Lecture 3.1

Using Graphics with JFrame

javax.swing.JFrame

- int x
- int y
- int width
- int height
- Color backgroundColor

«constructor»
+ JFrame(String)

«update»
+ void add(java.awt.Component, int)
+ void remove(java.awt.Component)
+ void repaint()
+ void setBackground(java.awt.Color)
+ void setBounds(int, int, int, int)
+ void setLayout(java.awt.LayoutManager)
+ void setVisible(boolean)
+ ...

© 2006 Pearson Addison-Wesley. All rights reserved
JFrame Class Specifications

Invariant
A JFrame object ...

• is a rectangular window placed upon a computer display.
• is positioned so that its upper left corner is \( x \) pixels from
  the left and \( y \) pixels from the top of the display
• has a visible region that is \( \text{width} \) pixels wide, \( \text{height} \)
  pixels high, and with a background color of \( \text{backColor} \).

(Below is a JFrame with \text{backColor} of gray on a white display.)

Constructor Methods

public JFrame(String s)

post: a new JFrame (window) object is created
  and \( s \) is displayed in the window’ s title bar

note: this method call needs to be followed by calls to setBounds and setVisible

Update Methods

public void add(java.awt.Component pic, int \( j \))

pre: \( j == 0 \) for best results
post: image \( pic \) will be drawn upon this JFrame

public void remove(java.awt.Component pic)

post: image \( pic \) will be removed from this JFrame (assuming it was previously added)

public void repaint()

post: this JFrame is marked to be redrawn as soon as possible

public void setBounds(int \( \text{newX} \), int \( \text{newY} \), int \( w \), int \( h \))

pre: \( w >= 0 \) and \( h >= 0 \)
post: \( x == \text{newX} \) and \( y == \text{newY} \)
  and \( \text{width} == w \) and \( \text{height} == h \)
3.1.5 JFrame Class Specifications (continued)

Update Methods

public void setBackground(java.awt.Color c)
post: backColor == c
note: This method must be applied to this getContentPane()

public void setLayout(java.awt.LayoutManager m)
pre: m == null (for our purposes)
post: added objects are rearranged via m

public void setVisible(boolean b)
post: b == true implies this JFrame is made visible and brought to the front

3.1.6 An Example

import java.awt.Color;
import javax.swing.JFrame;

public class Driver {
    private JFrame blackWin, greenWin;
    public Driver() {
        blackWin = new JFrame("Mine");
        blackWin.setBounds(10, 10, 100, 200);
        blackWin.setLayout(null);
        blackWin.getContentPane().setBackground(Color.black);
        blackWin.setVisible(true);
        greenWin = new JFrame("Yours");
        greenWin.setBounds(150, 100, 100, 50);
        greenWin.setLayout(null);
        greenWin.setVisible(true);
        greenWin.getContentPane().setBackground(Color.green);
    }
}

Note the import

String constants require double quotes.

Code Pattern for JFrame initialization

1) 2) 3) 4)
**the add method**

**method specification**

```java
public void add(java.awt.Component pic, int j)
    pre:  j == 0  for best results
    post:  image pic will be drawn upon this JFrame
```

The options for Component arguments include...

- `java.awt.Label`
- `Line`
- `Oval`
- `Rectangle`
- and many others

---

**java.awt.Label**

**Code Pattern for Label use**

1) instantiate: `new Label`
2) set x, y & dimensions: `setBounds`
3) add to some canvas
4) `repaint()`

**Example**

```java
import java.awt.*;
import javax.swing.JFrame;
public class Driver {
    private JFrame win;
    private Label wiLabel;
    public Driver() {
        win = new JFrame("the window");
        win.setBounds(10, 10, 200, 200);
        win.setLayout(null);
        win.setVisible(true);
        wiLabel = new Label("Wisconsin");
        wiLabel.setBounds(10, 40, 120, 20);
        wiLabel.setForeground(Color.blue);
        wiLabel.repaint();
        win.add(wiLabel, 0);
    }
}
```
Rectangle

- int x
- int y
- int width
- int height
- Color backColor

«constructor»
+ Rectangle(int, int, int, int)

«update»
+ void add(java.awt.Component, int)
+ void repaint()  
+ void setBackground(java.awt.Color)
+ void setLocation(int, int)
+ void setSize(int, int)

. . .

Note: no import required for Rectangle, but Rectangle.class file must be in same folder as Driver.class.

© 2006 Pearson Addison-Wesley. All rights reserved

Code Pattern for Rectangle use
1) instantiate: new Rectangle
2) setBackground
3) add to some canvas
4) repaint()

Example
import java.awt.Color;
import javax.swing.JFrame;
public class Driver {
    private JFrame win;
    private Rectangle box;
    public Driver() {
        win = new JFrame("the window");
        win.setBounds(10, 10, 200, 200);
        win.setLayout(null);
        win.setVisible(true);
        box = new Rectangle(50, 50, 10, 10);
        box.setBackground(Color.magenta);
        box.repaint();
        win.add(box, 0);
    }
}

Note: no import required for Rectangle, but Rectangle.class file must be in same folder as Driver.class.

the add method

A graphical object (excepting JFrame) only becomes visible if it is added to some other visible object (called the underlying, or parent, container).

Drawing Rules for container.add(object, 0);
1) A container might be a JFrame, Oval, Rectangle, (others).
2) Each object can be added to no more than one container.
3) Last add to the same container appears in front.
4) An added object is clipped to the bounding border of its container.
the repaint method

A call to repaint() is needed (sometimes) to cause an object to become visible.

Example

```java
... square = new Rectangle(30, 40, 50, 50);
window.add(square, 0);
square.repaint();
circle = new Oval(0, 0, 20, 20);
square.add(circle, 0);
circle.repaint();
```

or

```java
... square = new Rectangle(30, 40, 50, 50);
window.add(square, 0);
circle = new Oval(0, 0, 20, 20);
square.add(circle, 0);
window.repaint();
```

Example

```java
import java.awt.Color;
import javax.swing.JFrame;
public class Driver {
    private JFrame win;
    private Rectangle grayRect;
    private Oval redOval, whiteDot;

    public Driver() {
        win = new JFrame("the window");
        win.setBounds(10, 10, 200, 100);
        win.setLayout(null);
        win.getContentPane().setBackground(Color.lightGray);
        win.setVisible(true);
        grayRect = new Rectangle(40, 40, 40, 50);
        grayRect.setBackground(Color.darkGray);
        win.add(grayRect, 0);
        whiteDot = new Oval(10, 10, 80, 80);
        whiteDot.setBackground(Color.white);
        grayRect.add(whiteDot, 0);
        redOval = new Oval(60, -20, 100, 50);
        redOval.setBackground(Color.red);
        win.add(redOval, 0);
        win.repaint();
    }
}
```