CSCE 155 - Java

Lab 14 - Graphical User Interface Programming

Dr. Chris Bourke

Prior to Lab

Before attending this lab:

1. Read and familiarize yourself with this handout.

Some additional resources that may help with this lab:

- Oracle's Java Swing Tutorials:
  http://docs.oracle.com/javase/tutorial/uiswing/
- Another Swing tutorial: http://zetcode.com/tutorials/javaswingtutorial/

Peer Programming Pair-Up

To encourage collaboration and a team environment, labs will be structured in a *pair programming* setup. At the start of each lab, you will be randomly paired up with another student (conflicts such as absences will be dealt with by the lab instructor). One of you will be designated the *driver* and the other the *navigator*.

The navigator will be responsible for reading the instructions and telling the driver what to do next. The driver will be in charge of the keyboard and workstation. Both driver and navigator are responsible for suggesting fixes and solutions together. Neither the navigator nor the driver is "in charge." Beyond your immediate pairing, you are encouraged to help and interact and with other pairs in the lab.

Each week you should alternate: if you were a driver last week, be a navigator next, etc. Resolve any issues (you were both drivers last week) within your pair. Ask the lab instructor to resolve issues only when you cannot come to a consensus.

Because of the peer programming setup of labs, it is absolutely essential that you complete any pre-lab activities and familiarize yourself with the handouts prior to coming to lab. Failure to do so will negatively impact your ability to collaborate and work with others which may mean that you will not be able to complete the lab.

1 Lab Objectives & Topics

At the end of this lab you should be familiar with the following

- Graphical User Interface and Event-based Programming in Java using Swing
- Compiling and running a GUI program

2 Background

Many programs interact with users using a Graphical User Interface (GUI). Traditional GUI design involves the creation and interaction of widgets: general graphical elements that include labels, text boxes, buttons, etc. Most GUI programming is done using an Application Programmer Interface (API) which is a library or framework that provides a lot of the core functionality including:

- A window manager that handles the interaction of widgets
- A windowing system that works with the underlying operating system and hardware to render the graphics
- Factory functions that can be used to construct and configure widgets

The particular API that we will work with is Java?s Swing API which is part of Java?s Standard Developer Kit (SDK). Swing provides functionality for creating and managing widgets as well as the ability to change the look and feel of the application. Because it is Java, an application written in Swing will operate across different operating systems and should provide the same user experience.

A simple GUI program

We have provided a simple calculator program, (in the class file `Calculator.java`) that simulates a simple calculator. Two editable input boxes and one non-editable output box have been implemented along with several buttons to support arithmetic operations (addition and subtraction). The operations work by grabbing the values in the two input boxes, performing the arithmetic operation and placing the result into the output box.

You will extend the functionality of this program by adding another button to support

division. To do this, you will need to add code to create and configure the button, create an event listener (an `ActionListener`) and register it with your button. You will also need to add the button to the appropriate container.

3 Activities

Clone the project code for this lab from GitHub using the following URL: https://github.com/cbourke/CSCE155-Java-Lab14.

3.1 Getting Familiar with Swing

Several additional Swing examples have been provided to you including the incomplete calculator program. Familiarize yourself with the code by opening each source file and examining how they operate.

1. Open the source file for the three examples, `HelloWorld.java`, `KeyPad.java`, and `Calculator.java` source files
2. Examine the source files and answer the questions on your worksheet

3.2 Modifying the Program

You will now modify this program to support division.

1. Run the `HelloWorld` class in Eclipse and look at the code to get an idea of how GUI components work
2. Create a new button to support division:
   a) In the `initializeGUI` method, declare a new `JButton` object with the appropriate symbol representing division
   b) Create and add `ActionListener` objects and associate them with your new button
   c) Add your button to the appropriate container and in the appropriate order
3. Compile your program and run it on several values to make sure it works; demonstrate your working version to a lab instructor.

4 Advanced Activity (optional)

1. Explore Swing further by changing the appearance of your calculator program by adding borders, colors, etc. Add additional functionality by adding more buttons that compute mathematical functions.
2. Further explore Swing by changing the look and feel of your application. Read the following section of the Swing tutorial: http://docs.oracle.com/javase/tutorial/uiswing/lookandfeel/index.html and change the "theme" of your application.