

# Graphical User Interfaces

Mouse Listeners and Key Listeners

COMP2603  
Object Oriented Programming 1

Week 7

# Outline

- MouseListeners
- MouseEvents
- KeyListeners
- KeyEvents

# Event Driven Programming

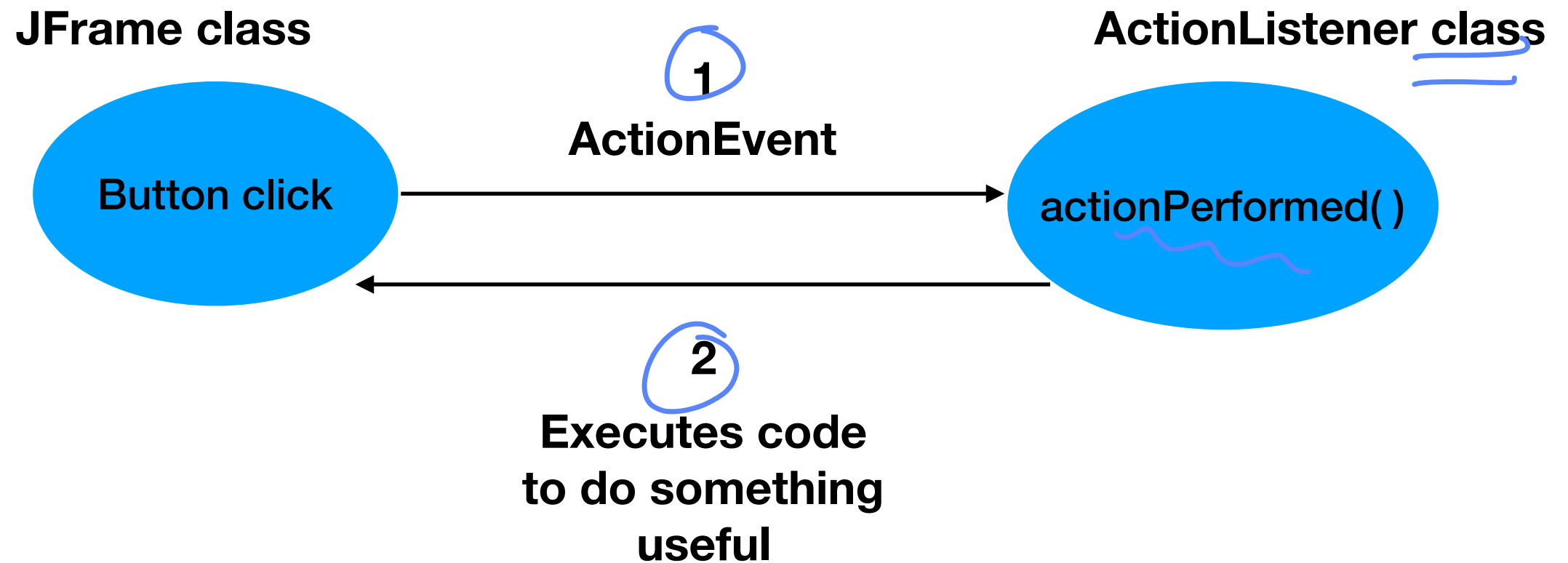
This involves writing application-specific code to take some action when a pre-determined event occurs.

