

Event Handling

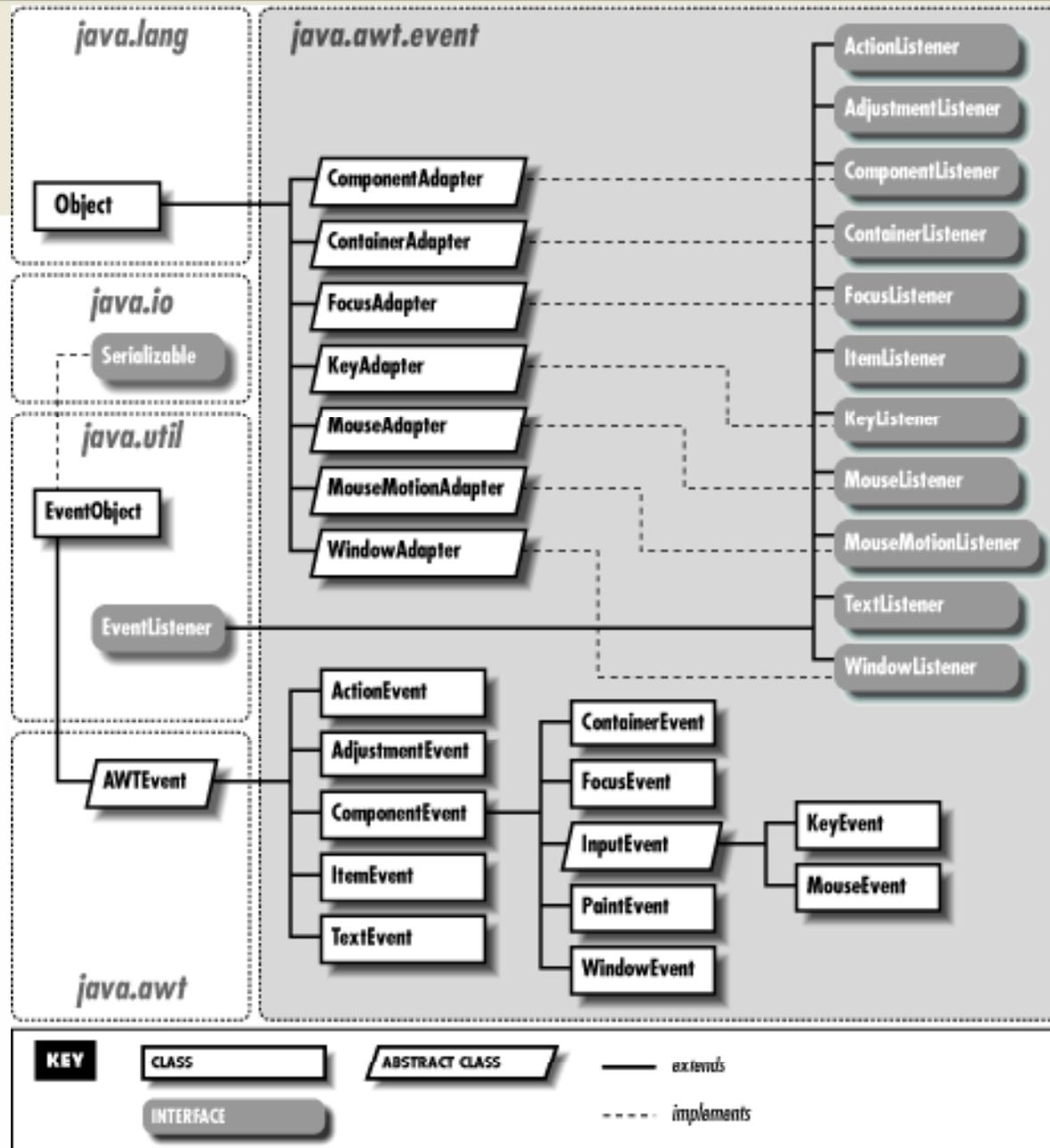
EVENT HANDLING

➤ GUIs are *event driven*

✦ Generate *events* when user interacts with GUI

➤ e.g., moving mouse, pressing button, typing in text field, etc.

