COMP 416

PROJECT 1-1 REPORT
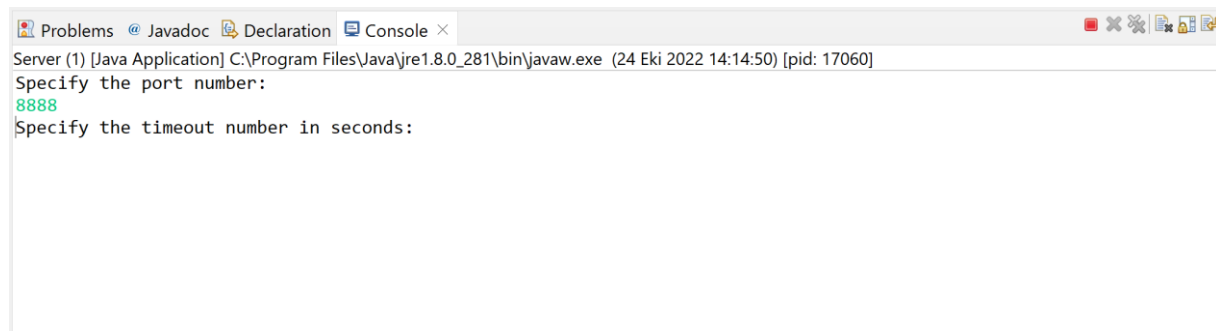
Meryem Karakaş   69074

ReadMe:

First, you must run the server program (Server.java). When you run it, it asks for the host port number. After giving the port number, it asks for the timeout number in seconds. This time is converted to milliseconds when given as a parameter to the setSoTimeout method in the server program. The timeout number indicates in how many seconds a client should send the first message after the server accepts the client's connection request. If the client does not send the first message within timeout seconds, the server gives warning and the server program is terminated. If the client tries to send a message after the time has expired, the client program will issue a warning and be terminated.

After giving the timeout number to the server program, you should run the client program (Client.java). When you run the program, it first asks for the host's port number. When you give the port number, the program asks for the host's address. If you want to connect to a server on the same device, you must type "localhost". However, if you want to connect with a different device, you can give the address of the device you want to connect to. I asked the user for the host address in order to establish connections with other computers.
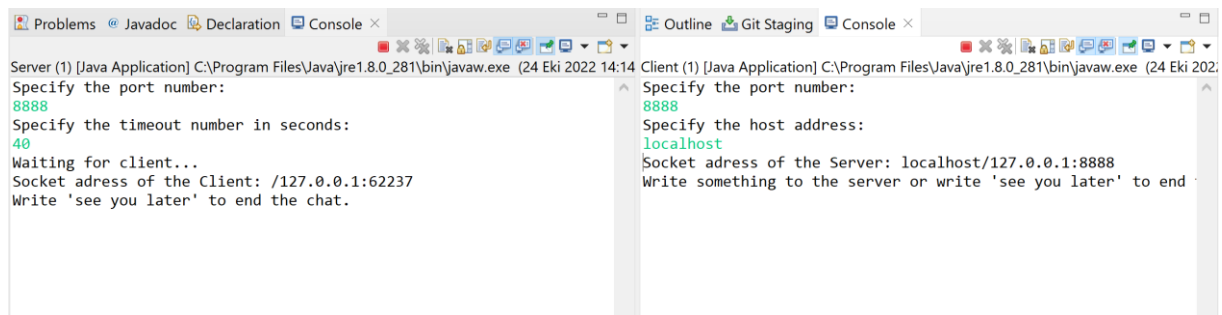
After both programs enter their inputs as I mentioned above, the server responds to the connection request. If a successful connection is established, both programs print the socket address of the other party to the screen. Then, in order for the two parties to chat, the client must send the first message within the time given to the server program by the user. If the client sends the first message within the valid time, the chat will be started. In this case, the two parties can send a message to each other in turn. When either party wants to end the conversation, that party should write "see you later". When this keyword is typed, both programs and the chat are terminated.
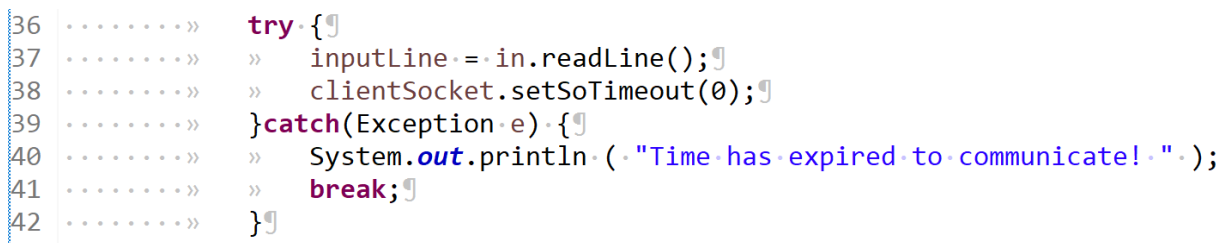
1.

