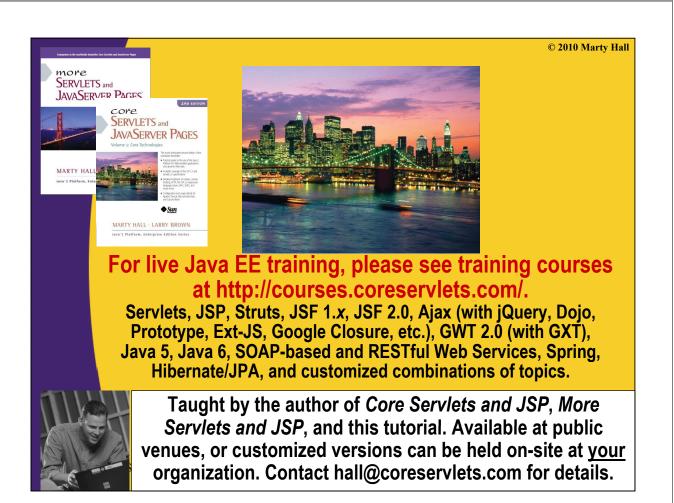
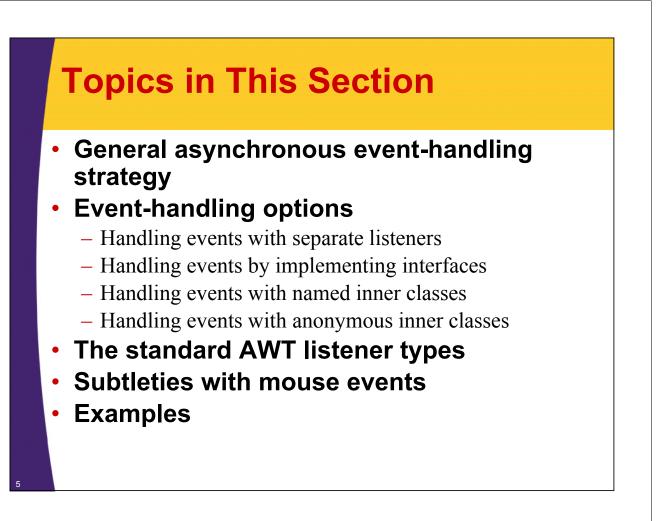


Handling Mouse and Keyboard Events Originals of Slides and Source Code for Examples:

Originals of Slides and Source Code for Examples: http://courses.coreservlets.com/Course-Materials/java5.html

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.





General Strategy

Determine what type of listener is of interest

- 11 standard AWT listener types, described on later slide.
 - ActionListener, AdjustmentListener, ComponentListener, ContainerListener, FocusListener, ItemListener, KeyListener, MouseListener, MouseMotionListener, TextListener, WindowListener

Define a class of that type

- Implement interface (KeyListener, MouseListener, etc.)
- Extend class (KeyAdapter, MouseAdapter, etc.)
- Register an object of your listener class with the window
 - w.addXxxListener(new MyListenerClass());
 - E.g., addKeyListener, addMouseListener



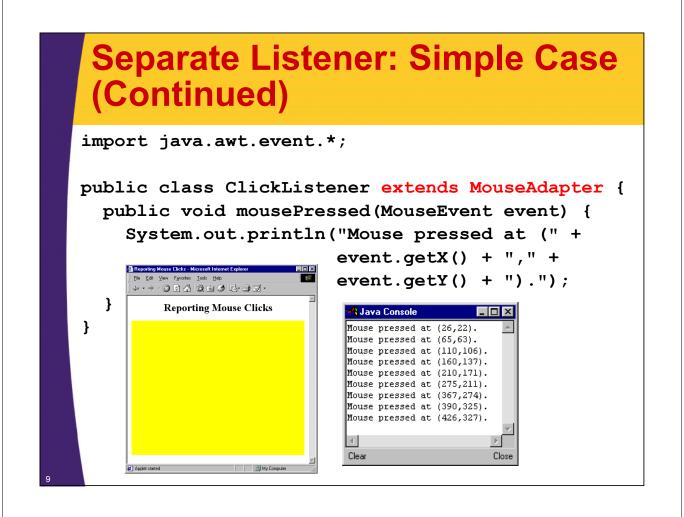
Using Separate Listener Classes

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Handling Events with a Separate Listener: Simple Case

