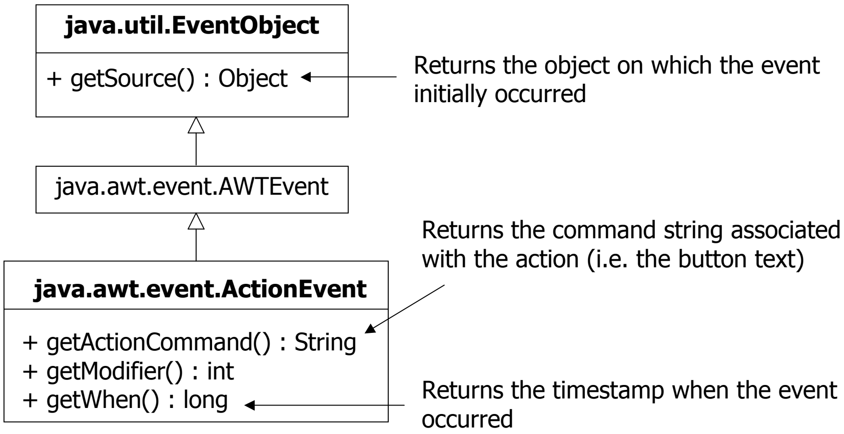
EventObject

- The **event object** contains information pertinent to the event type



java.util.EventObject

+ getSource() : Object

Returns the object on which the event initially occurred

java.awt.event.AWTEvent

java.awt.event.ActionEvent

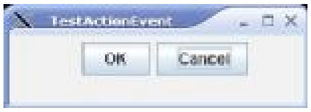
+ getActionCommand() : String
+ getModifier() : int
+ getWhen() : long

Returns the command string associated with the action (i.e. the button text)

Returns the timestamp when the event occurred

Action Events

- Write a program that displays two buttons, `OK` and `Cancel`, in the window. A message is displayed on the console to indicate which button was pressed and when.



```
% java TestActionEvent
The OK button was clicked at
 Mon Jan 24 20:13:58 EST 2005
The Cancel button was clicked at
 Mon Jan 24 20:14:03 EST 2005
```

- /usr/local/pub/sps/courses/cs2/events/ActionEvent

Window Events

- Write a program that demonstrates handling of window events:
  - Window opened
  - Window closing/closed
  - Window activated
  - Window deactivated
  - Window iconified
  - Window deiconified

- /usr/local/pub/sps/courses/cs2/events/WindowEvent

WindowAdapter

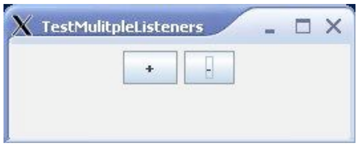
- The WindowAdapter class is a class that implements the WindowListener interface
  - The methods in this class are empty.
- To use the WindowAdapter class:
  - Extend this class to create a WindowEvent listener
  - Override the methods for the events of interest
  - Create a listener object using the extended class and then register it with a Window using the window's addWindowListener() method.

The Result

```
import javax.swing.*;
Import java.awt.event.*;

public class SwingFrame {
    public static void main( String args[] ) {
        JFrame win = new JFrame( "My First GUI Program" );

        win.addWindowListener(
            new WindowAdapter() {
                public void windowClosing( WindowEvent e ) {
                    System.exit ( 0 );
                }
            }
         );

        win.setSize( 250, 150 );
        win.setVisible(true);
    }
} // SwingFrame
```