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



# 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);  
                }  
            }  
    });  
    ...  
}
```

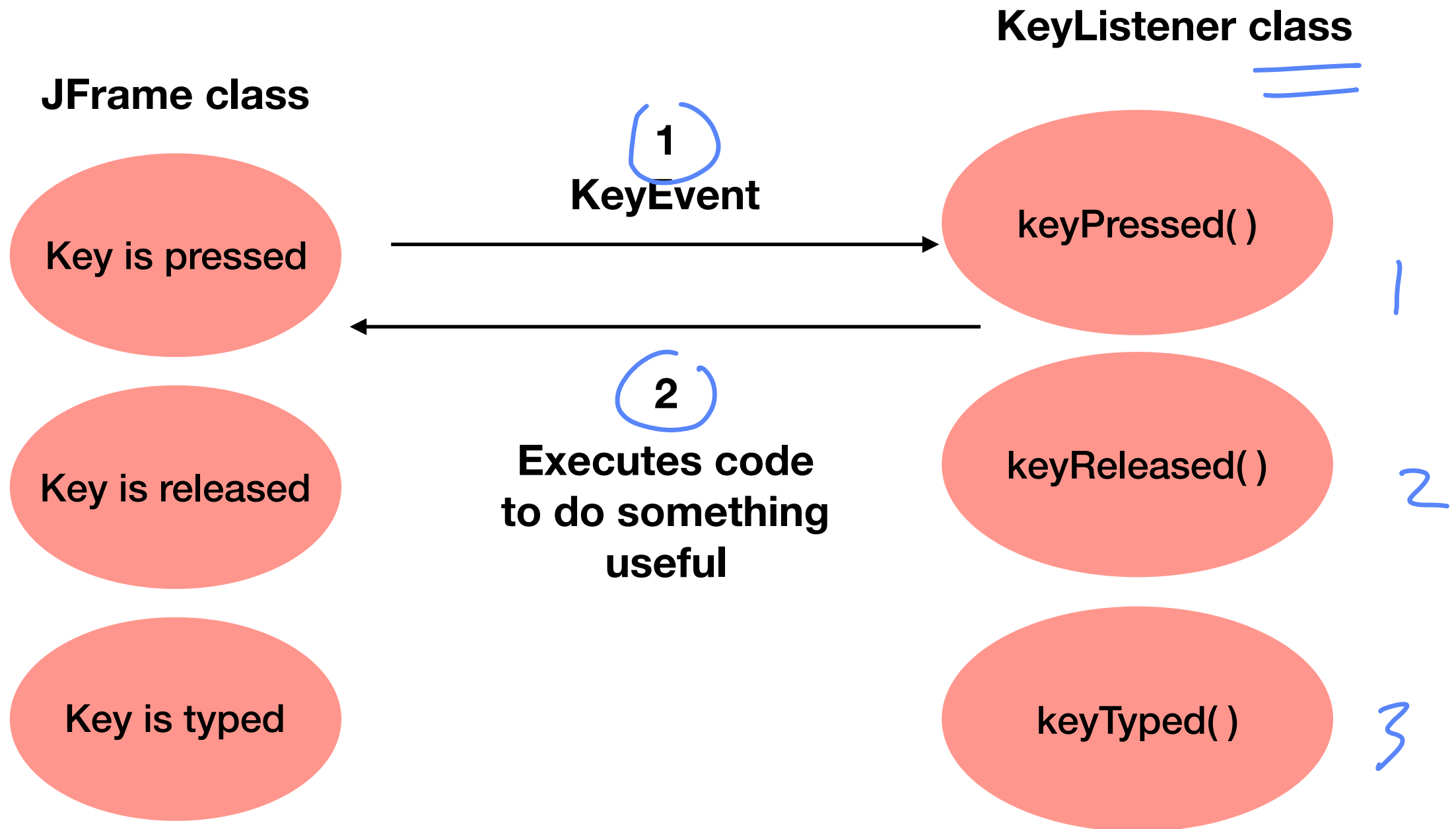
**This strategy grants the ActionListener class  
direct access to the GUI components**



# 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



# KeyListener Code

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

public class KeyHandler implements KeyListener {

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();
```

```
        "a" String keyText = e.getKeyText(keyCode);
```

```
        System.out.println("You released: " + keyText);
```

```
    }
```