2.

3.

After connection has established, I used setSoTimeout method with parameter timeout multiplied by 1000 (to convert seconds given by user to milliseconds).

```
Server.java ×    Client.java
26 ········Socket·clientSocket·=·serverSocket.accept();···»»
27 ····»    clientSocket.setSoTimeout(timeout*1000);·//converting·seconds·to·milliseconds
28 ····»    PrintWriter·out·=·new·PrintWriter(clientSocket.getOutputStream(),·true);
```
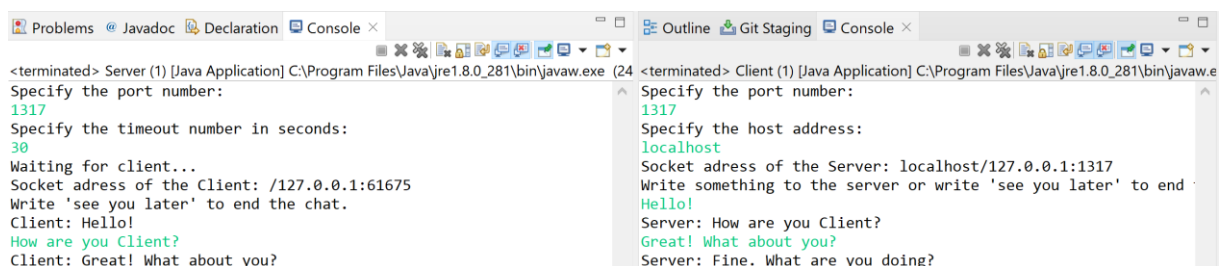
If the server can receive first message before time ends, I disable timeout by calling setSoTimeout method with parameter zero. Thus, a time limit is created for only the client's first message. If the first message is sent within the time limit, smooth chat is possible after that.

```
36 ········»    try·{
37 ········»    »    inputLine·=·in.readLine();
38 ········»    »    clientSocket.setSoTimeout(0);
39 ········»    }catch(Exception·e)·{
40 ········»    »    System.out.println(·"Time·has·expired·to·communicate!·"·);
41 ········»    »    break;
42 ········»    }
```

