



JFC: Using Swing



Spring 2006

Course Goals

- Learn about JFC
 - What is in JFC
 - How to use it
- Learn about general GUI design
- Continue familiarization with Object-Oriented programming
- General Structure for each area will be
 - General GUI design principles
 - JFC Classes used to accomplish
 - Code Examples

Course Specifics

- Optional Books
 - Java 2D Graphics, Jonathan Knudsen
 - Java Swing, Eckstein, Loy and Wood
- Reference Books
 - Core Java Foundation Classes, Kim Topley
 - Up to Speed with Swing, User Interfaces with Java Foundation Classes by Steven Gutz
 - Essential Guide to User Interface Design, Wilbert O. Galitz
 - Java2 Platform API
 - Suggest you get used to using this, as it is your best class reference example.
 - “Java Class Libraries” by Chan and Lee is also good (3 volumes)

Development Environments

From "Mr. Bunny's Big Cup o' Java"

- There have always been a great many Java development tools available. Now that the language has been invented, their popularity has soared.
- There are two basic approaches to writing Java code: typing and clicking.
 - If you have a computer keyboard and don't mind taking your hands off the mouse, then the JDK (pronounced JDK) is for you. It involves a lot of typing, but some programmers feel this is the most rewarding part of their jobs.
 - If you are more of a clickist than a typer, consider using an IDE (pronounced just like it sounds). With an IDE you let the development environment do the work for you; all you have to do is bring it coffee.
- Seriously, there are several good IDE's out there, *Eclipse*, *Netbeans* or *Java*, *VisualCafe* and *JBuilder* are three of the most popular. Sun offers Forte for Java for free. I still prefer Emacs over an IDE.

The UI Developers Goal

- From Microsoft's "Windows User Experience"
 - After an application is installed, a user should be able to use it productively within the first five minutes
 - A user's first impression often biases their entire history with a product
 - Use "tiers" of introduction
 - Tutorial
 - Initial screen that displays key features
 - Use contextual tips to help a user learn without overwhelming them at first

Homework #0

- Homework for this class will be developed as applications on your computer, but will be submitted as Applets
- HW#0 contains a set of source files and an image file. Your job is to compile, package and submit it.

Building an executable jar file

- The Java Archive (JAR) file format bundles multiple files into a single archive file
 - Class files
 - text/html
 - Images
- You can run jar files by
 - `java -jar jar-file`
 - Double clicking a jar file in windows
- To run the jar files you need to add a Main-Class header to the JAR files “manifest”
 - `Main-Class: classname`

Building HW#0

- Create a build directory `/tmp/HW0`
- Place `HW.java` and `gerbil.gif` in the `edu/jhu/en605482/part0` directory under your build directory
- Create the `mc0.mf` file in the build directory
 - Add `Main-Class: edu.jhu.en605482.part0.HW` to `mc0.mf`
- In the build directory, run `jar -cvmf mc0.mf HW0.jar edu`
- You should now have a `HW0.jar` file in your build directory
- Run the application by typing `java -jar HW0.jar`

Ant task

```
<!-- Make a submission jar file -->
<target name="pack1">
  <jar destfile="hw1.jar"
    basedir="."
    includes="edu/jhu/en605482/part1/**"
    excludes="**/*.java **/semantic.cache">
    <manifest>
      <attribute name="Built-By" value="R.B.Evans"/>
      <attribute name="Main-Class" value = "edu.jhu.en605482.part1.HW"/>
    </manifest>
  </jar>
</target>
```

Downloads

- Get the presentation at
- http://webdev.apl.jhu.edu/~rbe/java/SectionA_Intro/all.zip