Graphical User Interface and Graphics

Package 'java.awt'

contains:

graphics shapes, lines and images, fonts, colours etc.

components buttons, text fields, menus, scroll bars

layout managers arrange the layout on the screen

event handlers (java.awt.event) handles external events

image manipulation (java.awt.image) incorporate images in different formats

Graphical User Interface

- GUI - pronounced 'gooey'.
  How about wysiwyg - pronounced 'wizzywig' ?
  meaning?
  What you see is what you get.
- A useful package called Abstract Windowing Toolkit - awt - pronounced 'ought'.
- GUI is our main interest for this course.
- Reading:
  chp 10, Bishop's 'Java Gently' or
  chp 11-13, Deitel& Deitel's 'Java-How to Program'
  (many copies in the library).

Abstract classes in awt

- Components
- Container
- MenuComponent
- Graphics
- FontMetrics
- Image
- PrintJob
- Toolkit

Making a GUI

There are ways to make a GUI: you may

1. display graphics in a window
2. add some interaction facilities: e.g. add buttons or other components to allow the user input instructions.
3. integrate your program into the WWW to allow distant access, using *Applets*.

Some explanation

```
//including all classes in the 'java.awt' package
import java.awt.*;

public class testFrame{

  public static void main(String argv[]) {
  //create a new frame with the title "A Frame Test!"
    Frame testFrame = new Frame("A Frame Test!");
  //specify the size (width, height) of the window
    testFrame.setSize(200,100);
  //determine whether or not the frame is visible
    testFrame.setVisible(true);
  } // end main
} // end class testFrame
```


Example - a frame by AWT

This program below will display a small window of size 200x100 pixels:

```
import java.awt.*;

public class testFrame{

  public static void main(String argv[]) {
    Frame testFrame = new Frame("A Frame Test!");
    testFrame.setSize(200,100);
    testFrame.setVisible(true);
  } // end main
} // end class testFrame
```


Graphics

- Java's coordinate system

```
0,0
----------------- x axis
|
|          . (x,y)
|
|
y axis
```

- coordinate are measured in *pixels*.
- graphic objects

Package 'javax.swing'

- is another newer set of similar tools to 'awt' for GUI.
- implemented entirely in Java and it is a standard extension of Java
- programs are bigger so it is slower to run
- the shapes for buttons, text fields, menus, etc., are more consistent looking, independent from the machine that runs the programs

Swing also contains:

graphics shapes, lines and images, fonts, colours etc.

components windows, buttons, text fields, menus, scroll bars

layout managers arrange the layout on the screen

event handlers (java.awt.event) handles external events

image manipulation (java.awt.image) incorporate images in different formats

- principles of design are similar to 'awt'; some parts use 'awt's, so we shall see 'awt' but use mainly 'swing' for the course
- 'swing' is very big. see online *Java Tutorial* at `www.javasoft.com` for more information.

Constructing a frame

.. by creating a `JFrame` instance:

```
=========================================
import javax.swing.*;

public class FRAMENAME {
   public static void main(String[] args) {
     JFrame f = new JFrame();
     ... // other statements
   }
} // end FRAMENAME
=========================================
```

.. or by extending `JFrame`:

```
=========================================
class emptyFrame extends JFrame {
   public emptyFrame() {
     ...
     // other statements
     // such as defining the frame size, etc.
   } // end emptyFrame
} // end emptyFrame
=========================================
```

Example

```
import javax.swing.JFrame;

public class jFrame1 {
   public static void main(String[] args) {
      emptyFrame f = new emptyFrame();
      f.setTitle("An empty frame again");
      f.show();
   } // end main
} // end jFrame1

class emptyFrame extends JFrame {
   public emptyFrame() {
      final int defultFrameWidth = 300;
      final int defultFrameHeight = 250;
      setSize(defultFrameWidth, defultFrameHeight);
   } // end emptyFrame
} // end emptyFrame
```


Some values and methods

Possible value properties that can be checked by queries:

```
============================
public Color getForeground();
public Color getBackground();
public Point getLocation();
public Dimension getSize();
============================
```

Methods that can be used to update values:

```
==================================
public void setForeground(Color fg);
public void setBackground(Color bg);
public void setLocation(Point p);
public void setSize(Demension d);
==================================
```


Some JComponents

JComponent is a subclass of the AWT class java.awt.Component.