 Listener does not need to call any methods of the window to which it is attached

```
import java.applet.Applet;
import java.awt.*;
public class ClickReporter extends Applet {
   public void init() {
      setBackground(Color.YELLOW);
      addMouseListener(new ClickListener());
   }
}
```



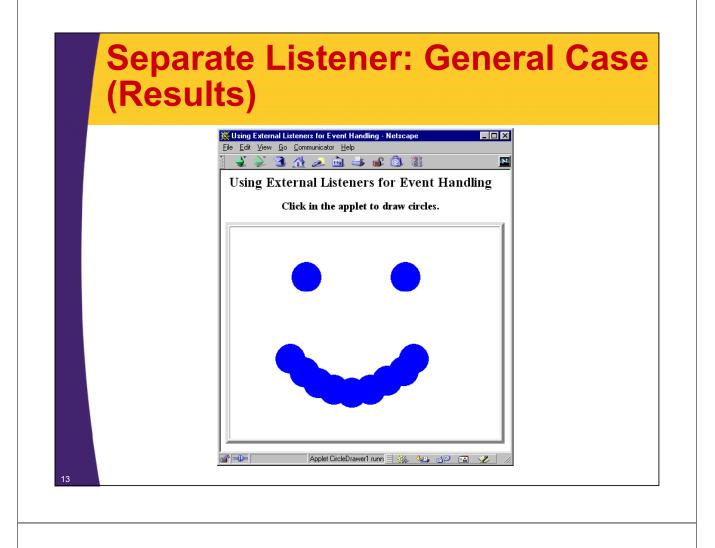
Generalizing Simple Case

- What if ClickListener wants to draw a circle wherever mouse is clicked?
- Why can't it just call getGraphics to get a Graphics object with which to draw?
- General solution:
 - Call event.getSource to obtain a reference to window or GUI component from which event originated
 - Cast result to type of interest
 - Call methods on that reference

Handling Events with Separate Listener: General Case

```
import java.applet.Applet;
import java.awt.*;
public class CircleDrawer1 extends Applet {
    public void init() {
        setForeground(Color.BLUE);
        addMouseListener(new CircleListener());
    }
}
```

Separate Listener: General Case (Continued)

