

Programming in Java

CBCS BHCS03

By Parul Chachra

Chapter 24

Event Handling

Reference: Book prescribed as per
the syllabus

Using Delegation Event Model

- **Step 1:** Implement the appropriate interface in the listener so that it will receive the type of event desired
- **Step 2:** Implement code to register and unregister (if necessary) the listener as a recipient for the event notifications
- **NOTE1:** A source may generate several types of events. Each event must be registered separately
- **NOTE2:** an object may register to receive several types of events, but it must implement all of the interfaces that are required to receive these events

Handling Mouse Events

- To handle mouse events, you must implement the **MouseListener** and the **MouseMotionListener** interfaces
- The following applet displays the current coordinates of the mouse in the applet's status window
- Each time a button is **pressed**, the word "**Down**" is displayed at the location of the mouse pointer. Each time the button is **released**, the word "**Up**" is shown. If a button is **clicked**, the message "**Mouse clicked**" is displayed in the upper-left corner of the applet display area.
- As the mouse **enters** or **exits** the applet window, a message is displayed in the upper-left corner of the applet display area.
- When dragging the mouse, a * is shown, which tracks with the mouse pointer as it is dragged.
- Notice that the two variables, **mouseX** and **mouseY**, store the location of the mouse when a mouse pressed, released, or dragged event occurs.
- The coordinates are then used by **paint()** to display output at the point of these occurrences.

Handling Mouse Events

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
   <applet code="MouseEvents" width=300 height=100>
</applet> */
public class MouseEvents extends Applet
    implements MouseListener, MouseMotionListener {
    String msg = "";
    int mouseX = 0, mouseY = 0; // coordinates of mouse
    public void init() {
        addMouseListener(this);
        addouseMotionListener(this);
    }
    // Handle mouse clicked.
    public void mouseClicked(MouseEvent me) {
        // save coordinates
        mouseX = 0;
        mouseY = 10;
        msg = "Mouse clicked.";
        repaint();
    }
}
```

Handling Mouse Events

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
   <applet code="MouseEvents" width=300 height=100>
</applet> */
public class MouseEvents extends Applet
    implements MouseListener, MouseMotionListener {
```

- **MouseEvents** class extends **Applet** and implements both the **MouseListener** and **MouseMotionListener** interfaces
- These two interfaces contain methods that receive and process the various types of mouse events
- The applet is both the source and the listener for these events
- **Component** class, which supplies the **addMouseListener()** and **addMouseMotionListener()** methods, is a superclass of **Applet**

Handling Mouse Events

```
String msg = "";  
int mouseX = 0, mouseY = 0; // coordinates of mouse  
public void init() {  
    addMouseListener(this);  
    addouseMotionListener(this);  
}
```

- Inside **init()**, the applet registers itself as a listener for mouse events.
- This is done by using **addMouseListener()** and **addMouseMotionListener()**, which, as mentioned, are members of **Component**
- They are shown here:

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

- Here, *ml* is a reference to the object receiving mouse events, and *mml* is a reference to the object receiving mouse motion events
- In this program, the same object is used for both
- In the following slides, the applet then implements all of the methods defined by the **MouseListener** and **MouseMotionListener** interfaces
- These are the event handlers for the various mouse events. Each method handles its event and then returns

Handling Mouse Events

```
// Handle mouse entered.
public void mouseEntered(MouseEvent me) {
    // save coordinates
    mouseX = 0;
    mouseY = 10;
    msg = "Mouse entered.";
    repaint();
}

// Handle mouse exited.
public void mouseExited(MouseEvent me) {
    // save coordinates
    mouseX = 0;
    mouseY = 10;
    msg = "Mouse exited.";
    repaint();
}

// Handle button pressed.
public void mousePressed(MouseEvent me) {
    // save coordinates
    mouseX = me.getX();
    mouseY = me.getY();
    msg = "Down";
    repaint();
}
```


Handling Mouse Events

```
// Handle mouse dragged.
public void mouseDragged(MouseEvent me) {
    // save coordinates
    mouseX = me.getX();
    mouseY = me.getY();
    msg = "*";
    showStatus("Dragging mouse at " + mouseX + ", " + mouseY);
    repaint();
}

// Handle mouse moved.
public void mouseMoved(MouseEvent me) {
    // show status
    showStatus("Moving mouse at " + me.getX() + ", " + me.getY());
}

// Display msg in applet window at current X,Y location.
public void paint(Graphics g) {
    g.drawString(msg, mouseX, mouseY);
}
}
```

Handling Mouse Events

- Sample Output of the Applet:

