Introduction to Graphical User Interfaces

**Purpose:**

The purpose of this lab is to introduce you to the Java Swing library, and to use and modify a simple graphical user interface (GUI).

**Set up:**

Create a directory named `Lab211/gui` in your Java directory, an copy the files form `~labCourse/Labs/Lab211/gui` to this directory.

**Java’s GUI API components:**

In this lab we build a graphical display for a very simple counter application. *We tie it to the model in the next lab.* To build a simple GUI, we need to be familiar with the following classes and interfaces: from the package `javax.swing`, `JPanel`, `JLabel`, `JButton`, `JTextField` and `BorderFactory`; from the package `java.awt`, `LayoutManager`, `FlowLayout`, `BorderLayout`, and `GridLayout`.

- Open the Java API documentation in a browser window. The documentation can be found here.

Choose `javax.swing` from the package list, and read about `JPanel`.

- Is a `JPanel` a `JComponent`?
- Is a `JPanel` a `Container`?

We will use a `JPanel` as a container: that is, as a visual component into which we can put other visual components.

- What methods does a `JPanel` inherit from `Container` that allows visual components to be put into it?

Before we add visual components to a `JPanel`, we should tell the `JPanel` how to organize the components. We do this by equipping the `JPanel` with a `LayoutManager`.

- `FlowLayout` is a `LayoutManager` defined in `java.awt`. How does it organize components?
- How does `BorderLayout` organize components?
- How does `GridLayout` organize components?
- Coming back to `JPanel`, how do we provide the `LayoutManager` we want to use?
- What is the default `LayoutManager` for a `JPanel`?

Now let’s look at some simple visual components we can add to a `JPanel` to build an interface. These are defined in `javax.swing`.

- Look up the specification for the class `JLabel`. How many constructors does `JLabel` have?
• Look up the specification for the class JTextField. How many constructors does JTextField have?
• Look up the specification for the class JButton. How many constructors does JButton have?
• When designing GUI’s, what will you use Label instances for?
• When designing GUI’s, what will you use Button instances for?
• When designing GUI’s, what will you use TextField instances for?

We can decorate visual components by giving them special borders. Read about the class javax.swing.BorderFactory.
• Name three kinds of borders.
• What method do you use to put a border on a JPanel?

Open the file CounterGUI.java and answer the following questions.
• How does the JFrame arrange the components placed in it?
• Does the JFrame have a border? If so, what kind of border?
• How many components does the JFrame have?
• Does button1 have a border?
• Does button2 have a border?
• Does label have a border?
• Is button2 part of the user interface? Why or why not?
• How will the components of the JFrame appear visually?

Now compile and execute Lab211.gui.CounterGUIStart. Note that there is no “intelligence” yet behind the interface. Close the window to terminate.

Edit CounterGUI and change the appearance a bit.
• Change the borders.
• Change the layout.
• Change the text on the components.

Complete architecture of an Application with a GUI front end

We are building the user interface as a JPanel. But a JPanel must be placed in another container such as a Window to be displayed. We will use a Window subclass JFrame, defined in the package javax.swing.

Look up the specification of JFrame.
• How many constructor does JFrame have?
• Is a JFrame a Container? a Component? a Window? a JComponent?

We are going to use JFrame methods getContentPane, setSize, pack, and setVisible.

• In what classes are each of these methods defined?
• What type of value is returned by getContentPane?
• What does pack do? Does it require any arguments?
• What does setVisible do? Does it require any arguments?

We don’t add the JPanel directly to the JFrame, rather, we add it to the content pane of the JFrame. Scan the description of JFrame to determine how this is done.

• What is the default layout manager of the content pane?

We have defined a class called Show whose only purpose is to provide a JFrame for displaying a JPanel.

• Read the source code of Show and CounterGUIStart. Note that the inFrame method of Show is static. What does this mean?
• Modify the last three arguments in the call to inFrame to try different size windows and different titles.

Post-lab:

Submit the following, as directed by your lab instructor:

• answers to the questions posed in the lab;
• a listing of CounterGUI with changes that you made and tried successfully.
• a listing of CounterGUIStart with changes that you made and tried successfully.