SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Coimbatore - 641035.

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++" Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

Department of Computer Applications

Course Code: 23CAT606

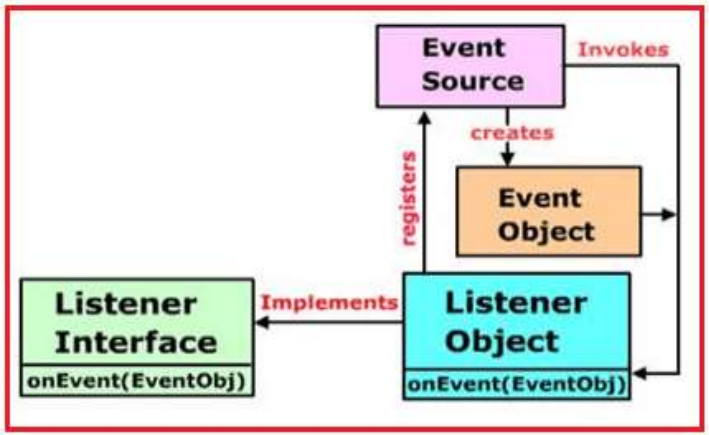
Course Name: Java Programming

Unit II: Package

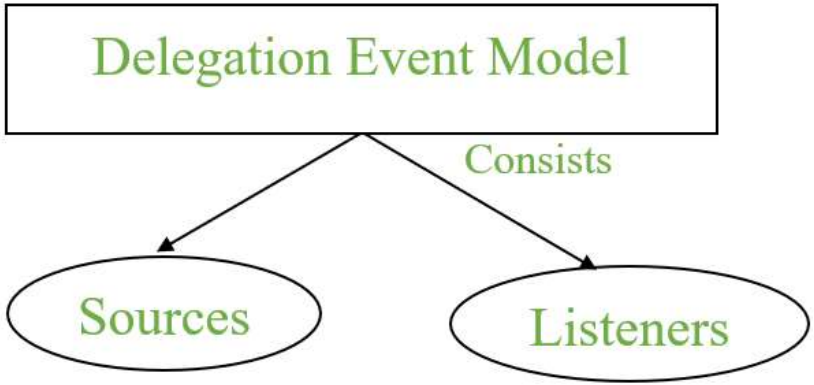
Topic 10: Event Package and Handle

Event Model

The Delegation Event model is defined to handle events in GUI programming languages.



Event Handling is **the mechanism that controls the event and decides what should happen if an event occurs**.



Within class
Other class
Anonymous class

Delegation Event Model

- **Source:** Events are generated from the source. There are various sources like buttons, checkboxes, list, menu-item, choice, scrollbar, text components, windows, etc., to generate events.
- **Listeners:** Listeners are used for handling the events generated from the source. Each of these listeners represents interfaces that are responsible for handling events.

- Two types of Event Handling: Foreground and Bacground

Registering the Source With Listener

addTypeListener()

Example 1: For KeyEvent we use addKeyListener() to register.

Example 2:that For ActionEvent we use addActionListener() to register.

EVENTS	SOURCE	LISTENERS
Action Event	Button, List,Menultem, Text field	ActionListener
Component Event	Component	Component Listener
Focus Event	Component	FocusListener
Item Event	Checkbox,CheckboxMenultem, Choice, List	ItemListener
Key Event	when input is received from keyboard	KeyListener
Text Event	Text Component	TextListener
Window Event	Window	WindowListener
Mouse Event	Mouse related event	MouseListener

Steps to Handle Event in Java

1. Whenever the user clicks the button an event is generated.
2. Now the object of the concerned event class will be automatically created and information about the source and the event gets populated within the same object.
3. Then the event object is forwarded to the method of the registered listener class.
4. Now the method will get executed and returned.

Registration Methods

1.	**Button** :	public void addActionListener(ActionListener a){}
2.	**MenuItem:**	public void addActionListener(ActionListener a){}
3.	**TextField** :	public void addActionListener(ActionListener a){}
		public void addTextListener(TextListener a){}
4.	**TextArea** :	public void addTextListener(TextListener a){}
5.	**Checkbox** :	public void addItemListener(ItemListener a){}
6.	**Choice** :	public void addItemListener(ItemListener a){}
7.	**List** :	public void addActionListener(ActionListener a){}
		public void addItemListener(ItemListener a){}

Simple Example of Event Handling in java

```
import java.awt.*;
import java.awt.event.*;
class EventHandling extends Frame implements ActionListener
{
EventHandling ()
{
TextField  textField = new TextField ();
textField.setBounds (60, 50, 170, 20);
Button button = new Button ("Show");
button.setBounds (90, 140, 75, 40);
                button.addActionListener (this);
add (button);
add (textField);
setSize (250, 250);
setLayout (null);
setVisible (true);
}
public void actionPerformed (ActionEvent e)
{
                textField.setText ("Hello World");
}
public static void main (String args[])
{
                new EventHandling ();
}
}
```

Summary

Reference

1. Herbert Schildt “ The Complete Reference Java 2, 8th edition , Tata McGraw Hill, 2011
2. Ralph Bravaco, Shai Simonson, “Java Programming: From the Ground up Tata McGraw Hill, 2012
3. https://www.javatpoint.com