COMS W1114 - Java Lab

Lab 13
Wednesday, April 28, 2004
&
Thursday, April 29, 2004

Notes

• HW5 ready
• Your grades are now up off of a link on the course website. Report any errors to Janak!
• HW6 Due Wed, May 5 @ 5p.
• Check bboard for OH changes

What we are covering today

• Go over HW5 solution
• Review from Lab 12
  – Event based programming
• Applets
• Packages
• Review
Simple Event

- Make a button do something
- We have our button myButton and we've added it:
  ```java
  Button myButton = new Button("Submit");
  add(myButton);
  ```
- Now need to "listen" for actions/events we care about
  ```java
  myButton.addActionListener (this);
  ```
  this means the current frame will be responsible for the code for some `ActionPerformed` method (what?! pretty easy...)

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

Different Kinds of events

<table>
<thead>
<tr>
<th>Event</th>
<th>Listener</th>
<th>methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionEvent</td>
<td>ActionListener</td>
<td>actionPerformed</td>
</tr>
<tr>
<td>MouseEvent</td>
<td>MouseListener</td>
<td>mouseClicked, mousePressed etc.</td>
</tr>
<tr>
<td>KeyEvent</td>
<td>KeyListener</td>
<td>keyPressed, keyTyped</td>
</tr>
<tr>
<td>TextEvent</td>
<td>TextListener</td>
<td>textValueChanged</td>
</tr>
<tr>
<td>WindowEvent</td>
<td>WindowListener</td>
<td>windowClosed, windowActivated ...</td>
</tr>
</tbody>
</table>

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 page 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
Applets (1)

- Want to display your programs/GUIs in a web browser?
- Write an applet! Really simple since we know how awt works
  - Import Applet and Graphics
    ```java
    import java.applet.Applet;
    import java.awt.Graphics;
    ```
  - Extend the Applet class
    ```java
    public class HelloWorld extends Applet {
    ```
  - Implement some Applet methods (paint!)
    ```java
    public void paint(Graphics g) {
      g.drawString("Hello World", 50, 25);
    }
    ```

Applets (2)

- Run it in a Browser
  html here
  ```html
  <APPLET CODE="HelloWorld.class" WIDTH=150 HEIGHT=25>
  </APPLET>
  ```
  more html
- That's it!
- http://java.sun.com/docs/books/tutorial/applet/

Packages

- We've seen package use already
  ```java
  import java.awt.*;
  ```
- What if we want to write a class that conflicts with an existing name?
  - We package our code
  - Use the package command
  - Check out
    ```html
    java.sun.com/docs/books/tutorial/java/interpack/packages.html
    ```
Congratulations!

• We’ve covered the fundamentals of programming:
  – Datatypes: Primitives, Objects, Arrays
  – Iteration/Looping: While, For, do…while
  – Conditionals: if…else…elseif, switch statement
  – Objects: Constructors, Methods, a Variable’s Scope
  – Basic I/O: interactive I/O, file I/O, Basic Exception Handling
• How to do things:
  – Coding practices, Debugging tools, advanced I/O
  – Object Oriented (OO) Design
    • properties, references, abstraction, inheritance
  – GUIs, Event based programming

Congratulations!

• You’ve built:
  – A simple calculator reading program arguments
  – A palindrome checker reading keyboard input
  – Bank Account Manager w/ interactive interface
  – Shape calculator w/ interactive interface
  – Shape plotter w/GUI
• What are you going to build next?

End Notes

• Thank you! (from Maryam too)
• 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!)
• OH changes - check bboard