

MODULE 4

CHAPTER 3 EVENT HANDLING



Prepared By Mr. EBIN PM, AP, IESCE

1

EVENT

- Change in the state of an object is known as event i.e. event describes the change in state of source.
- Events are generated as result of user interaction with the graphical user interface components.
- For example, clicking on a button, moving the mouse, entering a character through keyboard, selecting an item from list, scrolling the page are the activities that causes an event to happen.

Types of Event

The events can be broadly classified into two categories:

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE

2

Foreground Events

- Those events which require the direct interaction of user. They are generated as consequences of a person interacting with the graphical components in Graphical User Interface. For example, clicking on a button, moving the mouse, entering a character through keyboard, selecting an item from list, scrolling the page etc.

Background Events

- Those events that require the interaction of end user are known as background events. Operating system interrupts, hardware or software failure, timer expires, an operation completion are the example of background events.

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

3

EVENT HANDLING

- Event Handling is the mechanism that controls the event and decides what should happen if an event occurs.
- This mechanism have the code which is known as **event handler** that is executed when an event occurs.
- Java Uses the **Delegation Event Model** to handle the events.
- This model defines the standard mechanism to generate and handle the events.
- Let's have a brief introduction to this model.

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

4

The Delegation Event Model has the following key participants namely:

Source - The source is an object on which event occurs. Source is responsible for providing information of the occurred event to its handler. Java provides as with classes for source object.

Listener - It is also known as **event handler**. Listener is responsible for generating response to an event. From java implementation point of view the listener is also an object. Listener waits until it receives an event. Once the event is received, the listener processes the event and then returns.

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

5

- The benefit of this approach is that the user interface logic is completely separated from the logic that generates the event.
- The user interface element is able to delegate the processing of an event to the separate piece of code.
- In this model, listener needs to be registered with the source object so that the listener can receive the event notification.
- This is an efficient way of handling the event because the event notifications are sent only to those listeners that want to receive them.

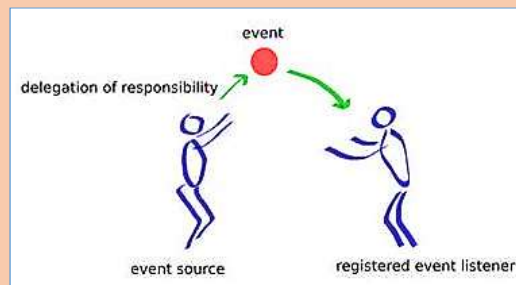
Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

6

How Events are handled

- A source generates an Event and send it to one or more listeners registered with the source.
- Once event is received by the listener, they process the event and then return.
- Events are supported by a number of Java packages, like java.util, java.awt and java.awt.event



Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

7

❖ Event classes and interface

Event Classes	Description	Listener Interface
ActionEvent	generated when button is pressed, menu-item is selected, list-item is double clicked	ActionListener
MouseEvent	generated when mouse is dragged, moved, clicked, pressed or released and also when it enters or exit a component	MouseListener
KeyEvent	generated when input is received from keyboard	KeyListener
ItemEvent	generated when check-box or list item is clicked	ItemListener
TextEvent	generated when value of textarea or textfield is changed	TextListener
MouseWheelEvent	generated when mouse wheel is moved	MouseWheelListener

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

8

WindowEvent	generated when window is activated, deactivated, deiconified, iconified, opened or closed	WindowListener
ComponentEvent	generated when component is hidden, moved, resized or set visible	ComponentEventListener
ContainerEvent	generated when component is added or removed from container	ContainerListener
AdjustmentEvent	generated when scroll bar is manipulated	AdjustmentListener
FocusEvent	generated when component gains or loses keyboard focus	FocusListener

Steps to handle events:

- Implement appropriate interface in the class.
- Register the component with the listener.

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

9

❖ Steps involved in event handling

- The User clicks the button and the event is generated.
- Now the object of concerned event class is created automatically and information about the source and the event get populated with in same object.
- Event object is forwarded to the method of registered listener class.
- The method is now get executed and returns.

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

10

❖ Points to remember about listener

- In order to design a listener class we have to develop some listener interfaces.
- These Listener interfaces forecast some public abstract callback methods which must be implemented by the listener class.
- If we do not implement the predefined interfaces then your class can not act as a listener class for a source object.

Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

11

SOURCES OF EVENT

Event Source	Description
Button	Generates action events when the button is pressed.
Check box	Generates item events when the check box is selected or deselected.
Choice	Generates item events when the choice is changed.
List	Generates action events when an item is double-clicked; generates item events when an item is selected or deselected.
Menu item	Generates action events when a menu item is selected; generates item events when a checkable menu item is selected or deselected.
Scroll bar	Generates adjustment events when the scroll bar is manipulated.
Text components	Generates text events when the user enters a character.
Window	Generates window events when a window is activated, closed, deactivated, deiconified, iconified, opened, or quit.

Prepared By Mr. EBIN PM, AP, IESCE

EDULINE

12

EVENT LISTENER INTERFACES

Interface	Description
ActionListener	Defines one method to receive action events.
AdjustmentListener	Defines one method to receive adjustment events.
ComponentListener	Defines four methods to recognize when a component is hidden, moved, resized, or shown.
ContainerListener	Defines two methods to recognize when a component is added to or removed from a container.
FocusListener	Defines two methods to recognize when a component gains or loses keyboard focus.
ItemListener	Defines one method to recognize when the state of an item changes.
KeyListener	Defines three methods to recognize when a key is pressed, released, or typed.
MouseListener	Defines five methods to recognize when the mouse is clicked, enters a component, exits a component, is pressed, or is released.
MouseMotionListener	Defines two methods to recognize when the mouse is dragged or moved.
MouseWheelListener	Defines one method to recognize when the mouse wheel is moved.
TextListener	Defines one method to recognize when a text value changes.
WindowFocusListener	Defines two methods to recognize when a window gains or loses input focus.
WindowListener	Defines seven methods to recognize when a window is activated, closed, deactivated, deiconified, iconified, opened, or quit.

Prepared By Mr. EBIN PM, AP, IESCE

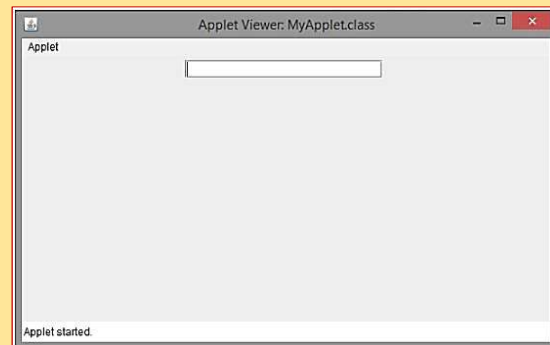
EDULINE

13

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class MyApplet extends JApplet implements KeyListener
{
    JTextField jtf;
    JLabel label;
    public void init()
    {
        setSize(600,300);
        setLayout(new FlowLayout());
        jtf = new JTextField(20);
        add(jtf);
        jtf.addKeyListener(this);
        label = new JLabel();
        add(label);
    }
    public void keyPressed(KeyEvent ke){}
    public void keyReleased(KeyEvent ke){}
    public void keyTyped(KeyEvent ke)
    {
        label.setText(String.valueOf(ke.getKeyChar()));
    }
}
```

Key Event Handling

Initial output of the program

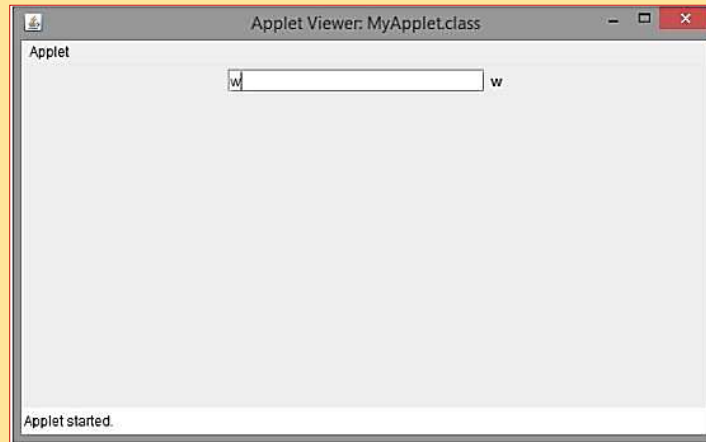


Prepared By Mr. EBIN PM, AP, IESCE

EDULINE

14

After the user enters a character into the text field, the same character is displayed in the label beside the text field as shown in the below image:



Prepared By Mr.EBIN PM, AP, IESCE

EDULINE

15