CS3331

Karla Chavez

Ericka Najera

Due: 10/29/2019

Dr. Mejia

HW3

2.

a.) First, we created the client class, the code is in the main. The first step was to create a socket to connect the client and server, once the socket is connected the user can start typing, we used DataInputString to get the input from the server and DataOutputStream to send data to server, we also used the BufferedReader to get input from user. Our code uses two empty strings and a while loop, since goodbye is the condition to terminate the program, as long as the user does not input goodbye the while loop will continue, each input from user will be saved in the first string and then be sent to the server(using DataOutputStream). We will get the response from the server using DataInputStream and we will save it on the second string. We use system out print to print the responses of the server and once the while loop is terminated we close the DataOutputStream.

Next, we created the server class; the code is also in the main. We created a ServerSocket and a Socket that accepts the serverSocket, we also use DataInputStream,DataOutputStream and BufferedReader similarly to the client class. This class only has one string, and we also use a while loop that will continue until the string equals goodbye. Inside the while loop we use an if statement that checks if the string received by the client contains the word time, if this is the case, we return the local time using DataOutputStream to the client and flush. If the if statement condition is not met we will return the same string we received to the client and also flush. Finally after the while loop is terminated: we close the DataOutputStream, the serverSocket and Socket.

b.) One of the issues we encounter was that the program wouldn’t behave in the way we expected if we didn’t flush every single time we used dataOutputStream, so we had to add it right after the used writeUTF. Additionally, in the client method we had problems to printing the word Goodbye after we received it from the server, although the client class would receive the word goodbye would still print the time at the end, we fixed this by adding an if statement in the while loop of the client class and once the first string contained goodbye we would make the second string equal goodbye, this way it would be printed at the end when we print what the server says.

c.) We planned what we wanted the code to look like for about thirty minutes, we did a little bit of research for about forty five minutes, and then finished the code in about two hours. We finished in about three hours and a half. We did ran into some issues as explained in part b.

3. Sources

<https://www.javatpoint.com/socket-programming>

<https://cs.lmu.edu/~ray/notes/javanetexamples/>

<https://www.baeldung.com/a-guide-to-java-sockets>

<https://www.geeksforgeeks.org/introducing-threads-socket-programming-java/>