CST8221 – JAP
Lab 4 – Choice – Check boxes and Radio buttons

Objectives

The purpose of the lab is to give you a hand-on experience of how to create and use check boxes and radio buttons.

The Nature of Things

To this point, you have seen and tried regular buttons many many times. Once you click on a button it returns to its original state immediately. We can say that a regular button cannot stay in more than one state, that is, it is *stateless*. Sometimes we need components that can have more than one state. Using a combination of these components allows the developer to create a complex choice interface.

Swing provides two important types of ***state buttons*: *JCheckBox*** and ***JRadioButton***. Both components have *on/off* or *true/false* states. Which means that a cluster of two check boxes provides the user with 4 different choices (why?)? Usually, several radio buttons are grouped together in a *button group* and are mutually exclusive – only one in the group can be selected in any given time, just like the buttons on a radio. A group of two radio buttons provides the user with two mutually exclusive options. Since *JCheckBox* inherits from *JToggleButton*, the checkbox can also be included in a button group but that feature is used in special cases only.

JavaFX provides similar controls: ***CheckBox***, ***RadioButton***, ***ToggleGroup***. The difference between Swing and JavaFX is that a *CheckBox* cannot be part of a toggle group but If needed, the mutually exclusive behavior can be simulated through event handling.

In this lab you will try four similar GUI applications – two using check boxes and the other two using radio buttons.

Tasks

Download the ***CST8221_Lab4_code.zip*** file from Blackboard. Extract the contents. Four different code examples are provided for you: ***CheckBox***, ***RadioButton***, ***CheckBoxFX***, and ***RadioButtonFX***. Run the examples one by one and examine carefully the code. The FX examples are interdependent – the radio buttons example needs the check box example java files to compile and run. Try to identify the components involved in the GUI, and try to identify the code segments involved in processing the events generated by the user actions. Pay attention to how the radio buttons are grouped together.

Try to understand as much as possible how the programs work. Consult the Java documentation to find out what the different classes and methods are implementing. Change the event handler in the ***CheckBox*** example so that it implements an *ActionListener* instead of *ItemListener*.

Pay special attention to the Swing examples. They will help you with the implementation of your first assignment. Pay special attention to the ***getActionCommand()*** method in the event handler of the ***RadioButton*** example.

Before the lab
Enjoy Java.

During the lab
Ask questions.

Before leaving the lab
Sign the attendance sheet.

After the lab
Remember what you have learned. You will need it for your first assignment.

Submission
No submission is required for this lab.

Marks
No marks are allocated for this lab, but do not forget that the joy of learning is priceless.

And do not forget that:

“A man is happy so long as he chooses to be happy.” Alexander Solzhenitsyn

but

“Does having too many choices make us less happy? “

Barry Schwartz’s *The Paradox of Choice: Why More Is Less*