

## Project #0: TCP Socket Programming

이름 : 비토리아

학번 : 2020049061

For Project 0 we needed to Implement a TCP Socket program. This program would be able to communicate with a remote process by exchanging data with TCP/IP by knowing the combination of protocol type, IP address, and port number.

**Source Files:** The programs were implemented in Java

TCPServer.java

```
1④ import java.io.*;
2 import java.net.*;
3
4 class TCPserver {
5④     public static void main(String argv[]) throws Exception{
6         String clientSentence;
7         String capitalizedSentence;
8         ServerSocket welcomeSocket = new ServerSocket(6092);
9
10        while(true) {
11            Socket connectionSocket = welcomeSocket.accept();
12
13            BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));
14
15            DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());
16
17            clientSentence = inFromClient.readLine();
18            capitalizedSentence = clientSentence.toUpperCase()+'\n';
19            outToClient.writeBytes(capitalizedSentence);
20        }
21
22    }
23 }
```

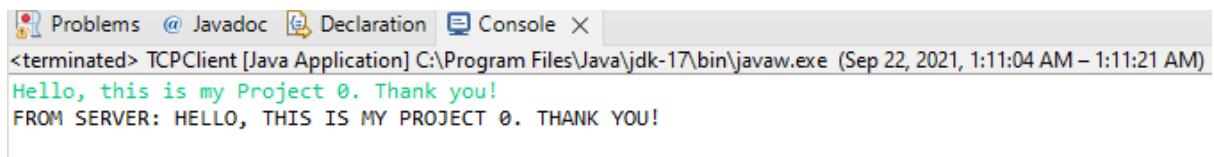
TCPClient.java

```
1④ import java.io.*;
2 import java.net.*;
3
4 class TCPClient {
5④     public static void main(String argv[]) throws Exception{
6         String sentence;
7         String modifiedSentence;
8
9         BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
10
11         Socket clientSocket = new Socket("localhost", 6092);
12
13         DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
14
15         BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
16
17         sentence = inFromUser.readLine();
18         outToServer.writeBytes(sentence + '\n');
19
20         modifiedSentence = inFromServer.readLine();
21         System.out.println("FROM SERVER: " + modifiedSentence);
22
23         clientSocket.close();
24    }
25 }
```

**Instructions:** To make it work properly, first we run the TCP Server program, so the connection between the address and ports can be done. After running the TCP Server, we will run the TCP Client and input the sentence. Also is important to quit the process after finishing running the program, otherwise, when we try to run the program again, the port number won't be available.

**How the program works:** Server TCP application will create a socket with the selected port and will create a 'welcomeSocket' variable to save the request that will be sent by the client. In Client TCP we also create a socket to receive the data from the server. Then Server will wait for the incoming connection request from the client. The client sends a request to the server using the client's socket. After accepting the request from 'welcomeSocket', 'connectionSocket' will have its request read, in this case, the input sentence, then it will write a reply to 'connectionSocket' that will be read by the client socket, and print the sentence sent by Server Socket, capitalized.

**Results:** The result screen



The screenshot shows a Java IDE's console tab. The title bar says 'Problems @ Javadoc Declaration Console X'. The console tab is active. The output window displays the following text:  
<terminated> TCPClient [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Sep 22, 2021, 1:11:04 AM – 1:11:21 AM)  
Hello, this is my Project 0. Thank you!  
FROM SERVER: HELLO, THIS IS MY PROJECT 0. THANK YOU!

**Other comments:** It was interesting to run the code testing different ports and see how it works. And also was interesting to see that when the program is running we occupy a port, if we don't end the process in the computer system manager, that same port number will be unavailable if we run the program again.