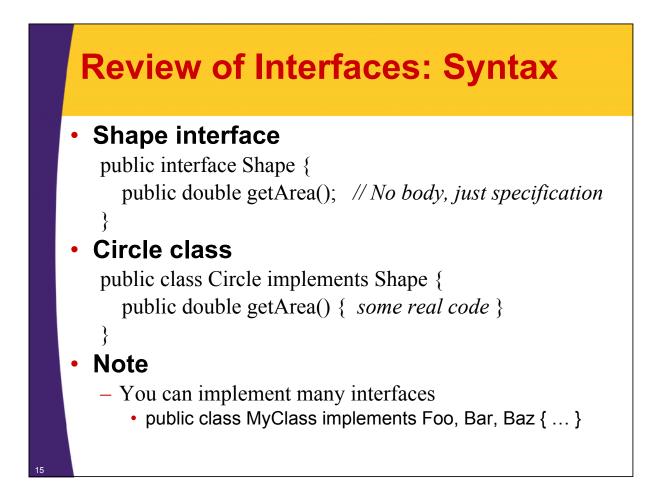


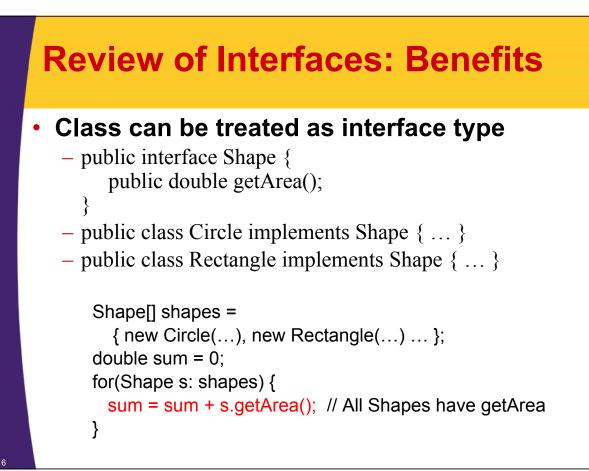


Implementing a Listener Interface

Customized Java EE Training: http://courses.coreservlets.com/

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



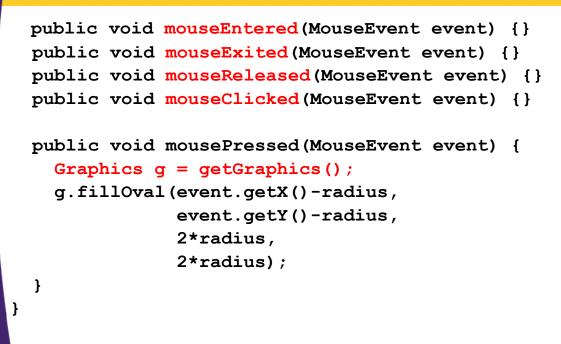


Source Code for MouseListener and MouseAdapter (Simplified)

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

Case 2: Implementing a Listener Interface

Implementing a Listener Interface (Continued)



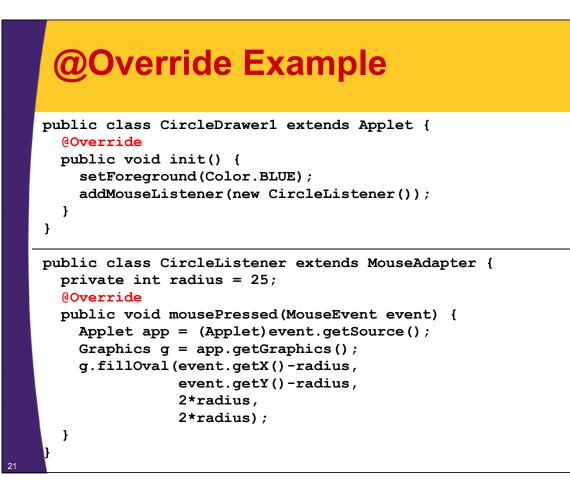
Adapters vs. Interfaces: Method Signature Errors

• What if you goof on the method signature?

- public void mousepressed(MouseEvent e)
- public void mousePressed()

Interfaces

- Compile time error
- Adapters
 - No compile time error, but nothing happens at run time when you press the mouse
- Solution for adapters (and overriding in Java 5+ in general): @Override annotation
 - Whenever you *think* you are overriding a method, put "@Override" on the line above the start of the method.
 - If that method is not actually overriding an inherited method, you get a compile-time error.

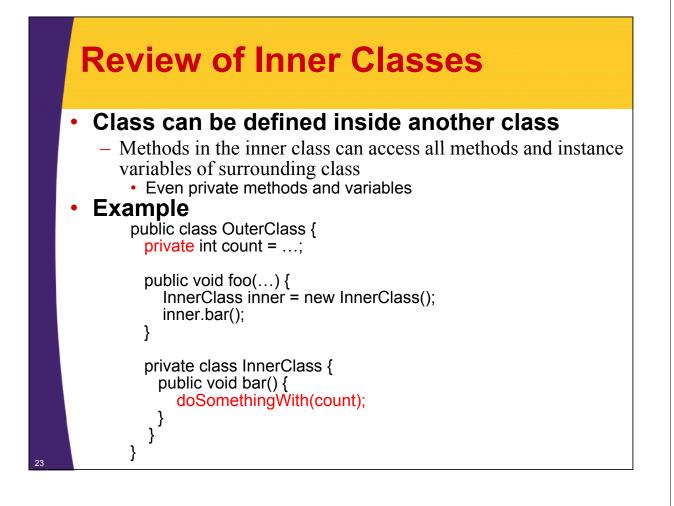




Using Inner Classes (Named & Anonymous)