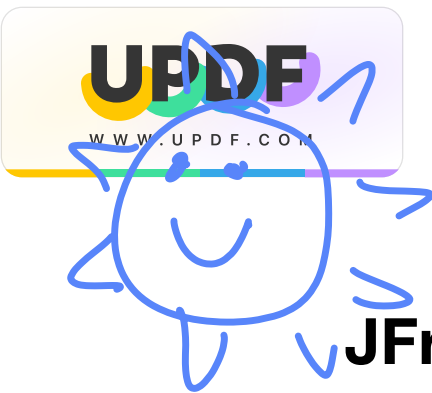
# KeyListener Code

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

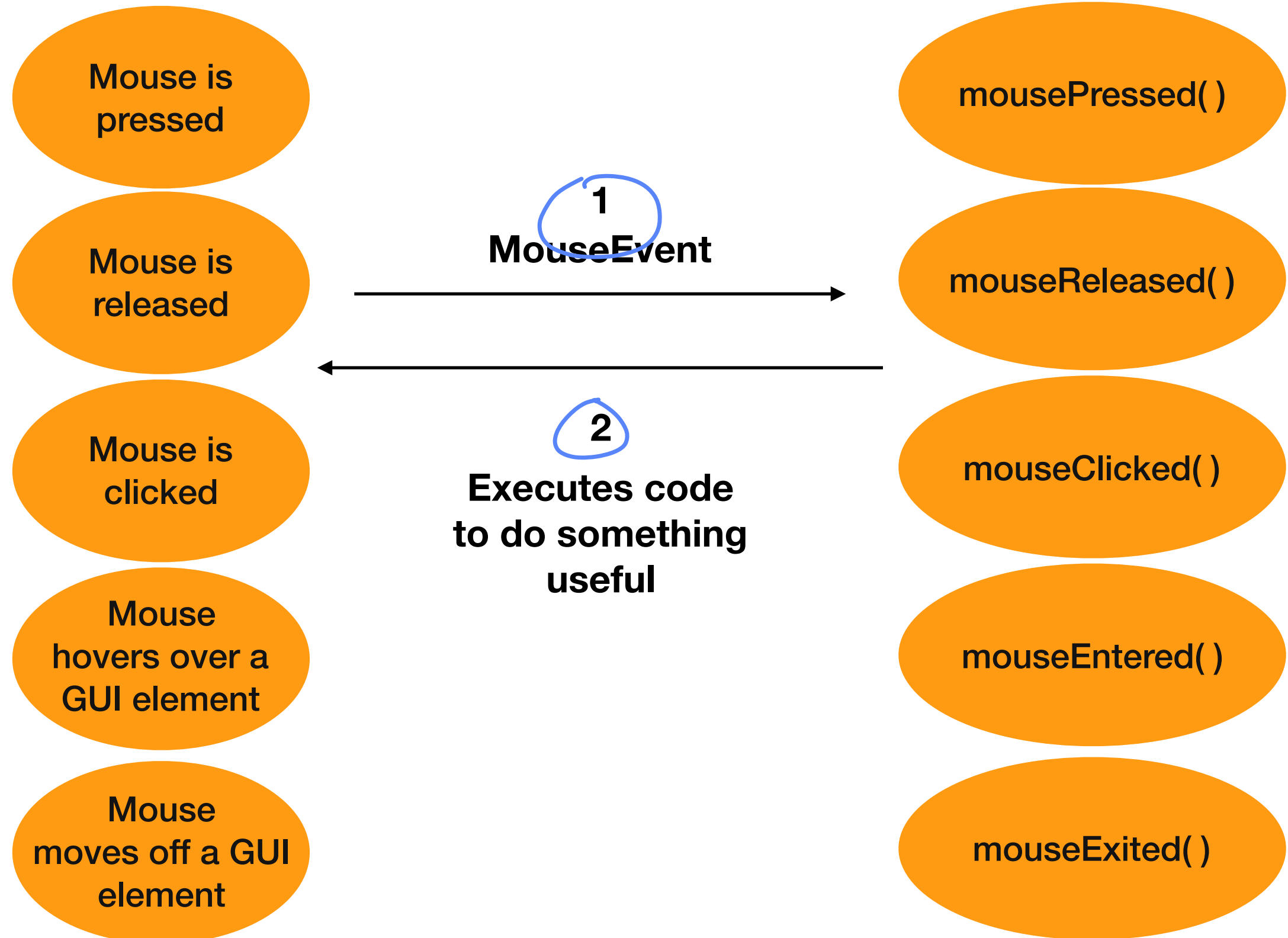
Interface	Interface Methods	Event Class
ActionListener	void actionPerformed (ActionEvent e)	ActionEvent
KeyListener	void keyPressed (KeyEvent e) void keyReleased (KeyEvent e) void keyTyped (KeyEvent e)	KeyEvent
MouseListener	void mouseClicked (MouseEvent e) void mouseEntered (MouseEvent e) void mouseExited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

