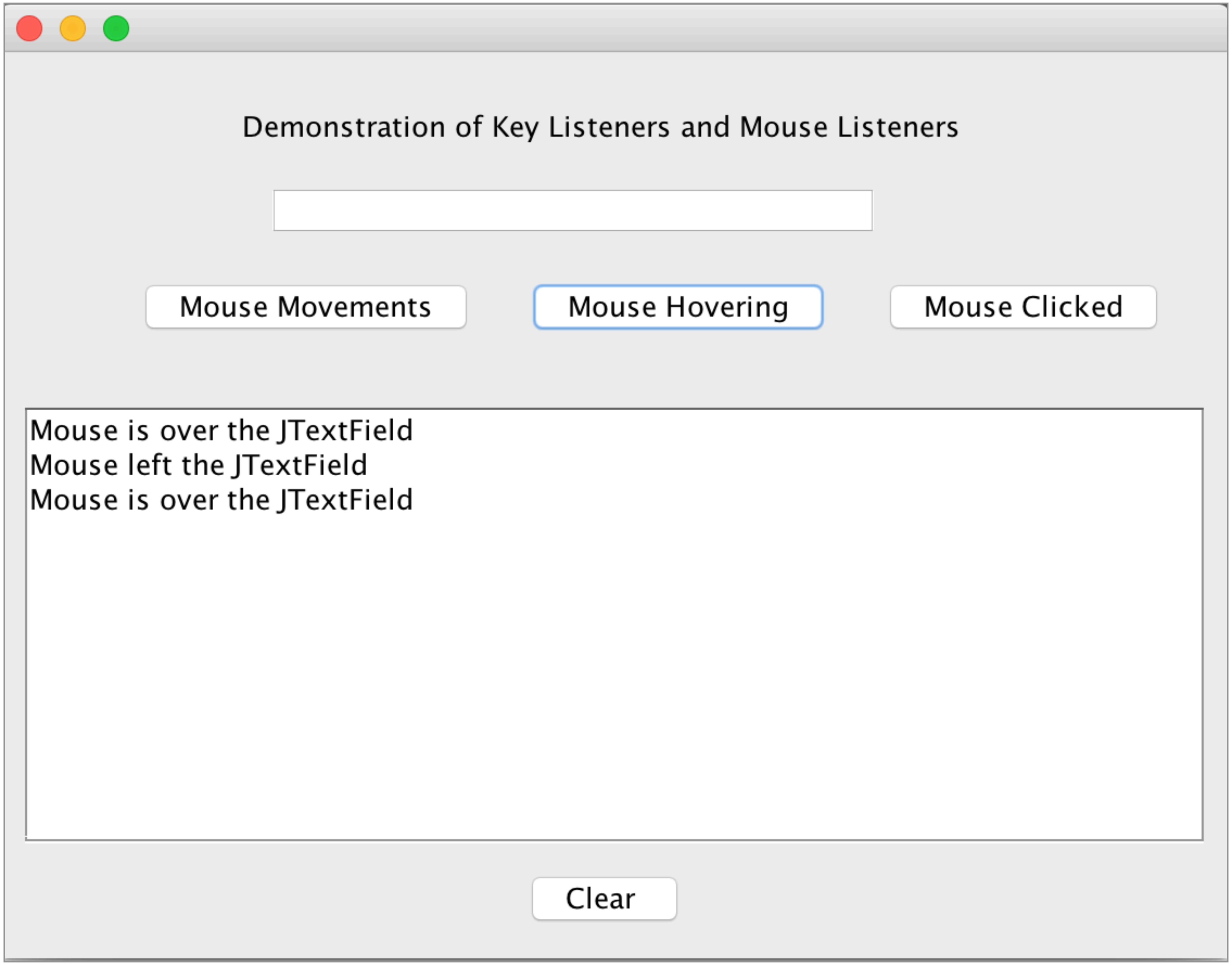


keyTyped	<none>
mouseClicked	jPanel1MouseClicked
mouseDragged	<none>
mouseEntered	<none>
mouseExited	<none>
mouseMoved	jPanel1MouseMoved
mousePressed	<none>
mouseReleased	<none>
mouseWheelMoved	<none>
propertyChange	<none>
vetoableChange	<none>

jPanel1MouseMove

jPanel1 [JPanel]