Customized Java EE Training: http://courses.coreservlets.com/

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

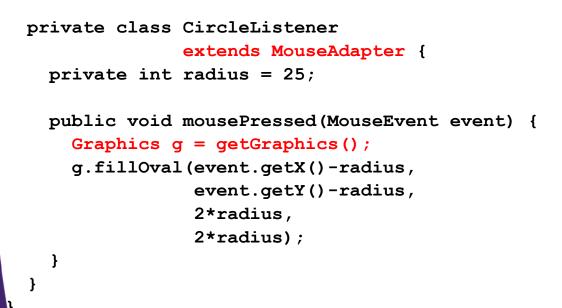


Case 3: Named Inner Classes

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class CircleDrawer3 extends Applet {
   public void init() {
     setForeground(Color.BLUE);
     addMouseListener(new CircleListener());
   }
```

Named Inner Classes (Continued)

Note: still part of class from previous slide



Case 4: Anonymous Inner Classes



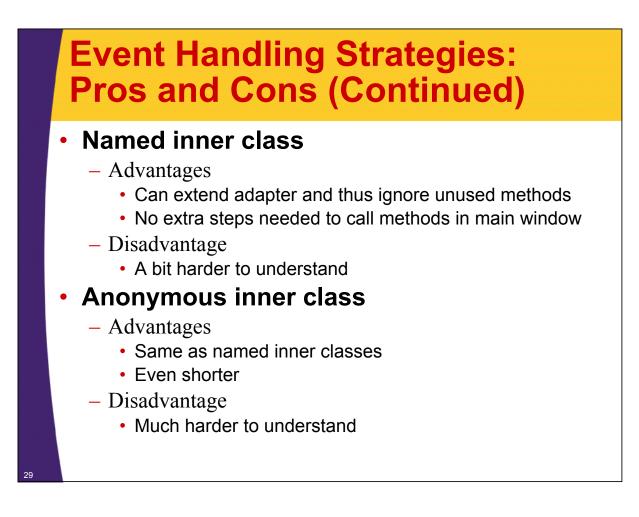
Summary of Approaches

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Event Handling Strategies: Pros and Cons

Separate Listener

- Advantages
 - Can extend adapter and thus ignore unused methods
 - · Separate class easier to manage
- Disadvantage
 - · Need extra step to call methods in main window
- Main window that implements interface
 - Advantage
 - · No extra steps needed to call methods in main window
 - Disadvantage
 - Must implement methods you might not care about





Event Handler Details and Examples

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Standard AWT Event Listeners (Summary)

| | Adapter Class | |
|---------------------|---------------------------|-------------------------------|
| Listener | (If Any) | Registration Method |
| ActionListener | | addActionListener |
| AdjustmentListener | | addAdjus <i>tmentListener</i> |
| ComponentListener | ComponentAdapter | addComponentListener |
| ContainerListener | ContainerAdapter | addContaine rListe ne r |
| FocusListener | FocusAdapter | addFocus Liste ne r |
| ItemListener | | addIte mListe ne r |
| KeyListener | Ke yAdapte r | addKeyListener |
| MouseListener | MouseAdapter | addMous e Lis te ne r |
| MouseMotionListener | MouseMotionAdapter | addMouseMotionListener |
| TextListener | | addTextListener |
| WindowListener | WindowAdapter | addWindowListener |

Standard AWT Event Listeners (Details)

ActionListener

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

AdjustmentListener

- Applies to scrolling
 - adjustmentValueChanged(AdjustmentEvent event)

ComponentListener

- Handles moving/resizing/hiding GUI objects
 - componentResized(ComponentEvent event)
 - componentMoved (ComponentEvent event)
 - componentShown(ComponentEvent event)
 - componentHidden(ComponentEvent event)

Standard AWT Event Listeners (Details Continued)

ContainerListener

- Triggered when window adds/removes GUI controls
 - componentAdded(ContainerEvent event)
 - componentRemoved(ContainerEvent event)

FocusListener

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

Standard AWT Event Listeners (Details Continued)

ItemListener

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

KeyListener

- Detects keyboard events
 - keyPressed(KeyEvent event) -- any key pressed down
 - keyReleased(KeyEvent event) -- any key released
 - keyTyped(KeyEvent event) -- key for printable char released

Standard AWT Event Listeners (Details Continued)

MouseListener

- Applies to basic mouse events
 - mouseEntered(MouseEvent event)
 - mouseExited(MouseEvent event)
 - mousePressed(MouseEvent event)
 - mouseReleased(MouseEvent event)
 - mouseClicked(MouseEvent event)
 - Release without drag. Do *not* use this for mousePressed!
 Applies on release if no movement since press

MouseMotionListener

- Handles mouse movement
 - mouseMoved(MouseEvent event)
 - mouseDragged(MouseEvent event)

MouseInputListener

- Combines MouseListener and MouseMotionListener
 - In javax.swing.event package, not java.awt.event
 - You have to call both addMouseListener and addMouseMotionListener, so it does not save much

Standard AWT Event Listeners (Details Continued)

TextListener

- Applies to textfields and text areas
 - textValueChanged(TextEvent event)

WindowListener

- Handles high-level window events
 - windowOpened, windowClosing, windowClosed, windowIconified, windowDeiconified, windowActivated, windowDeactivated
 - windowClosing particularly useful

Example: Simple Whiteboard

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class SimpleWhiteboard extends Applet {
  protected int lastX=0, lastY=0;
  public void init() {
    setBackground(Color.WHITE);
    setForeground(Color.BLUE);
    addMouseListener(new PositionRecorder());
    addMouseMotionListener(new LineDrawer());
  }
  protected void record(int x, int y) {
    lastX = x; lastY = y;
  }
```