➤ Class `java.awt.AWTEvent`



EVENT HANDLING

➤ Event-handling model

✦ Three parts

➤ Event source

- ⊕ GUI component with which user interacts

➤ Event object

- ⊕ Encapsulates information about event that occurred

➤ Event listener

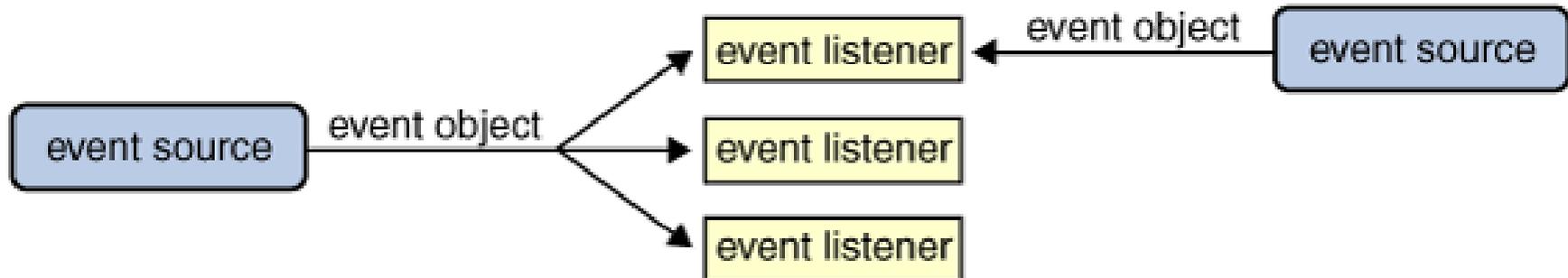
- ⊕ Receives event object when notified, then responds

✦ Programmer must perform two tasks

➤ Register event listener for event source

➤ Implement event-handling method (event handler)

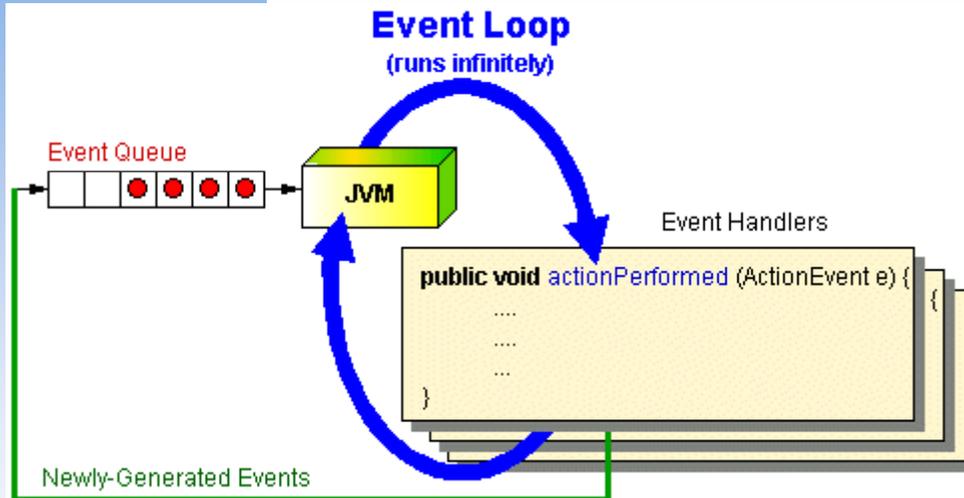
EVENT LISTENERS & SOURCES



WRITING EVENT LISTENERS

- Design Considerations
- รายละเอียดของ event: Event Objects
- Concepts: Low-Level Events and Semantic Events
- Event Adapters
- Inner Classes and Anonymous Inner Classes

