COMS W1114 - Java Lab

Lab 10
Wednesday, April 21, 2004
&
Thursday, April 22, 2004

-1-

Note

- Last homework will be out soon. You will be using AWT to create a GUI (graphical user interface)
- Your grades are now up off of a link on the course website. Report any errors to Janak!

-2-

What we are covering today

- Review from Lab 9
  - AWT
  - Graphics object
- Event based programming

-3-

AWT

- AWT is a java package that we will be using in order to create graphical user interfaces
- Some important classes within the AWT package
  - Containers:
    - Frame ← has an titlebar, can contain many ‘things’
    - Canvas
    - Panel
  - What we will generally do is create our own class, which ***extends*** one of the above classes
  - Each of the above containers has a paint method that we will ***inherit*** but will usually ***override*** when we want to customize the container’s graphcs.

-4-

Paint method

you never have to call the paint method. Java will automatically call the paint method for you:

1) when the container appears
2) when the container is being moved around

```
public void paint(Graphics g){
    //java code here
}
```

but if you want to explicitly repaint your canvas without waiting for the user to move the window around you should call repaint();

-5-

Paint method cont’d

```
public void paint(Graphics g){
    //java code here
}
```

so what’s the deal with Graphics g?

g is the variable name of the Graphics object that is passed into the paint method automatically. (this can be renamed)

In the Graphics class, you will see many useful methods

```
    drawLine(….);
    fillCircle(….); etc
```

which you can now access through the graphics object!

```
    g.drawLine(10,20, 30, 40);
    g.fillOval(5,4,2,2)
```

-6-

More awt objects

- Frame and Canvas are great for simple drawing. What if you want to make an interactive application?
- – Want TextFields
  - – TextField t = new TextField(“initial text”, 15);
  - – add (t)
- Want Labels
  - – `add(new Label (“some text”));`
- Want Buttons
  - – a little more involved, but rather straightforward
  1. create a Button object
     `Button myButton = new Button(“Submit”);`
  2. add it to the Frame/Canvas - recall, these are Container objects. Note that Containers have this *add* method (seen with Labels)
     `add(myButton);`
- Why no x/y coordinates for the Button???
  - – there is a *Layout Manager* to coordinate placement (nice :)

-7-

awt objects

- awt objects (like every other java object) has methods associated with them
- for example the once you create a TextField, you can call methods such as getText() which will return the string inside your textField.
  - – explore the API!

-8-

Layout Manager

- when you add components, you are adding them to your container, given that you have previously specified one (or will default to borderLayout)
- Layout Manger take control of the over the positioning of components and arrange them sensibly.
- There are 5 different managers! We’ll only talk about three:
  - – FlowLayout, BorderLayout(default) and GridLayout

```
setLayout(new Manager(parameter)); //format
```

example:

```
setLayout(new FlowLayout(FlowLayout.CENTER,horigap,vertigap));
```

-9-

Simple Event

- Make a button do something
- We have our button myButton and we've added it

```
    Button myButton = new Button("Submit");
       add(myButton);
```

- Now need to "listen" for actions/events we care about

```
    myButton.addActionListener (this);
```

this means the current frame will be responsible for the code for some *ActionPerformed* method(what?! pretty easy….)

```
public void actionPerformed (ActionEvent e){
   if (e.getSource() == buttonname1) {
         statements;
   } else
   if (e.getSource() == buttonname2) {
         statements;
   } //etc
}
```

-10-

Different Kinds of events

- so far we've only worked with ActionEvent which reports if any action has been performed on a specified component

Event	Listener	methods
ActionEvent	ActionListener	actionPerformed
MouseEvent	MouseListener	mouseClicked, mousePressed etc..
KeyEvent	KeyListener	keyPressed, keyTyped
TextEvent	TextListener	textValueChanged
WindowEvent	WindowListener	windowClosed, windowActivated …

-11-

so what would you do to get info from a textfield?

- lets write the pseudo code.

-12-

Interfaces

- so you want to use one of the event listeners?
- java has Listener interfaces which specifies the methods that the listener MUST defined (listed on previous slide and on pg 423)
- if you want to detect any of the actions, you need to implement its Listener, and then be sure to define all its methods!
- see code example for syntax

-13-

End Notes

- Fill out the course evaluation! Win your iPod

  http://oracle.seas.columbia.edu/wces/
- Please also remember to rate your TAs *(you can rate any TA in this class, not just your lab instructor!)*
- Maryam will be out of the country starting on Sunday 4/25-Thursday 5/6.

-14-