

Exercise01: Java Swing-1

Objectives

- To learn about the most basic Swing containers and components.
- To learn how to create handlers (or listeners).
- To learn to use WindowsBuilder.

BEFORE DOING ANYTHING ELSE!

Talk to your fellow students and form a group of two. Then SIGNUP to a group on Blackboard.

Overview

1. You create user interfaces by creating containers and placing elements in them.
2. Java Swing comes with many types of containers and many simple components (like JButtons, JTextField etc).
3. Class names of Swing components (and containers) are prefixed with a capital J.
4. **Layout Managers** help you place elements inside containers.
5. For complex user interfaces, instead of trying to place each element in some specific location in a single container, it is easier to put groups of elements in their own containers and then place these containers inside an outer container.
6. Swing components use Observer pattern to handle events. You can create your own "**listeners**" (basically observers) and "add" these observers to the desired subject. Thus, we can add a listener (that will handle a button click event) to a JButton.
7. To get radiobuttons to work correctly, they must be first put into a ButtonGroup.
8. There are many types of listeners for Swing components and you need to use the correct ones for a specific component. You will see this later when you read the examples.
9. WindowsBuilder is an Eclipse plugin that makes it easier for you to create GUIs.

1 Get started

You are to work in groups of two (NOT MORE than two please).

1.1 Try the simple examples

1. First, open blackboard, go to Course Contents, and then download exercise01.zip file into your workspace (U:\workspace or something like that!).
2. Then, unzip and traverse down to the 04_ExamplesSimple folder. Drag and drop the java files into the src folder of a new Java project in Eclipse.
3. Play with each of the given examples. Each of them has some "TODO" comments. Try following the instructions and observe what happens.

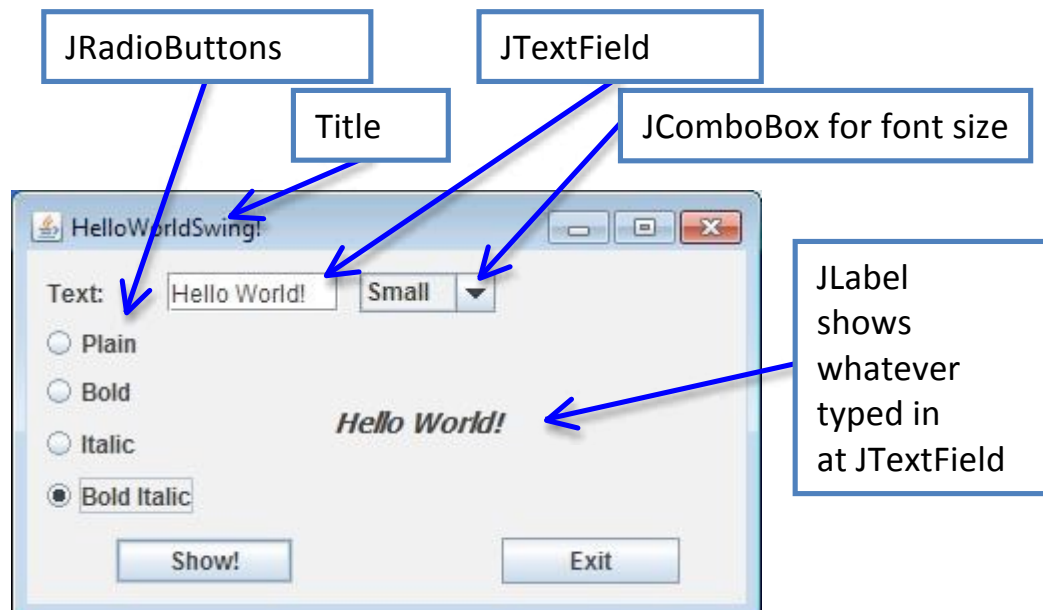
RAISE YOUR HAND IF YOU HAVE ANY QUESTIONS.

1.2 Try the intermediate examples

1. Traverse to the 05_ExamplesIntermediate folder. Drag and drop the java files into the src folder of a new Java project in Eclipse.
2. Open each of the examples in Eclipse and play with them.

2 Do this HelloWorldSwing exercise

Your first objective is to write Java Swing code to create the below user interface.



The UI has a few components (JFrame, JTextBox, JLabel, JButton, JRadioButton, JComboBox) and their associated action listeners. Make sure your code satisfies the below checklist. Use any number of containers and combination of layout managers to make it look as close as possible to the above image.

2.1 Use this CHECKLIST to make sure you got everything

- [] The window size should be set to width of 370 and height of 220.
- [] If the x button (located at the window control in the top right) is clicked the window should close **but the application should NOT terminate**.
- [] The title of the window should be set to "HelloWorldSwing!"
- [] Make it roughly look like the picture given (don't waste too much time perfecting this).
- [] When the user types in something in the JTextField, it shows up in the main area (which is a JLabel – for example, in the figure above "Hello World!" was typed in the JTextField and shows up in main area).
- [] When the user selects a radio button, the JLabel font gets changed appropriately.
- [] When the user selects an item (Tiny, Small, Medium, Large) in the JComboBox (which is like a dropdown), the JLabel font gets changed appropriately. Tiny is Arial and size 8. Small is Arial and size 12. Medium is Arial and size 20. Large is Arial and size 28.
- [] When the user clicks on the "Show" button, the JLabel's font color gets changed to red. Clicking it again makes it change back to black.
- [] When the user clicks on the "Exit" button, window is closed and the application is terminated.

3 Re-create HelloWorldSwing using WindowsBuilder

In this exercise, build the exact same application as in section 2 - but this time using the WindowsBuilder plugin. There is a tutorial in the exercise folder named "03_windowbuilderTutorial.pdf".

1. Choose the item labeled "Create new visual classes" from the main toolbar (it should be the second item from the left). Go to Swing->JFrame. In the wizard, name the JFrame "HelloWorldSwing2". Also, make sure that Source Folder and Package are set correctly.

2. You will see a new file named "HelloWordSwing2.java" is being created with some java code already in it. The generated source code creates and initializes a new JFrame.
3. At the bottom of the editor, there are two tabs: Source and Design. If you go to the Design view, you will get a visualized view of your application corresponding to what the user of the application will see. You will also have access to windows labeled "Structure", "Properties", and "Palette" that facilitate building and editing the JFrame visually.
4. Right-click on the empty window in the editor >Set Layout-> Absolute Layout.
5. Re-create the exact same application you made in section 2 – this time using WindowsBuilder. Use the same checklist. Reflect on how much time and effort it saved you!



4 Advanced



The new checkbox, label, and textbox.

Create a copy of HelloWorldSwing.java as HelloWorldSwing3.java. Make sure you you're your original copy of HelloWorldSwing.java as you will need to submit it. Modify HelloWorldSwing3.java to add the following NEW functionality!

4.1 Checklist

- [] Additional checkbox, label, and textbox.
- [] Only integer values 1 to 10 are allowed in the textbox (otherwise give error message)
- [] If checkbox is selected, then AFTER clicking the "Show!" button, the display (JLabel in the center) is cleared AFTER the number of seconds that had been entered into the textbox.
- [] If the checkbox is deselected, then the value in the textbox has no impact on the display.

5 Submission

Zip the three programs (HelloWorldSwing.java, HelloWorldSwing.java 2, HelloWorldSwing3.java) and upload to blackboard (for your group).