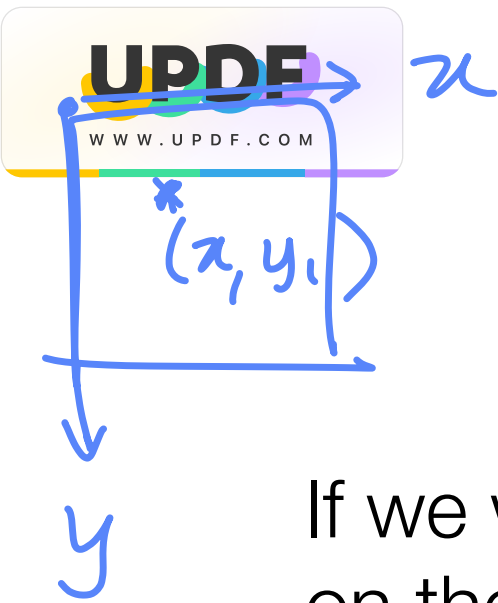


# MouseListener

**JFrame class**

**MouseListener class**





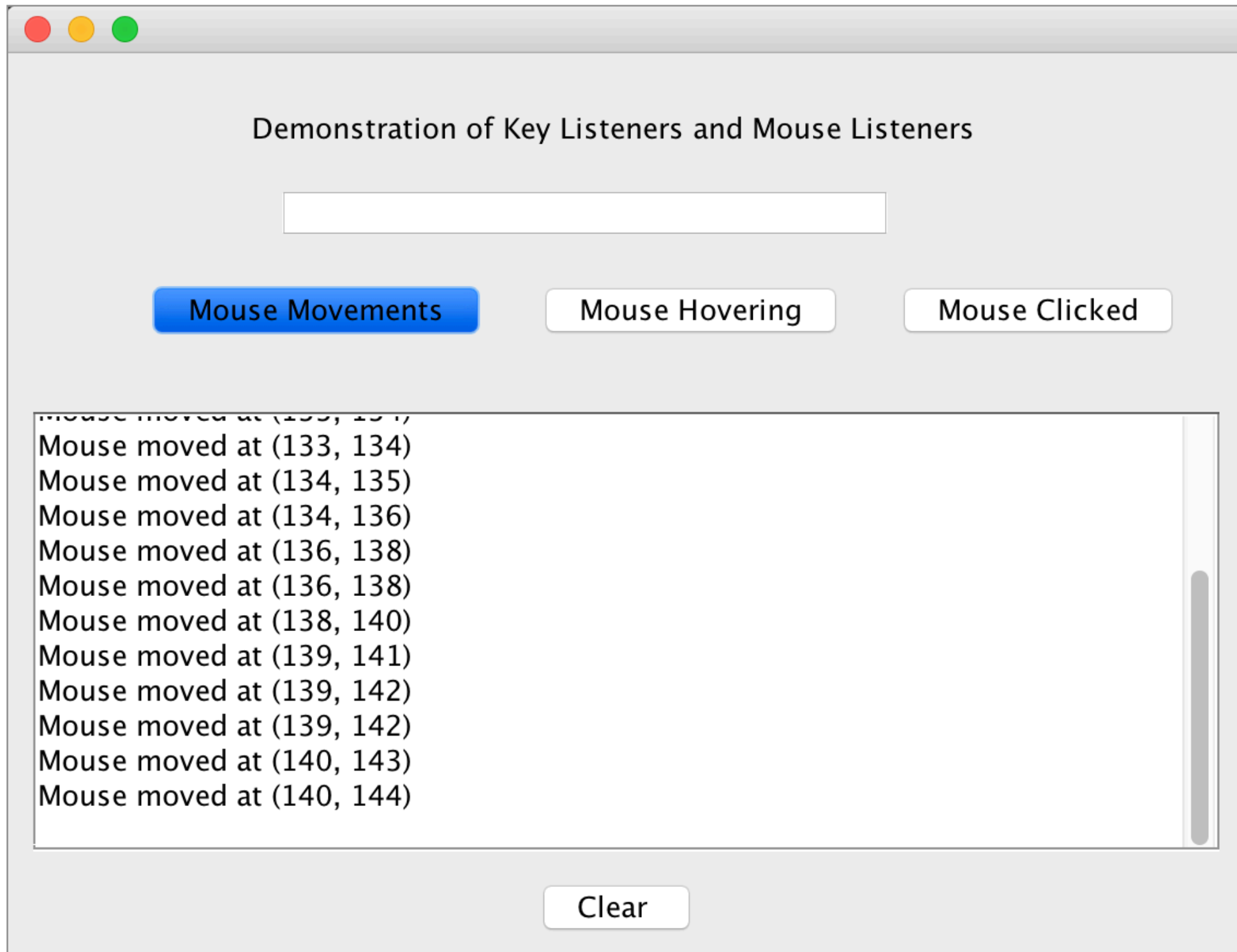
# MouseListener Code

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

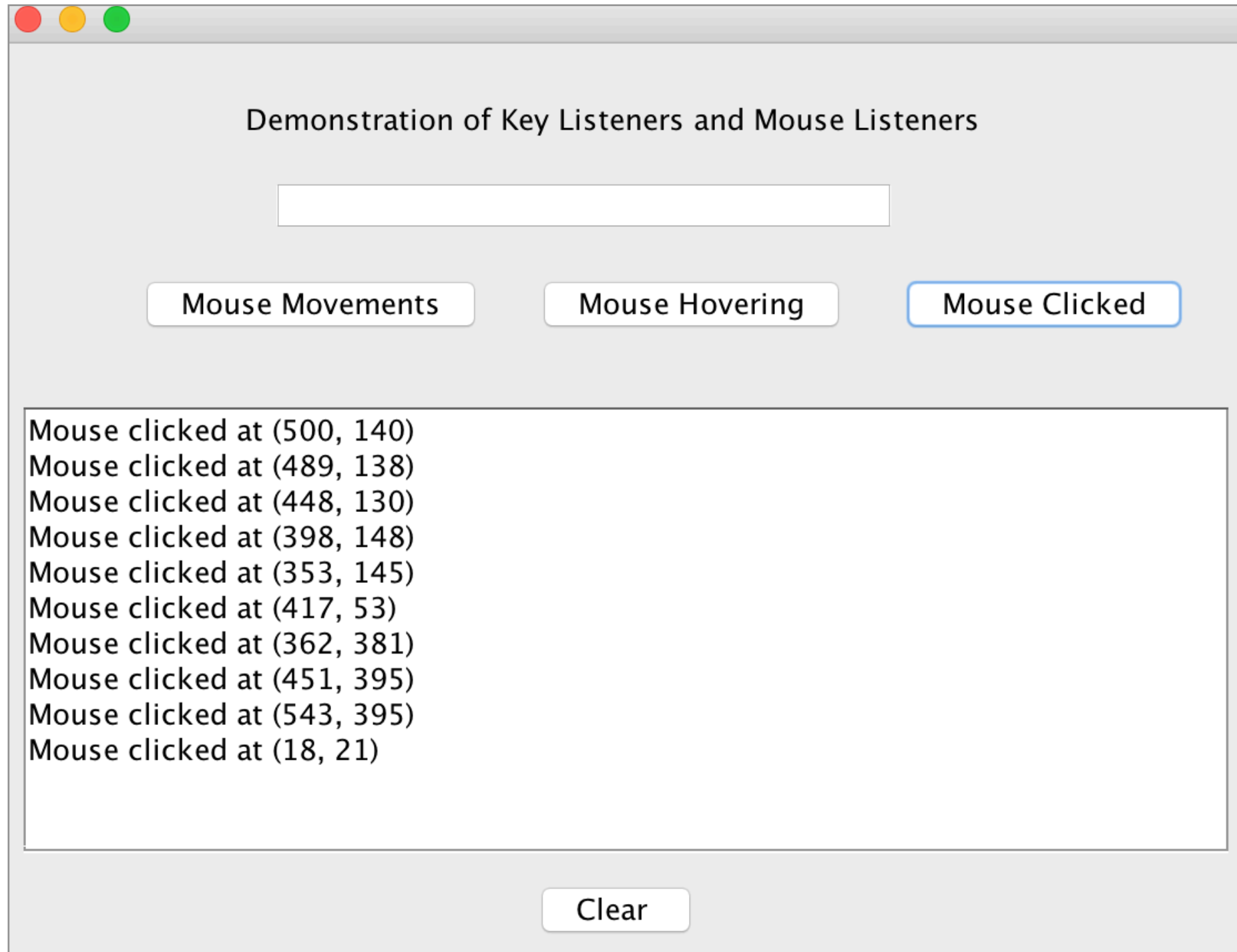
```
public void mouseClicked(java.awt.event.MouseEvent e){  
    int x = e.getX();  
    int y = e.getY();  
    System.out.println("Mouse click at: (" + x + ", " + y + ")");  
}
```



# Example - Mouse movements



# Example - Mouse clicks



Demonstration of Key Listeners and Mouse Listeners

Mouse Movements    Mouse Hovering    **Mouse Clicked**

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)

Clear

# Example - Mouse movements and clicks

The screenshot shows an IDE with two tabs: `ListenerProject.java` and `ListenerDemo.java`. The `ListenerDemo.java` tab is active, showing a Swing application in Design mode. The application window has a title bar and a content area with the following elements:

- A text field with the text "Demonstration of Key Listeners and Mouse Listeners".
- A text field below the title.
- Three buttons: "Mouse Movements", "Mouse Hovering", and "Mouse Clicked".
- A green rectangular area below the buttons.
- A "Clear" button at the bottom right.