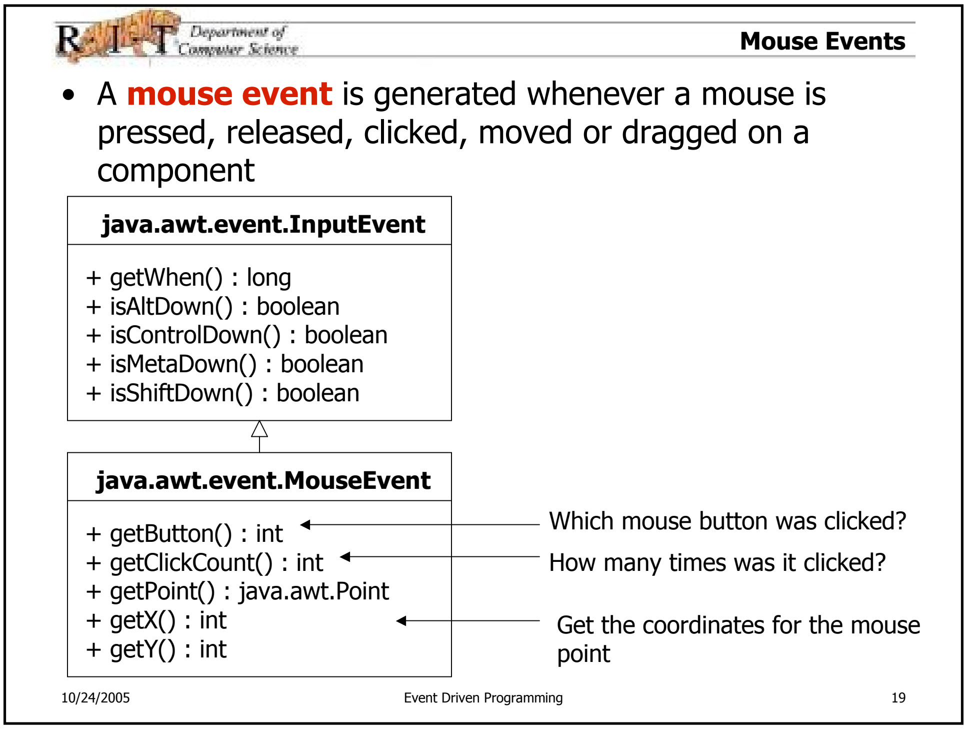
Multiple Listeners

- Write a program which changes values by 1’s and 2’s using multiple listeners



```
Listener 1 created: 0
Listener 2 created: 0
Listener 2 inc: 2
Listener 1 inc: 1
Listener 2 inc: 4
Listener 1 inc: 2
Listener 2 dec: 2
Listener 1 dec: 1
Listener 2 inc: 4
Listener 1 inc: 2
```

- /usr/local/pub/sps/courses/cs2/events/MultipleListeners



RIT Department of Computer Science

Mouse Events

- A **mouse event** is generated whenever a mouse is pressed, released, clicked, moved or dragged on a component

java.awt.event.InputEvent

+ getWhen() : long
+ isAltDown() : boolean
+ isControlDown() : boolean
+ isMetaDown() : boolean
+ isShiftDown() : boolean

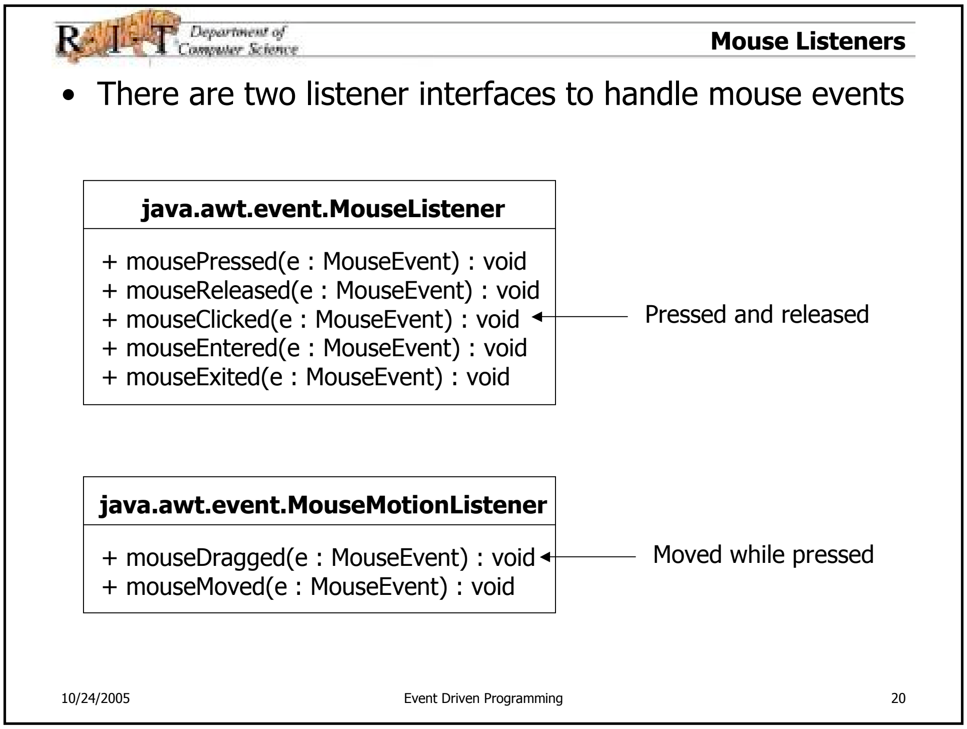
java.awt.event.MouseEvent

+ getButton() : int
+ getClickCount() : int
+ getPoint() : java.awt.Point
+ getX() : int
+ getY() : int

Which mouse button was clicked?

How many times was it clicked?

Get the coordinates for the mouse point



RIT Department of Computer Science

Mouse Listeners

- There are two listener interfaces to handle mouse events

java.awt.event.MouseListener

+ mousePressed(e : MouseEvent) : void
+ mouseReleased(e : MouseEvent) : void
+ mouseClicked(e : MouseEvent) : void
+ mouseEntered(e : MouseEvent) : void
+ mouseExited(e : MouseEvent) : void

