

Java Programming

Event Handling

Contents

- The Delegation Event Model
- Event Classes
- Event Listeners
- Adapter Classes
- Inner Classes
- Anonymous Inner Classes

The Delegation Event Model

- Applet is event-driven.
- Delegation Event model: JDK 1.1 introduced

addTypeListener() Method, removeTypeListener() Method

```
public void addTypeListener(TypeListener el)
public void addTypeListener(TypeListener el) throws TooManyListenersException
public void removeTypeListener(TypeListener el)
```

addActionListener() Method, removeActionListener() Method

```
void addActionListener(ActionListener al)
void removeActionListener(ActionListener al)
```

actionPerformed() Method

```
void actionPerformed(ActionEvent ae)
```

<http://java.sun.com/j2se/1.5.0/docs/api/java/awt/Color.html>

Event Classes

EventObject Constructor

```
EventObject(Object src)
```

ComponentEvent Constructor

```
ComponentEvent(Component src, int type)
```

getSource() Method

```
Object getSource()
String toString()
```

Primary Event Classes

```
ActionEvent
AdjustmentEvent
ComponentEvent
ContainerEvent
FocusEvent
InputEvent
ItemEvent
KeyEvent
MouseEvent
TextEvent
WindowEvent
```

AWTEvent Constructor

```
AWTEvent(Object source, int id)
```

getID(), toString() Methods

```
int getID()
String toString()
```

Event Classes

getComponent() Method

```
Component getComponent()
```

isAltDown(), ... Methods

```
boolean isAltDown()  
boolean isControlDown()  
boolean isMetaDown()  
boolean isShiftDown()
```

getModifiers() Method

```
int getModifiers()
```

MouseEvent Constructor

```
MouseEvent(Component src, int type, long when, int modifiers, int x, int y, int clicks, boolean triggersPopup)
```

getX(), getY() Method

```
int getX() int getY()
```

getPoint() Method

```
Point getPoint()
```

translatePoint() Method

```
void translatePoint(int x, int y)
```

getClickCount() Method

```
int getClickCount()
```

Event Listeners

xxxListener() Methods

```
void addMouseListener(MouseListener ml)  
void addMouseMotionListener(MouseMotionListener mml)  
void removeMouseListener(MouseListener ml)  
void removeMouseMotionListener(MouseMotionListener mml)
```

mouseXXX() Methods

```
void mouseClicked(MouseEvent me)  
void mouseEntered(MouseEvent me)  
void mouseExited(MouseEvent me)  
void mousePressed(MouseEvent me)  
void mouseReleased(MouseEvent me)
```

mouseDragged(), mouseMoved() Methods

```
void mouseDragged(MouseEvent me)  
void mouseMoved(MouseEvent me)
```

```
import java.applet.*;  
import java.awt.*;  
import java.awt.event.*;  
/*  
< applet code="MouseEvents" width=300 height=300 >  
</ applet >  
*/  
public class MouseEvents extends Applet  
implements MouseListener {  
    public void init() {  
        addMouseListener(this);  
    }  
    public void mouseClicked(MouseEvent me) {  
        setBackground(Color.blue);  
        repaint();  
    }  
    public void mouseEntered(MouseEvent me) {  
        setBackground(Color.green);  
        repaint();  
    }  
    public void mouseExited(MouseEvent me) {  
        setBackground(Color.red);  
        repaint();  
    }  
    public void mousePressed(MouseEvent me) {  
        setBackground(Color.white);  
        repaint();  
    }  
    public void mouseReleased(MouseEvent me) {  
        setBackground(Color.yellow);  
        repaint();  
    }  
}
```

Adapter Classes

Adapter Classes

<i>Adapter Class</i>	<i>Listener Interface</i>
<i>ComponentAdapter</i>	<i>ComponentListener</i>
<i>ContainerAdapter</i>	<i>ContainerListener</i>
<i>FocusAdapter</i>	<i>FocusListener</i>
<i>KeyAdapter</i>	<i>KeyListener</i>
<i>MouseAdapter</i>	<i>MouseListener</i>
<i>MouseMotionAdapter</i>	<i>MouseMotionListener</i>
<i>WindowAdapter</i>	<i>WindowListener</i>

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class MouseAdapterDemo extends Applet {

    public void init() {
        setBackground(Color.green);
        addMouseListener(new MyMouseAdapter(this));
    }
}

class MyMouseAdapter extends MouseAdapter {
    MouseAdapterDemo mad;

    public MyMouseAdapter(MouseAdapterDemo mad) {
        this.mad = mad;
    }

    public void mousePressed(MouseEvent me) {
        mad.setBackground(Color.red);
        mad.repaint();
    }


    public void mouseReleased(MouseEvent me) {
        mad.setBackground(Color.green);
        mad.repaint();
    }
}
```

Delegated Event Model

■ Stages for Event Handling

- First, import event class
 - `import java.awt.event.*;`
- Define an overriding class of event type

```
Class ButtonListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        //...
    }
}
```

- Create an event listener object
 - `ButtonListener bt = new ButtonListener();`
- Register the event listener object
 - `b1 = new Button("OK");`
 - `b1.addActionListener(bt);`  [EventHandling.java]

Inner Classes

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*
 * <applet code="MouseInnerDemo" width=100 height=100>
 * </applet>
 */
public class MouseInnerDemo extends Applet {
    public void init() {
        setBackground(Color.green);
        addMouseListener(new MyMouseAdapter());
    }
    class MyMouseAdapter extends MouseAdapter {
        public void mousePressed(MouseEvent me) {
            setBackground(Color.red);
            repaint();
        }
        public void mouseReleased(MouseEvent me) {
            setBackground(Color.green);
            repaint();
        }
    }
}
```

Anonymous Inner Classes

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*
 * <applet code="MouseAnonymousDemo" width=100
 * height=100>
 * </applet>
 */
public class MouseAnonymousDemo extends Applet {
    public void init() {
        setBackground(Color.green);
        addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent me) {
                setBackground(Color.red);
                repaint();
            }
            public void mouseReleased(MouseEvent me) {
                setBackground(Color.green);
                repaint();
            }
        });
    }
}
```

An Anonymous Inner Class