

Week 9 Lecture 2

Handling Events



1

This Week

Java Genesis:

–Ch9: Graphics and event handling

Lab Assessment 7 (deadline Week 10)

Quick Quiz for Chapter 8

2

Event Handling

To handle (i.e. detect and respond) to events such as

- pressing the mouse button
- moving the mouse
- pressing a key on the keyboard
- etc.....

we need to add **event listeners** to our window.

3

Event Listeners

When an event listener is added to a window it will

- listen out for specific events
- respond in its own specific way when an event it is listening for occurs

4

To specify the way an event listener responds when an event it is listening for occurs, we use Java's **anonymous inner class** construct.

5

```
public class NumberToy {  
    // instance variables  
    public int number;  
  
    // constructor  
    public NumberToy (int num) {  
        number = num%10;  
        if (number == 0) number = 10;  
    }  
  
    public void pressButton ( ) {  
        number = (number + 1)%10;  
        if (number == 0) number = 10;  
    }  
}
```

6

----(code not shown)-----

```
public class NumberToyGUI extends JFrame{  
    // instance variables  
    public NumberToy toy = new NumberToy(1);
```

----(code not shown)-----

7

```
public class AnotherNumberToy extends NumberToy {  
    // constructor  
    public AnotherNumberToy (int num) {  
        super(num);  
    }  
  
    public void pressButton ( ) {  
        number = (number - 1)%10;  
        if (number == 0) number = 10;  
    }  
}
```

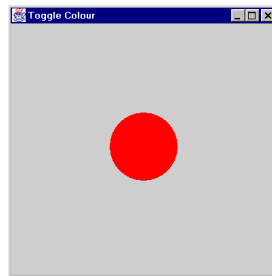
8

----(code not shown)-----

```
public class NumberToyGUI extends JFrame{  
    // instance variables  
    public NumberToy toy = new NumberToy(1){  
        public void pressButton ( ) {  
            number = (number - 1)%10;  
            if (number == 0) number = 10;  
        }  
    };
```

----(code not shown)-----

9



10

```
public class Main {  
  
    public static void main (String [ ] args) {  
        TogColourFrame togWin =  
            new TogColourFrame ( );  
        togWin.setVisible(true);  
    }  
}
```

11

```
import javax.swing.*.*;  
import java.awt.*.*;  
  
public class TogColourFrame extends JFrame {  
  
    // constructor  
    public TogColourFrame ( ) {  
        setTitle("Toggle Colour");  
        setBounds(400, 100, 400, 400);  
        TogColourPanel panel =  
            new TogColourPanel ( );  
        Container c = getContentPane ( );  
        c.add(panel);  
    }  
}
```

12

```

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

public class ToggleColourPanel extends JPanel {

    // instance variables
    private Color colour = Color.red;
    private int xPosn = 196, yPosn = 180;

    public void paintComponent (Graphics g) {
        super.paintComponent(g);
        g.setColor(colour);
        g.fillOval(xPosn-50, yPosn-50, 100, 100);
    }
}

```

13

The class MouseAdapter

```

public abstract class MouseAdapter {

    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){ }
}

```

14

The class MouseMotionAdapter

```

public abstract class MouseMotionAdapter {

    public void mouseMoved(MouseEvent e){ }
    public void mouseDragged(MouseEvent e){ }
}

```

15

```

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

public class ToggleColourPanel extends JPanel {

    // instance variable
    private Color colour = Color.red;
    private int xPosn = 196, yPosn = 180;

```

16

```

// constructor
public ToggleColourPanel ( ) {
    addMouseListener(new MouseAdapter ( ) {
        public void mousePressed (MouseEvent e) {
            if (colour == Color.red)
                colour = Color.blue;
            else colour = Color.red;
            repaint();
        }
        public void mouseEntered(MouseEvent e) {
            colour = Color.yellow;
            repaint();
        }
        public void mouseExited(MouseEvent e) {
            colour = Color.magenta;
            repaint();
        }
    });
}

```

17

```

addMouseMotionListener
    (new MouseMotionAdapter ( ) {
        public void mouseMoved(MouseEvent e) {
            xPosn = e.getX();
            yPosn = e.getY();
            repaint();
        }
    });
}

public void paintComponent (Graphics g) {
    super.paintComponent(g);
    g.setColor(colour);
    g.fillOval(xPosn-50, yPosn-50, 100, 100);
}
}

```

18