DESIGN CONSIDERATIONS

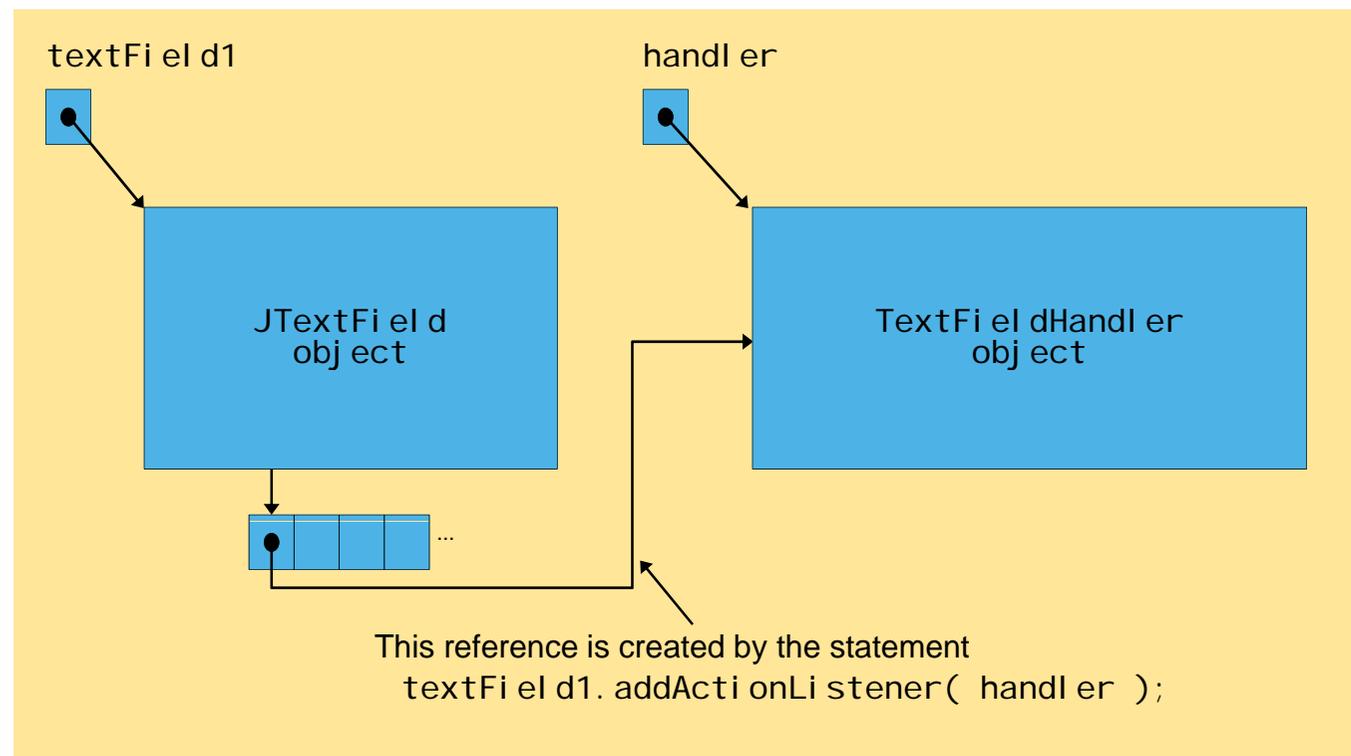


- ทำงานเร็ว ๆ
- สร้าง thread ใหม่
- implement อย่างไร?
 - ✦ separate classes
 - ✦ self handling
 - ✦ private vs. public

รายละเอียดของ EVENT: EVENT OBJECTS

Object getSource()

Returns the object that fired the event.



CONCEPTS: LOW-LEVEL EVENTS AND SEMANTIC EVENTS

➤ Low level events

- ✦ window-system occurrences
- ✦ low-level input
- ✦ เช่น mouse, key

➤ Semantic events

- ✦ อื่น ๆ เช่น action, item
- ✦ บาง semantic event อาจเกิดจาก user input
- ✦ บาง semantic event ไม่ได้เกิดจาก user input

➤ ควรจะสนใจ semantic event ก่อน low-level

EVENT ADAPTER

- การจัดการ **event** ต้องทำการ **implement Event Listener** (เป็น **interface**) ชนิดต่างๆ
- **interface: abstract methods**
- ต้องเขียน **method** ใหม่ (**override**) ทั้งหมด
- บาง **listener interface** มีหลาย **method** แต่ต้องการที่จะ **override** เพียง **1** หรือ **2** ทำให้ต้องเขียนเยอะ
- **EventAdapter** ก็จะเป็นคลาสที่ได้ทำการ **implement** เมทอดทั้งหมดใน **EventListener** นั้น ๆ ไว้หมดแล้ว
- **override** เฉพาะ

MyClass

implements MouseListener

```
//An example that implements a listener interface directly.
public class MyClass implements MouseListener {
    ...
    someObject.addMouseListener(this);
    ...

    /* Empty method definition. */
    public void mousePressed(MouseEvent e) { }
    /* Empty method definition. */
    public void mouseReleased(MouseEvent e) { }
    /* Empty method definition. */
    public void mouseEntered(MouseEvent e) { }
    /* Empty method definition. */
    public void mouseExited(MouseEvent e) { }

    public void mouseClicked(MouseEvent e) {
        ...//Event listener implementation goes here...
    }
}
```

MOUSEADAPTER

```
public class MouseAdapter
    implements MouseListener {

    public void mousePressed(MouseEvent e) { }
    public void mouseReleased(MouseEvent e) { }
    public void mouseEntered(MouseEvent e) { }
    public void mouseExited(MouseEvent e) { }
    public void mouseClicked(MouseEvent e) { }

}
```

MyClass

extends MouseAdapter

```
/*  
 * An example of extending an adapter class instead of  
 * directly implementing a listener interface.  
 */  
public class MyClass extends MouseAdapter {  
    ...    someObject.addMouseListener(this);  
    ...  
  
    public void mouseClicked(MouseEvent e) {  
        ...//Event listener implementation goes here...  
    }  
}
```

ตัวอย่าง EVENT และ EVENT LISTENERS ที่เกี่ยวข้อง

Act that Results in the Event	Listener Type
กดปุ่ม, กด Enter ในขณะที่พิมพ์ใน text field, หรือเลือกรายการในเมนู	ActionListener
ผู้ใช้ปิดเฟรม (main window)	WindowListener
ผู้ใช้กดปุ่มเมาส์ ในขณะที่อยู่บน component	MouseListener
ผู้ใช้เคลื่อนเมาส์ไปอยู่บน component	MouseMotionListener
Component มองเห็น	ComponentListener
Component ได้รับโฟกัส	FocusListener
ตาราง หรือ ลิสต์ มีการเปลี่ยนแปลง	ListSelectionListener
property บางอย่างใน component เปลี่ยน เช่น ข้อความของ label	PropertyChangeListener

REGISTER EVENT LISTENER FOR EVENT SOURCE

component.addXYZListener(an instance of
XYZListener)

where XYZ is the type of event e.g.

```
// MyButtonListener & MyWindowListener are defined  
// somewhere
```

```
aButton.addActionListener(new MyButtonListener());  
aFrame.addWindowListener(new MyWindowListener());
```

```
aButton.addActionListener(new ActionListener() {  
    // anonymous inner class  
    public void actionPerformed(ActionEvent e) {  
        // code to be executed  
    }  
});
```

INNER CLASS

```
//An example of using an inner class. public
class MyClass extends Applet {
    ...
    someObject.addMouseListener(new MyAdapter());
    ...

    class MyAdapter extends MouseAdapter {
        public void mouseClicked(MouseEvent e) {
            ...
            //Event listener implementation goes here
            ...
        }
    }
}
```

ขอดี คลาสที่สร้างไม่ต้อง extends adapter
ทำให้สามารถ extend คลาสอื่นที่ต้องการได้

ANONYMOUS INNER CLASS:

ไม่ต้องมีชื่อคลาส

```
//An example of using an anonymous inner class.
public class MyClass extends Applet {
    ...
    someObject.addMouseListener(new MouseAdapter() {
        public void mouseClicked(MouseEvent e) {
            ...
            //Event listener implementation goes here
            ...
        }
    });
    ...
}
```

ข้อดี ส่วนของโปรแกรมที่จัดการ **event** อยู่ใกล้กับ **object** ที่เป็น ต้นกำเนิด

COMMONLY USED EVENT LISTENERS

➤ ActionListener

- ✦ Handles buttons and a few other actions
 - `actionPerformed(ActionEvent event)`

➤ CaretListener

- ✦ when caret (cursor position in a text component) changes
 - `caretUpdate(CaretEvent event)`

➤ ChangeListener

- ✦ when the object registered changes
 - `stateChanged(ChangeEvent event)`

COMMONLY USED EVENT LISTENERS

➤ ComponentListener

✦ Handles moving/resizing/hiding GUI objects

- componentResized(ComponentEvent event)
- componentMoved(ComponentEvent event)
- componentShown(ComponentEvent event)
- componentHidden(ComponentEvent event)