The IDE interface includes a Palette with the following categories:

- Swing Containers:** Panel, Scroll Pane, Internal Frame, Tabbed Pane, Tool Bar, Layered Pane, Split Pane, Desktop Pane.
- Swing Controls:** Label, Check Box, Combo Box, Text Area, Progress Bar, Button, Radio Button, List, Scroll Bar, Formatted Field, Toggle Button, Button Group, Text Field, Slider, Password Field.

The Properties window for `jPanel1 [JPanel]` is open, showing the Events tab. The following table represents the data in the Events tab:

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

# Example - Mouse enter and exit

Demonstration of Key Listeners and Mouse Listeners

Mouse Movements    Mouse Hovering    Mouse Clicked

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

Clear

# Example - Mouse enter and exit

The screenshot shows an IDE with two tabs: 'ListenerProject.java' and 'ListenerDemo.java'. The 'Design' view is active, showing a Swing window titled 'Demonstration of Key Listeners and Mouse Listeners'. The window contains a text field, three buttons labeled 'Mouse Movements', 'Mouse Hovering', and 'Mouse Clicked', and a 'Clear' button at the bottom. The 'Events' tab in the Properties window is selected, displaying a table of events and their listener methods.

Event	Listener Method
keyTyped	textFieldKeyTyped
mouseClicked	<none>
mouseDragged	<none>
mouseEntered	textFieldMouseEntered
mouseExited	textFieldMouseExite
mouseMoved	<none>
mousePressed	<none>
mouseReleased	<none>
mouseWheelMoved	<none>
propertyChange	<none>
vetoableChange	<none>

The 'textFieldMouseExite' method is highlighted with a yellow tooltip.

# Example - Mouse movements and clicks

The screenshot shows an IDE with two tabs: 'ListenerProject.java' and 'ListenerDemo.java'. The 'Design' tab is active, displaying a Swing window titled 'Demonstration of Key Listeners and Mouse Listeners'. The window contains three buttons: 'Mouse Movements', 'Mouse Hovering', and 'Mouse Clicked'. A 'Clear' button is located at the bottom. The 'Palette' window on the right lists various Swing components under 'Swing Containers' and 'Swing Controls'. The 'Properties' window for 'jLabel1' is open, showing the 'Events' tab with a list of mouse events and their corresponding listeners.

Property	Value
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

Demonstration of Key Listeners and Mouse Listeners

Hi

Mouse Movements   Mouse Hovering   Mouse Clicked

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

Clear

# Example - Key Pressed, Released and Typed

The screenshot shows an IDE with two tabs: `ListenerProject.java` and `ListenerDemo.java`. The `Design` view is active, showing a window titled "Demonstration of Key Listeners and Mouse Listeners". Inside the window, there is a text field and three buttons: "Mouse Movements", "Mouse Hovering", and "Mouse Clicked". A "Clear" button is at the bottom. The "Palette" on the right lists Swing Containers and Controls. The "Events" tab for the selected text field (`textField [JTextField]`) is open, showing a list of events and their corresponding listener methods. The `keyTyped` event is highlighted with a yellow box, showing the listener `textFieldKeyTyped`.

Event	Listener Method
<code>hierarchyChanged</code>	<none>
<code>inputMethodTextChanged</code>	<none>
<code>keyPressed</code>	<code>textFieldKeyPressed</code>
<code>keyReleased</code>	<code>textFieldKeyReleased</code>
<code>keyTyped</code>	<code>textFieldKeyTyped</code>
<code>mouseClicked</code>	<none>
<code>mouseDragged</code>	<none>
<code>mouseEntered</code>	<code>textFieldMouseEntered</code>
<code>mouseExited</code>	<code>textFieldMouseExited</code>
<code>mouseMoved</code>	<none>
<code>mousePressed</code>	<none>