Exercise03: Java Swing+Server/Client+Threads

Objectives:

- To use basic JTable.
- To learn to use Threads
- To learn to write server client code

Work with your group (or by yourself). Each group should upload only one submission.

First, open blackboard, go to Course Contents, and then download exercise03.zip file into your workspace (U:\workspace or something like that!). Then, unzip.

1 Part 1: JTable

1: Run Eclipse and make sure that the workspace is set to U:\workspace (your COMS home directory).

2. Go to File->New->Project->Java Project. For the project name, type in “Lab3-Swing” and click Finish. You should see the project “Lab3-Swing” built and shown in the “Package Explorer”.

We have NOT read or learnt about JTable. However, there are lots of great tutorials on the web. Here is an example http://docs.oracle.com/javase/tutorial/uiswing/components/table.html
We will assume you will be able to read and learn to use JTable (which is similar to JList and JTree) from these.

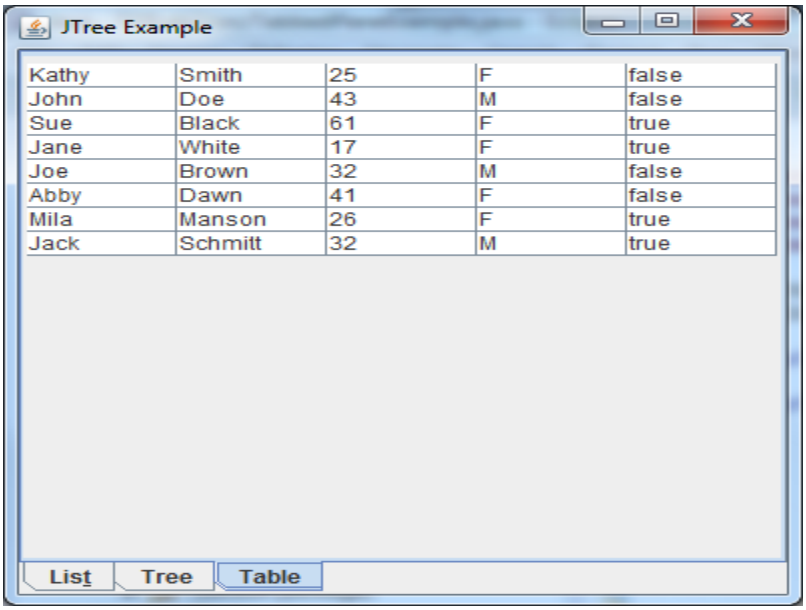
The objective in this part is to create a Table (i.e. JTable component) in a JPanel to read and show some contents. The following are the columns of the table:

```
String[] columnNames = {"First Name","Last Name","Age","Gender","Vegetarian"};
```

And the following are the contents that are to be shown in the table:

```
Object dataValues[][] = {
    {"Kathy", "Smith", 25, 'F', false}
    {"John", "Doe", 43, 'M', false}
    {"Sue", "Black", 61, 'F', true}
    {"Jane", "White", 17, 'F', true}
    {"Joe", "Brown", 32, 'M', false}
    {"Abby", "Dawn", 41, 'F', false}
    {"Mila", "Manson", 26, 'F', true}
    {"Jack", "Schmitt", 32, 'M', true}
}
```

Make sure the table is scrollable (make use of JScrollPane). The end result would like the following:



CheckList for JTable

[ ] (YOU CAN IGNORE THE TABS FROM PREVIOUS EXERCISE)

[ ] JTable is there and is scrollable

[ ] JTable has all five columns

[ ] JTable shows the contents properly

[ ] JTable looks similar to the given figure

[ ] Extra credit: make the table sync with a CSV (comma separated values) file (using model etc.)

2 PLAYING WITH Threads & Server/Client

2.1 Threads

Play with the programs in "threadExamples".

There are four examples.

1. The first one shows how to create Threads by extending the Thread class.
2. The second example shows a great alternative using implementations of the Runnable interface.
3. The third example shows problems when sharing modifiable data between threads.
4. The fourth example shows how such problems can be handled in Java by using the synchronized keyword that makes a thread that is currently doing the method to complete it before letting other methods start the method.
5. There is a lot more to threading – but that is outside the scope of this class. This info should be enough to get you started.

2.2 Server Client

Play with the programs in "serverClientExamples".

There are two programs.

1. “MyServer.java” shows code for a sample Server program. Run it on Eclipse. If you run it multiple times, the later runs will all fail because port 4444 is already being reserved by the first run.
2. “MyClient.java” shows code for a sample client program. Run it several times. Each time a client program is run, it sends data to the server - which prints out the information on the screen.
3. Don’t forget to kill the server program after you are done playing with the code.

3 Part 2: Server Client/Thread/JList

Take the JList code that you wrote for section 2.2 of Exercise02 and modify it so that

a) The JList is managed by the client code.
b) "companies.txt" is managed by the SERVER side clientHandler code.
c) In other words, when you add to the JList, the client sends data to the server which adds the data to the "companies.txt" file.

4 Part 3: Extra Credit

EXTRA CREDIT: The server side should also allow some way to add/remove data to/from the "companies.txt" file. On doing so, the server sends data back to the client which in turns fixes the JList view.

In other words, changes to the JList on the client side should change companies.txt on the server side. ALSO, changes to companies.txt from the server side should change JList view on the client side.

HINT-1: To have the CLIENT not "hang" waiting for the server to send messages, you will need to create a "server handling" thread whose sole purpose is to listen for messages from the server.

HINT-2: To keep to GUI single event-handling thread rules, the above "server handling thread" should use SwingUtilities.invokeLater to queue any GUI related task.

5 Submission:

Zip your Eclipse project and submit on black board. Remember there is only one submission per group. Make sure to include all the files that are needed in order to run your program(s).