JButton
JCheckBox
JComboBox
JfileChooser
JLabel
JList
JRadioButton
JScrollPane
JSlider
JTextArea
JTextField

Example - an empty Frame by Swing

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

public class sFrame{
  static void myFrame () {
    JFrame tFrame = new JFrame("A JFrame Test!");
    tFrame.setSize(200,100);
    tFrame.setVisible(true);
  }

  public static void main(String [] argv) {
    myFrame ();
  } // end main
} // end class sFrame
```

Adding components (layout)

We create a new object of JButton with a label "myButton".

```
=====================================
JButton b = new JButton(``myButton'');
Container cp = f.getContentPane();
cp.add(b);
=====================================
```

Each container is equipped with a *layout manager*.

```
=============================================
public LayoutManager getLayout();
public void setLayout(LayoutManager manager);
=============================================
```


Some standard layout managers

```
===============
FlowLayout()
BorderLayout()
GridLayout()
CardLayout()
BoxLayout()
Overlayout()
===============
```

Adding FlowLayout

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

public class flowLayout{
   public static void main (String[] args) {
      JFrame f = new JFrame("A flow layout");
      JButton b1 = new JButton("button 1");
      JButton b2 = new JButton("button 2");
      JButton b3 = new JButton("button 3");
      Container cp = f.getContentPane();
      cp.setLayout(new FlowLayout());
      cp.add(b1);
      cp.add(b2);
      cp.add(b3);
      f.setSize(300,70);
      f.setVisible(true);
   } // end main
} // end flowLayout
```


Adding BorderLayout

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

public class dFrame {
  public static void main (String [] args) {
    JFrame f = new JFrame("A Frame");
    Container cp = f.getContentPane();
    // cp.setLayout(new BorderLayout()); // default
    cp.add(new JButton("N"), BorderLayout.NORTH);
    // here add takes 2 arguments:
    //'public void add (Component comp,
    //                    Object constraints);'
    cp.add(new JButton("S"), BorderLayout.SOUTH);
    cp.add(new JButton("W"), BorderLayout.WEST);
    cp.add(new JButton("E"), BorderLayout.EAST);
    cp.add(new JButton("C"), BorderLayout.CENTER);
    f.setSize(300,300);
    f.setVisible(true);
  } // end main
} // end dFrame
```

Methods in *Graphics* class

```
=============================================
clearRect(int X,int Y,int WIDTH,int HEIGHT)
copyArea(int X,int Y,int WIDTH,int HEIGHT,
                              int DX,int DY)
drawChars(char [] DATA,int OFFSET,int LENGTH,
                                int X,int Y)
drawLine(int X1, int Y1, int X2, int Y2)
drawOval(int X, int Y, int WIDTH, int HEIGHT)
drawRect(int X, int Y, int WIDTH, int HEIGHT)
drawString(String STRING, int X, int Y)
fillOval(int X, int Y, int WIDTH, int HEIGHT)
fillRect(int X, int Y, int WIDTH, int HEIGHT)
setColor(Color C)
setFont(Font F)
=============================================
```

```
      g.drawOval (80, 80, 50, 40);
      g.setColor (Color.yellow);
      g.fillRect (40, 120, 200, 40);
      g.setColor (Color.black);
      g.drawString ("A test!", 100, 180);
    } // end paintComponent
  } // end myGraphics
} // end draw1
```


Drawing in a frame

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

class draw1 extends Frame {
  public draw1 () {
    add ("Center", new myGraphics());
    setTitle ("A test");
    setSize(300,280);
    setVisible (true);
  } // end draw1()

  public static void main (String [] arg) {
    new draw1 ();
  } // end main

  class myGraphics extends JPanel {
    public void paintComponent(Graphics g) {
      g.setColor (Color.red);
      g.fillRect (40,20,200,40);
      g.setColor (Color.green);
      g.fillOval (10, 40, 80, 30);
      g.setColor (Color.red);
```