Introduction to Event Handling in JAVA

BY

M. BABY ANUSHA,

ASST.PROF IN CSE DEPT.,

RGUKT,NUZVID

What is an 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 Events :

The events can be broadly classified into two categories:

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.

Types of Events :

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.

What is Event Handling?

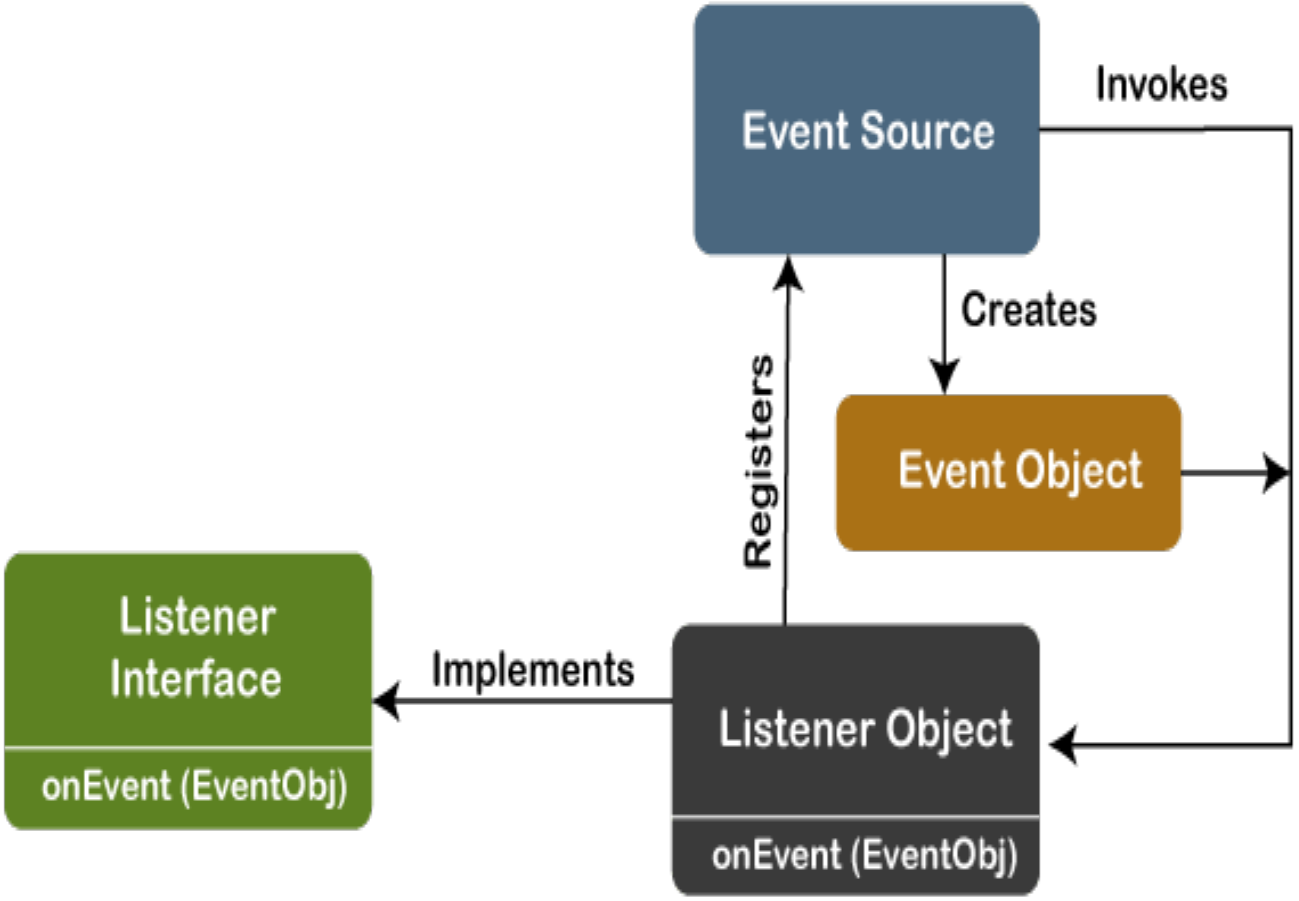
- Event Handling is the mechanism that controls the event and decides what should happen if an event occurs.
- This mechanism have the code which is known as event handler that is executed when an event occurs.
- Java Uses the Delegation Event Model to handle the events.
- This model defines the standard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

Delegation Event Model :

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 it's handler. Java provide 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 process the event an then returns.

Delegation Event Model :



Event and Listener : (Java Event Handling)

- Changing the state of an object is known as an event. For example, click on button, dragging mouse etc.
- The java.awt.event package provides many event classes and Listener interfaces for event handling.

Java Event classes, Event Sources and Listener interfaces :

EVENTS	SOURCE	LISTENERS
Action Event	Button, List, MenuItem, Text field	ActionListener
Component Event	Component	Component Listener
Focus Event	Component	FocusListener
Item Event	Checkbox, CheckboxMenuItem, 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 perform Event Handling

Following steps are required to perform event handling:

- Register the component with the Listener

Registration Methods

- For registering the component with the Listener, many classes provide the registration methods. For example:

Button

- public void addActionListener(ActionListener a){}

Steps to perform Event Handling

TextField

- public void addActionListener(ActionListener a){}
- public void addTextListener(TextListener a){}

TextArea

- public void addTextListener(TextListener a){}

Checkbox

- public void addItemListener(ItemListener a){}

Steps to perform Event Handling

MenuItem

- public void addActionListener(ActionListener a){}

Choice

- public void addItemListener(ItemListener a){}

List

- public void addActionListener(ActionListener a){}
- public void addItemListener(ItemListener a){}

Java Event Handling : Example

```
import java.awt.*;
import java.awt.event.*;
class AEvent extends Frame implements ActionListener
{
TextField tf;
AEvent()
{
  //create components
tf=new TextField();
tf.setBounds(60,50,170,20);
Button b=new Button("click me");
b.setBounds(100,120,80,30);
b.addActionListener(this);//passing current instance
```

Java Event Handling : Example

```
//add components and set size, layout and visibility
add(b);add(tf);
setSize(300,300);
setLayout(null);
setVisible(true);
}
public void actionPerformed(ActionEvent e){
tf.setText("Welcome");
}
public static void main(String args[]){
new AEvent();
}
}
```

Java Event Handling : Example

Output :


Thank you!
shutterstock.com · 567687052