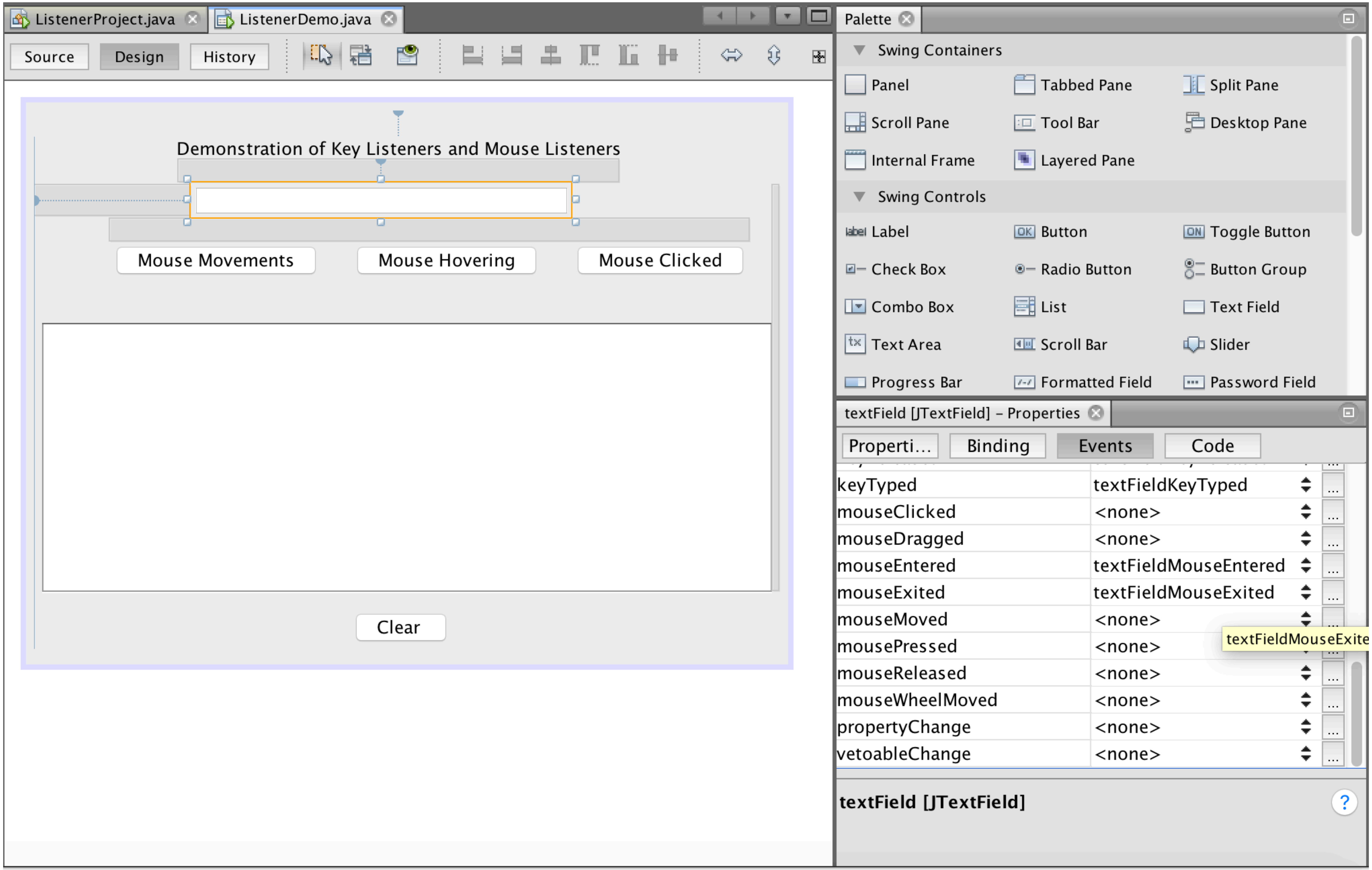
Example - Mouse enter and exit



Demonstration of Key Listeners and Mouse Listeners

```
Mouse is over the JTextField
Mouse left the JTextField
Mouse is over the JTextField
```

