

Exercise03: Server/Client+Threads

Objectives:

- 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 PLAYING WITH Threads & Server/Client

1.1 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 an 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.

1.2 Server Client

Read serverClient.pptx file which located in the main root.

Play with the programs in "serverClientExamples".

There are two programs.

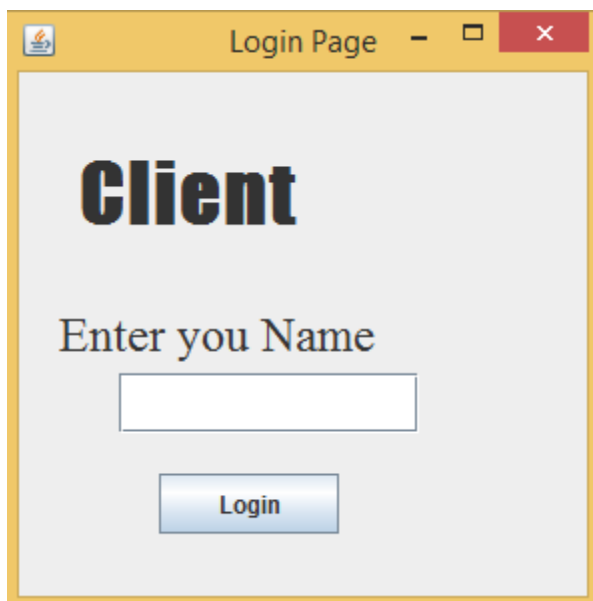
1. "MyServer.java" shows code for a sample Server program. Run it on Eclipse. If you run it multiple times, the later runs will all fail because port 4444 is already being reserved by the first run.
2. "MyClient.java" shows code for a sample client program. Run it several times. Each time a client program 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. For killing the server you need to press the red square in the Eclipse.



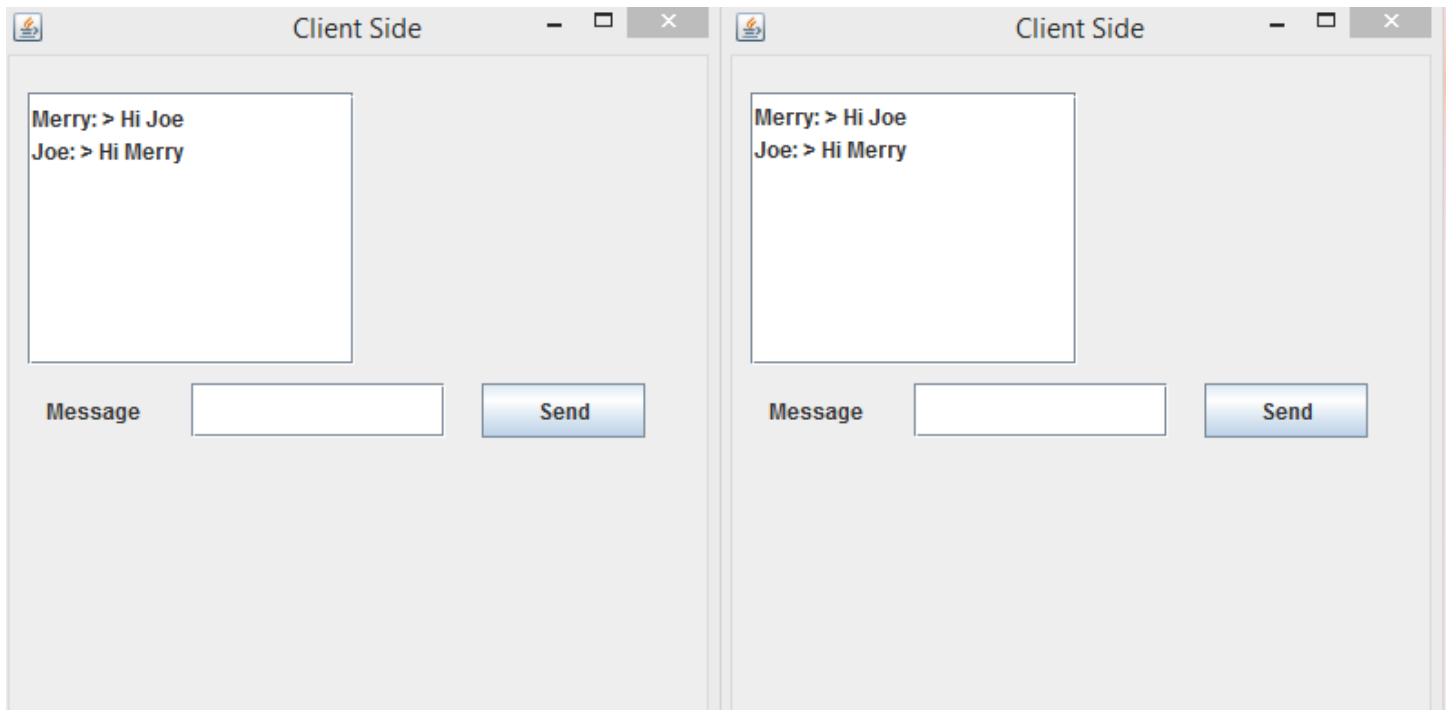
2 Part 2: Server Client/Thread

Create a chat application using Server Client.

- a) Create a login page similar to following:



- b) After pressing Login button, client should be connected to server and send message.



c) The server should keep history of chat in "chat.txt".

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.

3 Part 3: Extra Credit

EXTRA CREDIT: The server side should also allow some way (with UI/ Console) to send data to all of clients. If somebody using UI/Console server and submit a message, the message should be broadcast to all of clients.

4 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).