Pressed and released

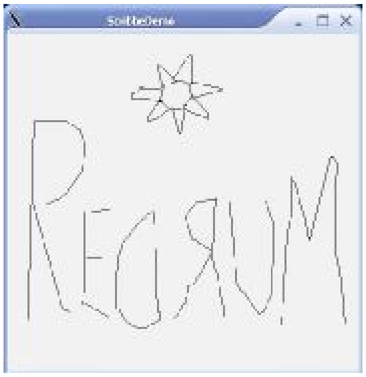
java.awt.event.MouseMotionListener

+ mouseDragged(e : MouseEvent) : void
+ mouseMoved(e : MouseEvent) : void

Moved while pressed

ScribbleDemo

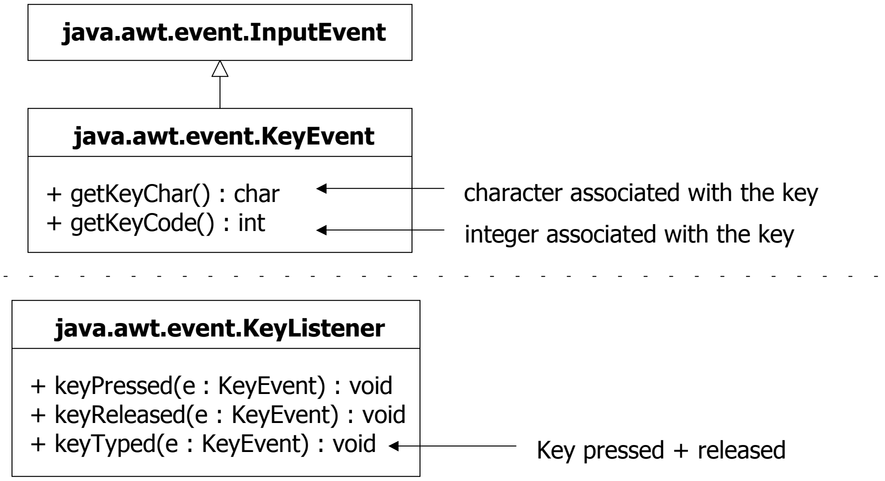
- Write a program which uses the mouse for scribbling. You can draw with the left mouse button and erase with the right mouse button



- /usr/local/pub/sps/courses/cs2/events/ScribbleDemo

Keyboard Events

- A **key event** is generated whenever a key is pressed, released, or typed on a component



java.awt.event.InputEvent

java.awt.event.KeyEvent

+ getKeyChar() : char
+ getKeyCode() : int

character associated with the key

integer associated with the key

java.awt.event.KeyListener

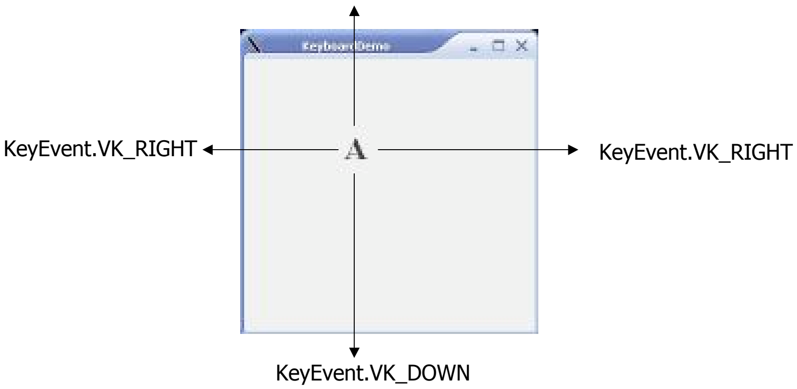
+ keyPressed(e : KeyEvent) : void
+ keyReleased(e : KeyEvent) : void
+ keyTyped(e : KeyEvent) : void

Key pressed + released

Department of Computer Science

Keyboard Demo

- Write a program that displays a user input character which can be moved around with the arrow keys and changed



KeyEvent.VK_RIGHT

KeyEvent.VK_RIGHT

KeyEvent.VK_RIGHT

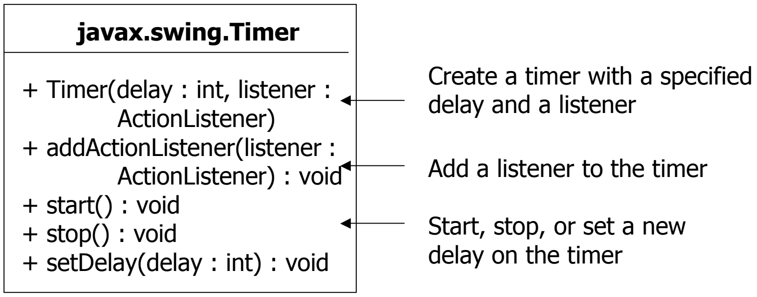
KeyEvent.VK_DOWN

- /usr/local/pub/sps/courses/cs2/events/KeyboardDemo

Department of Computer Science

Timers

- A **timer** is a source object which can trigger an `ActionEvent` at a predefined rate
  - It's not a visible GUI component



javax.swing.Timer

+ Timer(delay : int, listener : ActionListener)
+ addActionListener(listener : ActionListener) : void
+ start() : void
+ stop() : void
+ setDelay(delay : int) : void

Create a timer with a specified delay and a listener

Add a listener to the timer

Start, stop, or set a new delay on the timer