

Week 9, Lecture 1

Displaying Graphics



1

This Week

Lecture 1: Displaying graphics

Lecture 2: Handling events

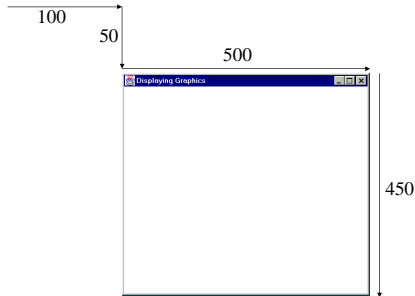
Java Genesis:

–Ch9: Graphics and event handling

Lab Assessment 7 (deadline Week 10)

Quick Quiz for Chapter 8

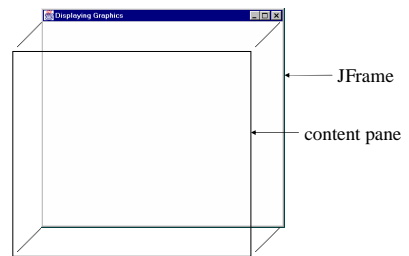
2



`setBounds(100, 50, 500, 450)`

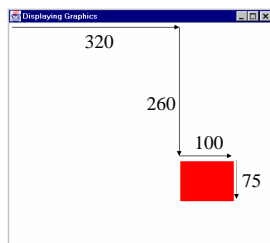
3

Structure of JFrame objects (abbreviated)



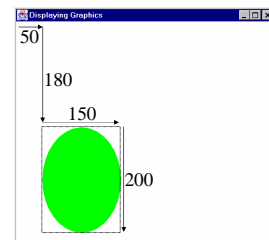
Graphical objects are painted on a **JPanel** object and this panel is then added to the frame's content pane.

4



`fillRect(320, 260, 100, 75)`

5



`fillOval(50, 180, 150, 200)`

6



drawImage(im, 200, 20, this)

7

```
import genesis.*;
public class Main {

    public static void main (String [ ] args) {
        OurFrame frame = new OurFrame();
        frame.setVisible(true);
        while (true) {
            Delay.milliseconds(20);
            frame.moveGraphic();
        }
    }
}
```

8

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

public class OurFrame extends JFrame {

    // instance variable
    private OurPanel panel = new OurPanel ();

    public OurFrame ( ) { // constructor
        setTitle("Displaying Graphics");
        setBounds(100, 50, 500, 450);
        Container c = getContentPane();
        c.add(panel);
    }
    public void moveGraphic ( ) {
        panel.moveGraphic();
    }
}
```

9

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

public class OurPanel extends JPanel {

    //instance variables
    private int xCorner = 200, yCorner = 200;
    private int step = 5;

    // constructor
    public OurPanel ( ) {
        setBackground(Color.white);
    }
}
```

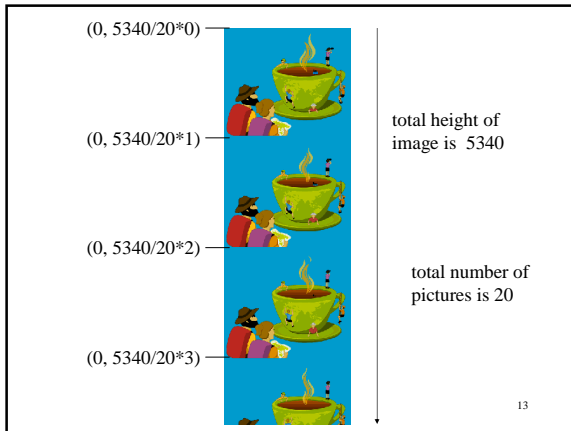
10

```
// paint graphical objects
public void paintComponent (Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.red);
    g.fillRect(320, 260, 100, 75);
    g.setColor(Color.green);
    g.fillOval(50, 180, 150, 200);
    Toolkit tk = Toolkit.getDefaultToolkit();
    Image im = tk.getImage("Evangeli on.gif");
    g.drawImage(im, 200, 20, this);
    g.setColor(Color.magenta);
    g.fillOval(xCorner, yCorner, 100, 100);
}
```

11

```
public void moveGraphic ( ) {
    if (xCorner > 390 || xCorner < 0) {
        step = -step;
    }
    xCorner = xCorner + step;
    repaint();
}
}
```

12



```
import genesis.*;
public class Main {

    public static void main (String [ ] args)
    {
        ImageFrame imageWin = new ImageFrame( );
        imageWin.setVisible(true);
        while (true) {
            Delay.miliseconds(200);
            imageWin.nextPicture( );
        }
    }
}
```

14

```
import javax.swing.*;
import java.awt.*;
public class ImageFrame extends JFrame {

    // instance variables
    private ImagePanel panel = new ImagePanel ( );

    // constructor
    public ImageFrame ( ) {
        setTitle("Image Animation");
        setBounds(100, 50, 310, 295);
        Container c = getContentPane( );
        c.add(panel);
    }

    public void nextPicture ( ) {
        panel.nextPicture( );
    }
}
```

15

```
import javax.swing.*;
import java.awt.*;
public class ImagePanel extends JPanel {

    // instance variables
    private Image im;
    private int imageCount = 20;
    private int imageHeight = 5340;
    private int current = 0;

    // constructor
    public ImagePanel ( ) {
        Toolkit tk = Toolkit.getDefaultToolkit( );
        im = tk.getImage("Javabig.gif");
    }
}
```

16

```
public void paintComponent (Graphics g) {
    super.paintComponent(g);
    int offset =
        -(imageHeight/imageCount)*current;
    g.drawImage(im, 0, offset, null);
}

public void nextPicture ( ) {
    current = (current+1)%imageCount;
    repaint();
}
}
```

17