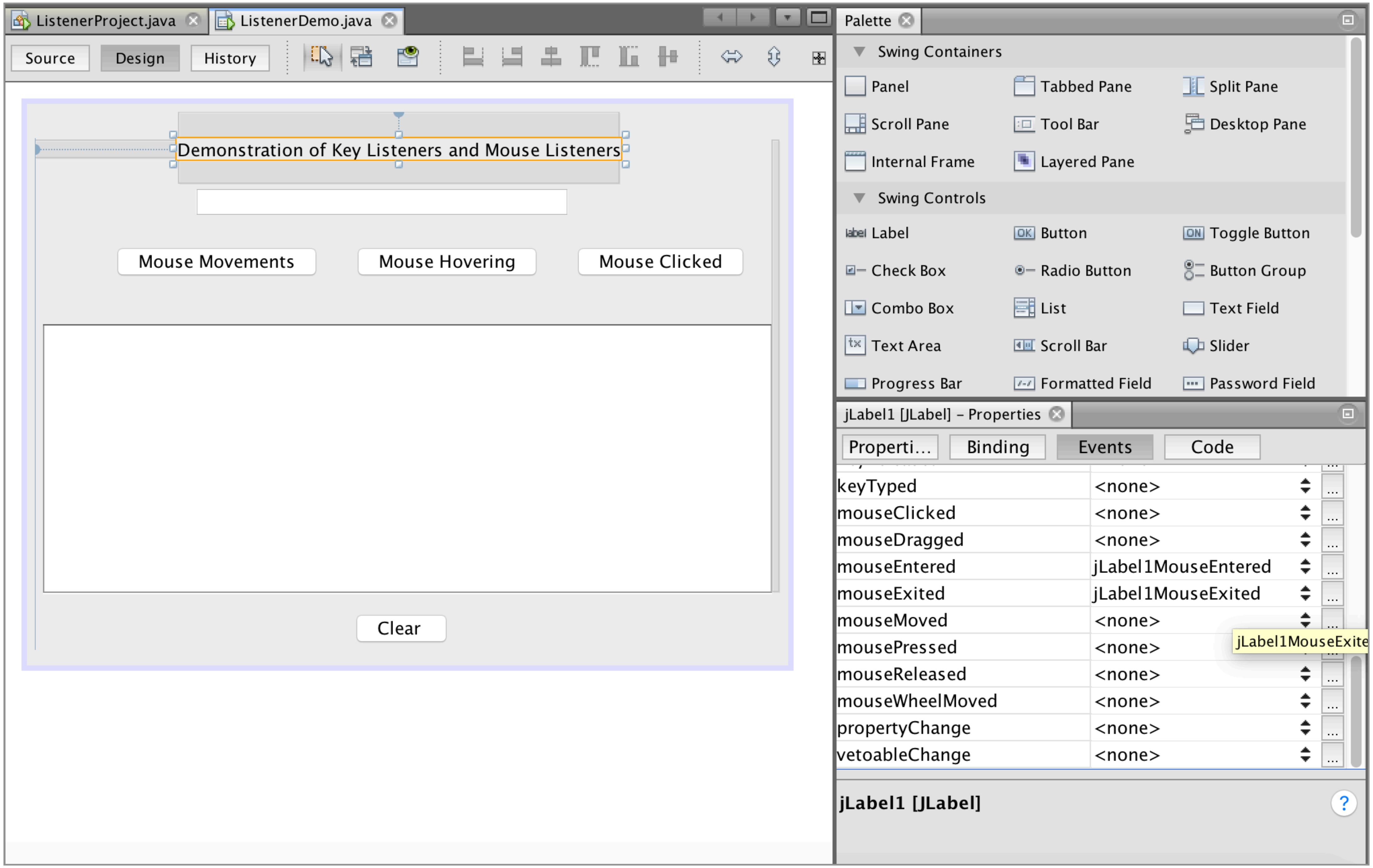
Example - Mouse enter and exit



keyTyped	textFieldKeyTyped
mouseClicked	<none>
mouseDragged	<none>
mouseEntered	textFieldMouseEntered
mouseExited	textFieldMouseExited
mouseMoved	<none>
mousePressed	<none>
mouseReleased	<none>
mouseWheelMoved	<none>
propertyChange	<none>
vetoableChange	<none>