➤ ContainerListener

✦ when component added or remove from a container

- componentAdded(ContainerEvent event)
- componentRemoved(ContainerEvent event)

COMMONLY USED EVENT LISTENERS

➤ DocumentListener

- when the content of the document changes
 - `changeUpdate(DocumentEvent event)`
 - `insertUpdate(DocumentEvent event)`
 - `removeUpdate(DocumentEvent event)`

➤ FocusListener

- Detects when controls get/lose keyboard focus
 - `focusGained(FocusEvent event)`
 - `focusLost(FocusEvent event)`

➤ ItemListener

- Handles selections in lists, checkboxes, etc.
 - `itemStateChanged(ItemEvent event)`

COMMONLY USED EVENT LISTENERS

➤ KeyListener (KeyAdapter)

- ✦ when user is typing at keyboard
 - keyPressed(KeyEvent event)
 - keyReleased(KeyEvent event)
 - keyTyped(KeyEvent event)

➤ ListDataListener

- ✦ when the content of the list changes
 - intervalAdded(ListDataEvent event)
 - intervalRemoved(ListDataEvent event)
 - contentChanged(ListDataEvent event)

COMMONLY USED EVENT LISTENERS

➤ ListSelectionListener

- ✦ when selection is changed

- valueChanged(ListSelectionEvent event)

➤ MouseListener (MouseListener/MouseInputAdapter)

- ✦ when mouse is used

- mouseClicked(MouseEvent event)

- mouseEntered(MouseEvent event)

- mouseExited(MouseEvent event)

- mousePressed(MouseEvent event)

- mouseReleased(MouseEvent event)

COMMONLY USED EVENT LISTENERS

➤ MouseMotionListener

✦ Handles mouse movement

- mouseDragged(MouseMotionEvent event)
- mouseMoved(MouseMotionEvent event)

➤ MouseWheelListener

✦ when mouse-wheel is rotated

- mouseWheelMoved(MouseWheelEvent event)

➤ PropertyChangeListener

- propertyChange(PropertyChangeEvent event)

COMMONLY USED EVENT LISTENERS

➤ WindowListener

- `windowOpened(WindowEvent event)`
- `windowClosing(WindowEvent event)`
- `windowClosed(WindowEvent event)`
- `windowIconified(WindowEvent event)`
- `windowDeiconified(WindowEvent event)`
- `windowActivated(WindowEvent event)`
- `windowDeactivated(WindowEvent event)`

➤ WindowFocusListener

- `windowGainedFocus(WindowEvent event)`
- `windowLostFocus(WindowEvent event)`

➤ WindowStateListener

- `windowStateChanged(WindowEvent event)`

ACTION LISTENER

- action event เกิดเมื่อผู้ใช้ ทำ action กับ componet เช่น กดปุ่ม, เลือกเมนู, กด Enter ใน textfield
- จะส่ง actionPerformed ไปให้กับ listener ที่รออยู่
- ตัวอย่าง
- <http://java.sun.com/docs/books/tutorial/uiswing/examples/events/index.html>

KEY LISTENER

➤ เมื่อ keyboard ถูกไฟกัส

➤ ส่งข้อมูล

✦ unicode (key-typed event)

✦ กด หรือ ปล่อย ปุ่ม (key-pressed/key-released)

STANDARD AWT EVENT LISTENERS

Listener	Adapter class (if any)	Registration method
ActionListener		addActionListener
AdjustmentListener		addAdjustmentListener
ComponentListener	ComponentAdapter	addComponentListener
ContainerListener	ContainerAdapter	addContainerListener
FocusListener	FocusAdapter	addFocusListener
ItemListener		addItemListener
KeyListener	KeyAdapter	addKeyListener
MouseListener	MouseAdapter	addMouseListener
MouseMotionListener	MouseAdapter	addMouseMotionListener
TextListener		addTextListener
WindowListener	WindowAdapter	addWindowListener

REFERENCE

➤ <http://java.sun.com/docs/books/tutorial/uiswing/events/index.html>

➤ ตัวอย่าง

➤ <http://java.sun.com/docs/books/tutorial/uiswing/examples/events/index.html>