Since timeout is not required for the client, I only adjusted timeout settings for the connected socket in the server code.
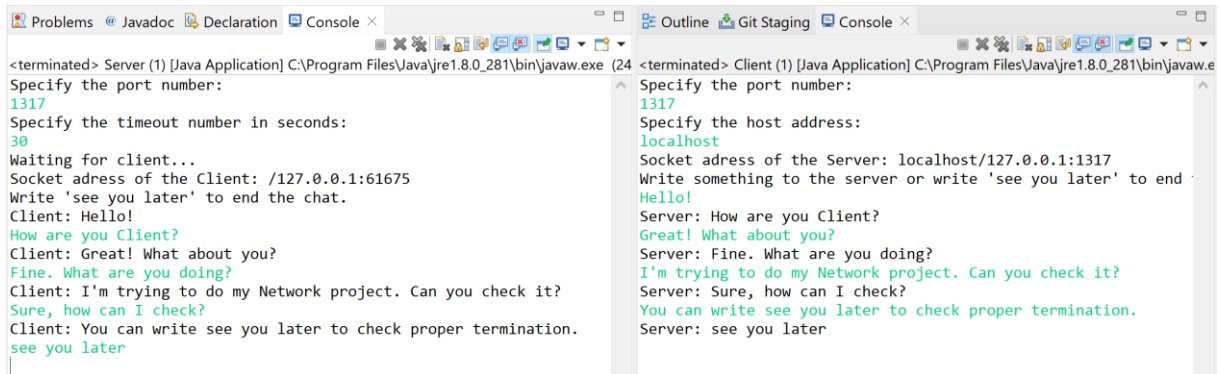
4.

5.

```
<terminated> Server (1) [Java Application] C:\Program Files\Java\jre1.8.0_281\bin\javaw.exe (24
Specify the port number:
1317
Specify the timeout number in seconds:
30
Waiting for client...
Socket adress of the Client: /127.0.0.1:61675
Write 'see you later' to end the chat.
Client: Hello!
How are you Client?
Client: Great! What about you?
Fine. What are you doing?
Client: I'm trying to do my Network project. Can you check it?
Sure, how can I check?
Client: You can write see you later to check proper termination.
see you later
```
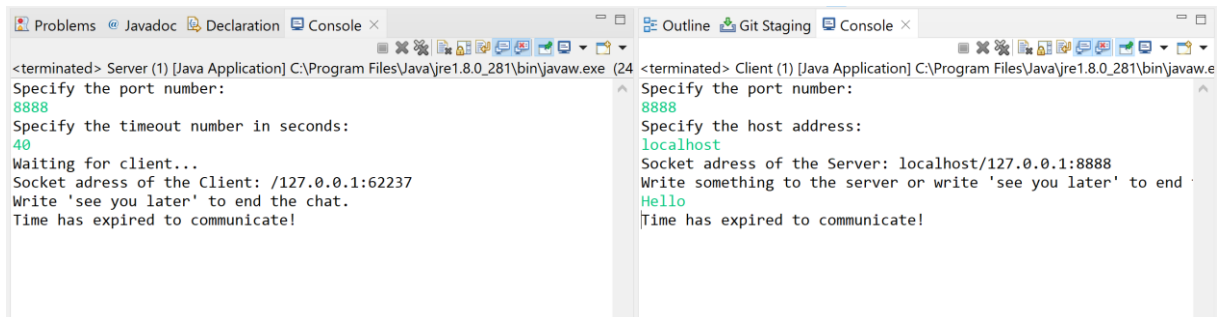
```
<terminated> Client (1) [Java Application] C:\Program Files\Java\jre1.8.0_281\bin\javaw.e
Specify the port number:
1317
Specify the host address:
localhost
Socket adress of the Server: localhost/127.0.0.1:1317
Write something to the server or write 'see you later' to end
Hello!
Server: How are you Client?
Great! What about you?
Server: Fine. What are you doing?
I'm trying to do my Network project. Can you check it?
Server: Sure, how can I check?
You can write see you later to check proper termination.
Server: see you later
```

6.

```
<terminated> Server (1) [Java Application] C:\Program Files\Java\jre1.8.0_281\bin\javaw.exe (24
Specify the port number:
8888
Specify the timeout number in seconds:
40
Waiting for client...
Socket adress of the Client: /127.0.0.1:62237
Write 'see you later' to end the chat.
Time has expired to communicate!
```

```
<terminated> Client (1) [Java Application] C:\Program Files\Java\jre1.8.0_281\bin\javaw.e
Specify the port number:
8888
Specify the host address:
localhost
Socket adress of the Server: localhost/127.0.0.1:8888
Write something to the server or write 'see you later' to end
Hello
Time has expired to communicate!
```