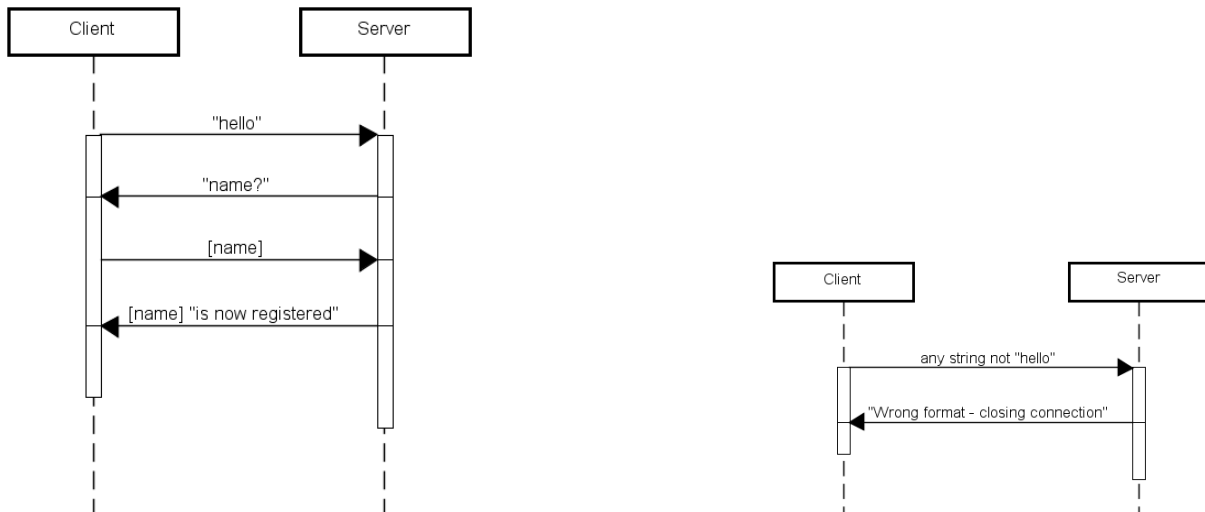


Exercises session 5, SDJ2-S18

Exercise 5.01 – Register yourselves for SDJ lessons today

Implement a client application using Sockets with TCP and connected to a server on the wireless network VIAnet. The server computer has the IP address 10.152.144.13 and the application uses port 2910. The client application has to follow the communication flow shown below (the left diagram represents a correct communication and the right diagram the exception flow):



Note that the client have to send a string "hello" to the server and for any other string the server will send an error message and close the connection after 10 seconds. The server will for a correct string send a string "name?" and the client has to send his name as a string to the server. Finally, the server sends a message that the client has been registered.

Also note that you should use `DataInputStream` for input stream and `DataOutputStream` for output.

The information will be logged on the server including a timestamp and thus, use your correct name and the server manager can always see if you were present in class today.

The user print on the client computer could be the following (bold is keyboard input)

```
Type your name: Bob the Builder
Bob the Builder> hello
Server> name?
Bob the Builder> Bob the Builder
Server> Bob the Builder is now registered
```

Exercise 5.02 – Write the server application

Write the server application for the program in the previous exercise. You should make a multi-threaded server (but it is legal to make a version 1 with the server application as a simple TCP Server).

You can get inspired by the following code snippet to log the client IP and host name on the server:

```
ServerSocket welcomeSocket = new ServerSocket(PORT);
Socket connectionSocket = welcomeSocket.accept();
String client = connectionSocket.getInetAddress().getHostName();
```

```
+ " / " + connectionSocket.getInetAddress().getHostAddress();
```

Exercise 5.03 – Group work

Work in a group for this exercise in which you have to do the following:

- 1) Agree on a communication flow like the diagram in Exercise S7.01 where you send/receive at least one value of primitive type and at least one String. Write the communication flow (diagram) on paper, in Astah or in any other file format
- 2) Make one or more diagrams for exception flows
- 3) Split up the group with one part implementing the server and the other part implementing the client
- 4) Run the server and the client's. Be sure to test the correct communication and the exception flows
- 5) Extra: use methods on the Socket and ServerSocket classes to log on the server side the port number, local port number and IP and host name for the client computer. Run multiple client applications (on the same or on different computers) to observe what happens

Exercise 5.04

Implement a TCP based Client/Server application where the server as a service offers to calculate the sum of two integer values. The communication between client and server is the following; the client sends two integers to the server and the server calculates the sum and returns this to the client.

Exercise 5.05

Implement a UDP based Client/Server application where a client can contact a server to get the current time. The client sends a datagram to the server including the string "time" and in case of the correct string in the datagram the server sends to the client the current time as a long
(`GregorianCalendar.getInstance().getTimeInMillis()`).