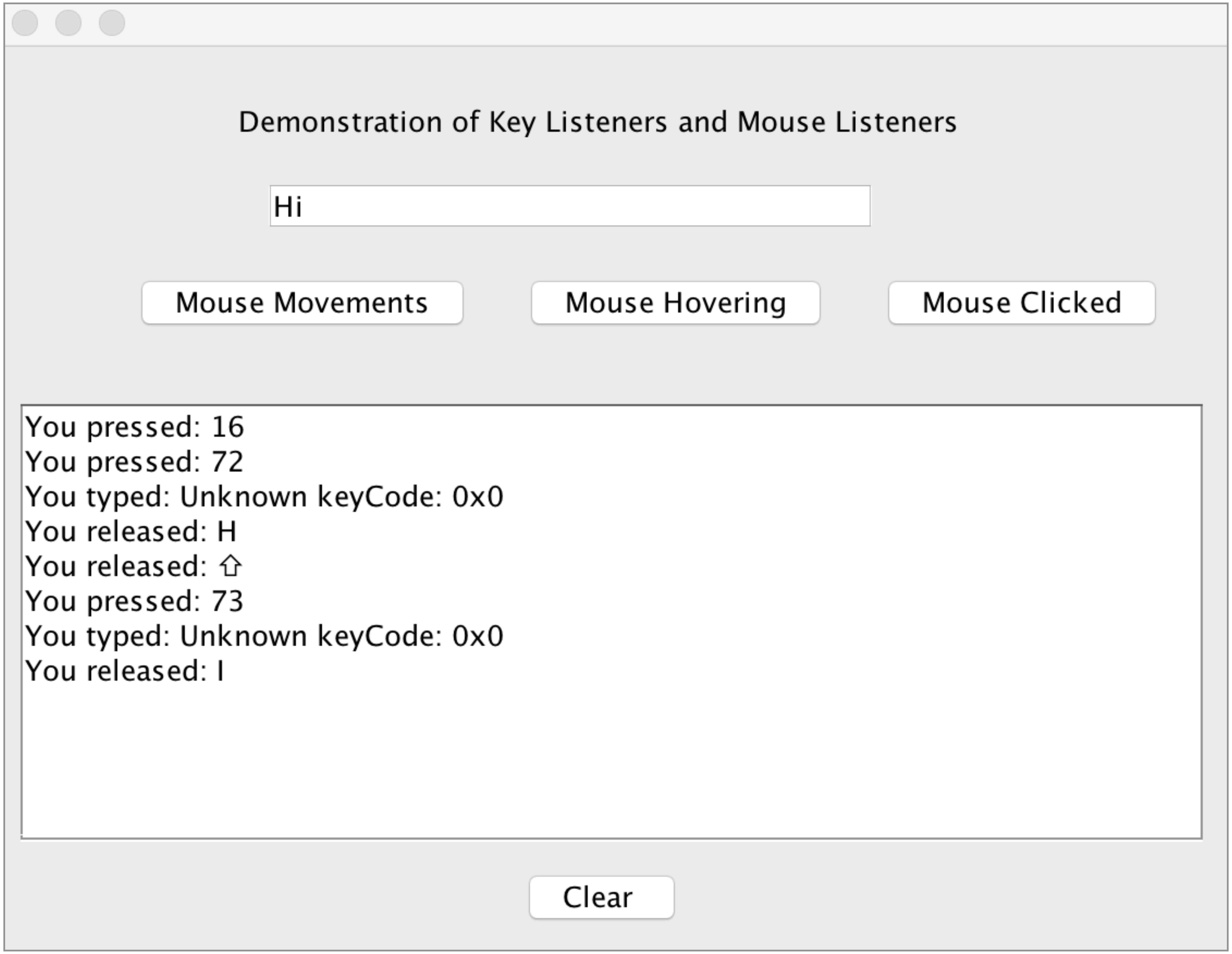
textField [JTextField]