Simple Whiteboard (Continued)

```
private class PositionRecorder extends MouseAdapter {
   public void mouseEntered(MouseEvent event) {
      requestFocus(); // Plan ahead for typing
      record(event.getX(), event.getY());
   }
   public void mousePressed(MouseEvent event) {
      record(event.getX(), event.getY());
   }
} ...
```

Simple Whiteboard (Continued)

```
private class LineDrawer extends MouseMotionAdapter {
   public void mouseDragged(MouseEvent event) {
      int x = event.getX();
      int y = event.getY();
      Graphics g = getGraphics();
      g.drawLine(lastX, lastY, x, y);
      record(x, y);
   }
}
```

Simple Whiteboard (Results)

| j 🔮 🗼 🏾 | - 🟦 🧈 🖻 🕹 📽 🕲 🐰 | |
|--------------------------|---------------------|--|
| Simple Whiteboard Applet | | |
| | | |
| | 1 | |
| | | |
| | $ \chi \bigcirc$ | |
| | | |
| | | |
| | | |
| | | |

}

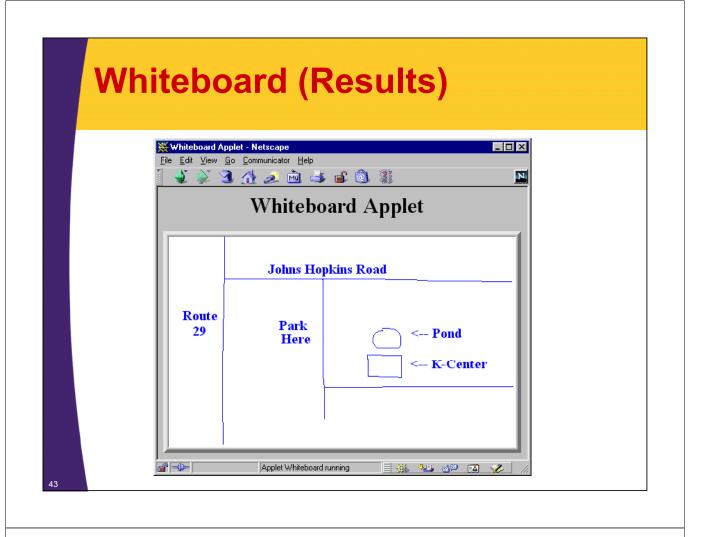
Whiteboard: Adding Keyboard Events

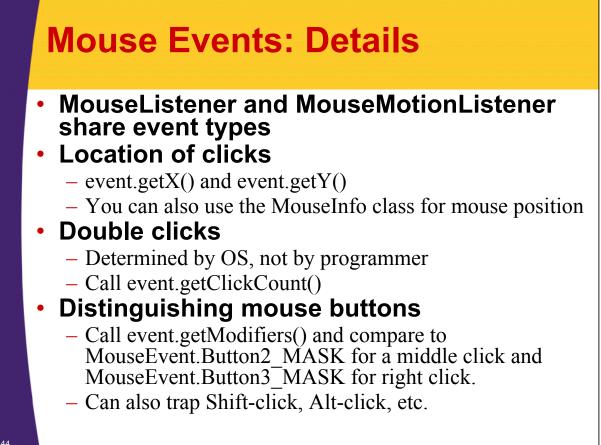
```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class Whiteboard extends SimpleWhiteboard {
  protected FontMetrics fm;
  public void init() {
    super.init();
    Font font = new Font("Serif", Font.BOLD, 20);
    setFont(font);
    fm = getFontMetrics(font);
    addKeyListener(new CharDrawer());
  }
```

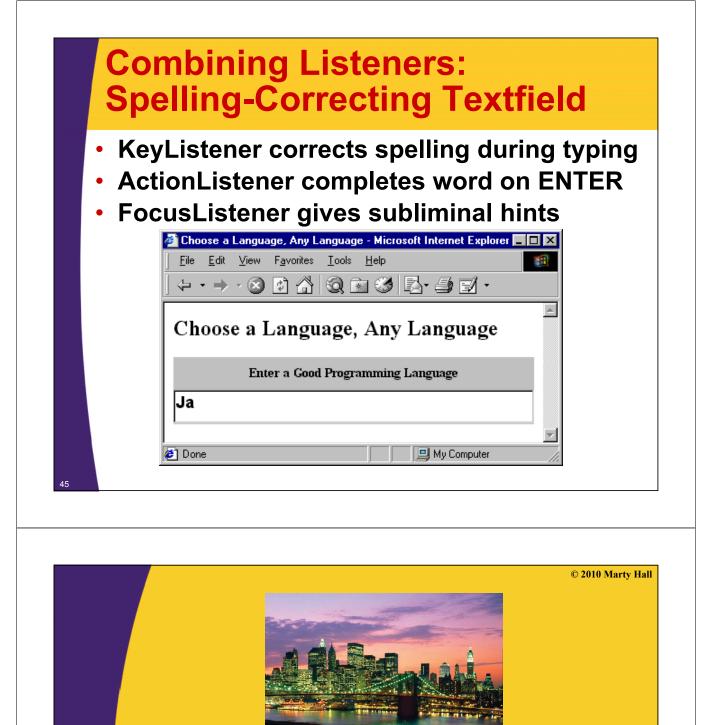
```
Whiteboard (Continued)
```

```
private class CharDrawer extends KeyAdapter {
    // When user types a printable character,
    // draw it and shift position rightwards.
    public void keyTyped(KeyEvent event) {
        String s = String.valueOf(event.getKeyChar());
        getGraphics().drawString(s, lastX, lastY);
        record(lastX + fm.stringWidth(s), lastY);
    }
}
```

}







Wrap-Up

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

General strategy

- Determine what type of listener is of interest
 - Check table of standard types
- Define a class of that type
 - Extend adapter separately, implement interface, extend adapter in named inner class, extend adapter in anonymous inner class
- Register an object of your listener class with the window
 - Call addXxxListener

Understanding listeners

- Methods give specific behavior.
 - Arguments to methods are of type XxxEvent
 - Methods in MouseEvent of particular interest

Preview of Later Topics

Whiteboard had freehand drawing only

Need GUI controls to allow selection of other drawing methods

Whiteboard had only "temporary" drawing

- Covering and reexposing window clears drawing
- After cover multithreading, we'll see solutions to this problem
 - Most general is double buffering

• Whiteboard was "unshared"

 Need network programming capabilities so that two different whiteboards can communicate with each other



Questions?

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.