



Package: `java.awt.event`.

By,

Hitha Paulson

Assistant Professor, Dept. of Computer Science

LF College, Guruvayoor

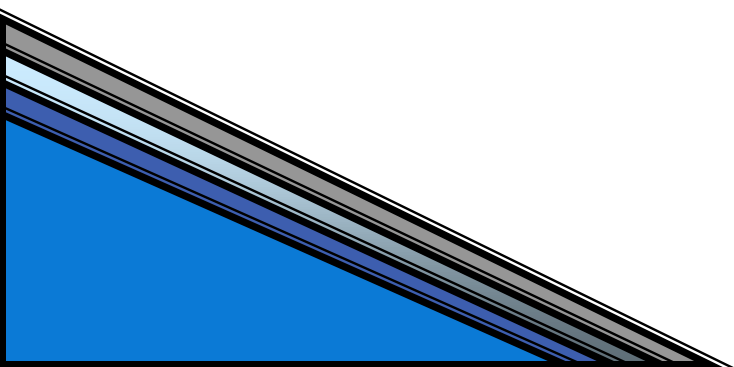
The Delegation Event Model

- ✓ Standard and consistent mechanism to generate and process events
- ✓ Concept
 - Source generates events and send to one or more listeners
 - Listener waits for an event to occur
 - When event happens, listener process the event and returns
 - Hint: Listener must register in order to receive an event notification
- ✓ A user interface element is able to “delegate” the processing of an event to a separate piece of code

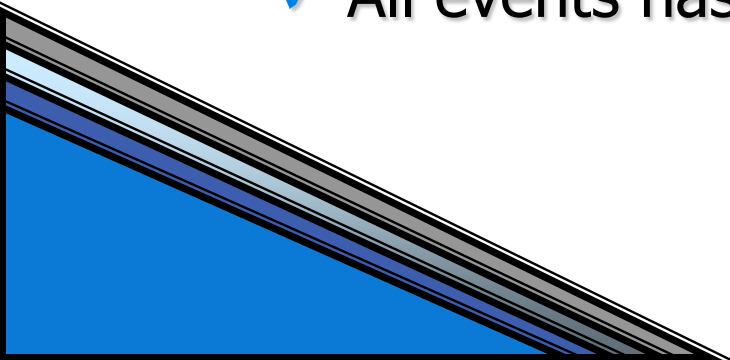


Events

- ✓ An event is an object that describes a state change in a source
- ✓ Events are generated
 - As a consequence of a user interacting with GUI element (Eg: Button Click, Entering a character)
 - As a consequence of a previous event (Eg: Lost focus and Got focus)
 - Event are also generated without any user interaction (Eg: Timer expires, counter exceeds)

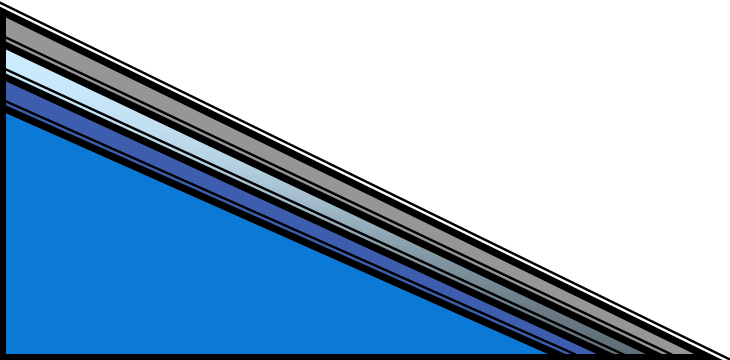


Event Sources

- ✓ A source is an object that generates an event
 - ✓ A source may generate more than one type event
 - ✓ Source must register listeners in order for listener to receive notification about the event
 - ✓ When an event occurs, all registered listeners are notified and receive a copy of the event object, and is known as *multi-casting* the event
 - ✓ General Form:
 - `public void add TypeListener(TypeListener obj)`
 - ✓ Listener can remove from source
 - Eg: `button1.removeMouseListener(this)`
 - ✓ All events has its own registration method
- 

Event Listeners

- ✓ A listener is an object that is notified when an event occurs
- ✓ An event listener must
 - Registered with one or more sources to receive notification
 - Implement methods to receive and process notifications
 - These methods are defined as interfaces and found in `java.awt.event` package



Mouse and Keyboard events