Example - Mouse movements and clicks



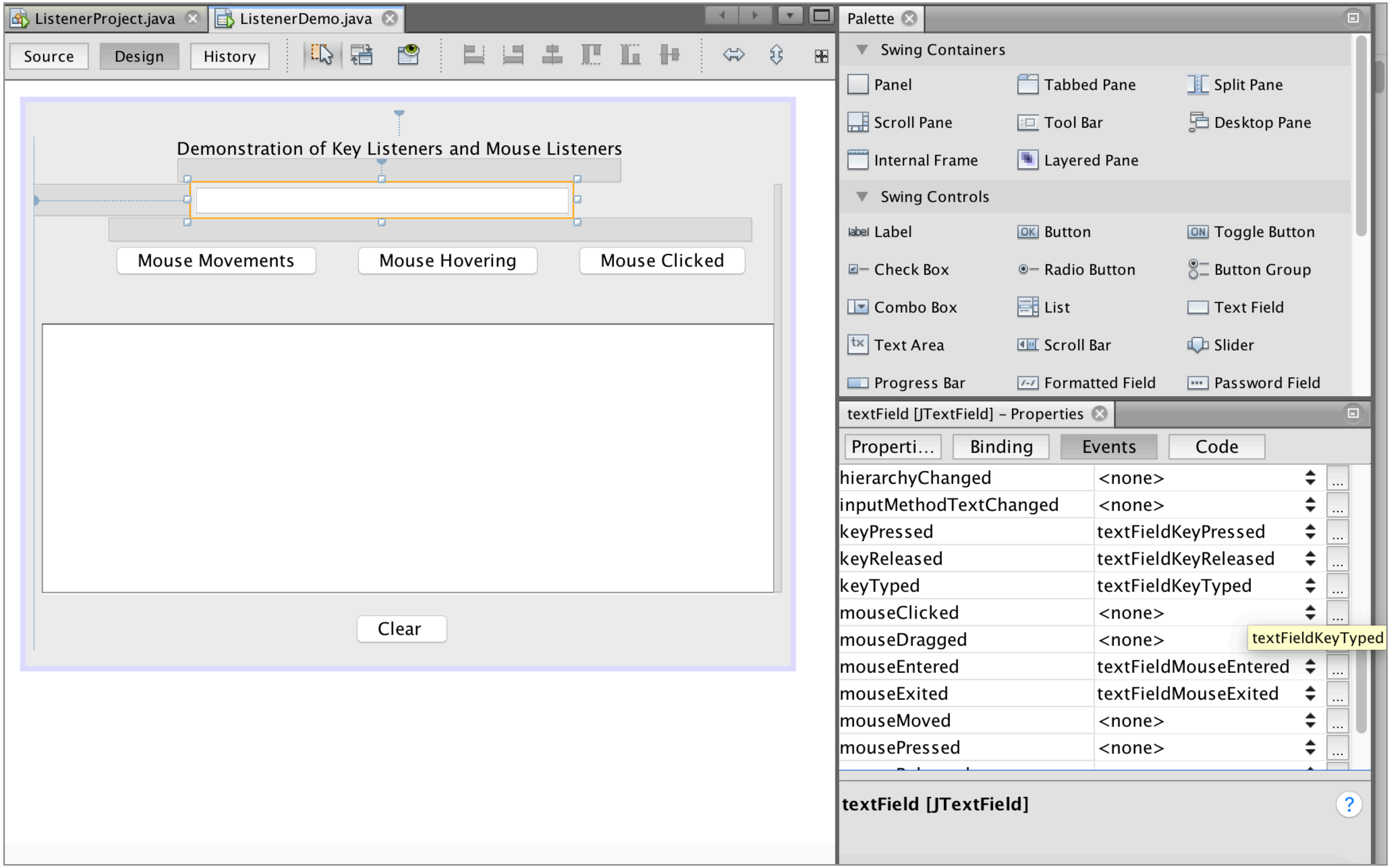
keyTyped	<none>
mouseClicked	<none>
mouseDragged	<none>
mouseEntered	jLabel1MouseEntered
mouseExited	jLabel1MouseExited
mouseMoved	<none>
mousePressed	<none>
mouseReleased	<none>
mouseWheelMoved	<none>
propertyChange	<none>
vetoableChange	<none>

Example -
Key Pressed, Released and Typed



```
You pressed: 16
You pressed: 72
You typed: Unknown keyCode: 0x0
You released: H
You released: ⇧
You pressed: 73
You typed: Unknown keyCode: 0x0
You released: I
```

Example -
Key Pressed, Released and Typed



textField [JTextField] – Properties

hierarchyChanged	<none>
inputMethodTextChanged	<none>
keyPressed	textFieldKeyPressed
keyReleased	textFieldKeyReleased
keyTyped	textFieldKeyTyped
mouseClicked	<none>
mouseDragged	<none>
mouseEntered	textFieldMouseEntered
mouseExited	textFieldMouseExited
mouseMoved	<none>
mousePressed	<none>

textField [JTextField]