

Abstract TCP Server – part 1

Mission

To build a framework for ease the implementation of TCP servers, this include concurrent server.

Background

Straightforward TCP server programming from 3rd semester was in Python, to make a TCP server in C# is similar – you can see here:

- [Simple tcp server](#)
- [Concurrent tcp server](#)

For download a running TCP-server (Echo server) and the Client

<https://github.com/RO23F-VF-ASWC/TCPEchoServer>

Template Design Pattern see

- [C#Note](#) chap. OOProg3 pp.1-17 + 43-47
- [Dotfactory template in C#](#)
- [C# design patterns in general](#) (extra)

Warning up

Assignment 1 – To get some start-up coding - A simple Server

Create a .Net Core Console application and create a simple Echo TCP server on port 7007

Ensure your server is concurrent by using a Task.Run to execute each client.

You are NOT asked to implement a Client-program.

Try your implementation with SocketTest:

Download and unzip SocketTest : <https://sourceforge.net/projects/sockettest/>

If SocketTest do not work, you possible need a Java Runtime Environment so download and install jre : <https://www.java.com/en/download/>

Now you start making a framework for TCP-servers

Assignment 2 – Make a Simple Framework for TCP Servers

Now you are to build a simple **Framework** supporting easy creation of a TCP-server.

This means you are making classes in a class library, which others (programmers) can make use of.

Create a **Class Library** and make a folder “TCPServer”.

In this folder, create an **abstract** class ‘AbstractTCPServer’, where you insert most of the code from the example from assignment 1.

The only exception is the part where you read from the StreamReader and write the string back to the StreamWriter.

This part should be extracted into a method ‘TcpServerWork’, with the reader- and writer-stream as parameters. This method should be abstract. (THE TEMPLATE METHOD)

Assignment 3 – Improve the Simple Framework

You can improve your AbstractTCPServer by:

- Pass the port number as a parameter to the AbstractTCPServer
- Pass a name to the Server e.g. “EchoServer”
- Printout state information like “started at port xxxx” etc. – In a later version you are to log or trace this information instead.

Here you try out you framework

Assignment 4 – Use your Simple Framework

Create a new .Net Core Console Application project / or solution.

In this project make a folder ‘Server’, where you create a new class e.g. ‘MyServer’, that inherit from the abstract server from the Library. (I.e. add reference to your Library).

Implement the abstract method – again as echo, perhaps by capitalize the letters.

In the main, make an object of your Server and start it. – **Try your server with SocketTest**

I hope that you find it easier this time to create a server.

Extra

Make a similar framework but for TCP-clients

Make a similar framework for UDP-server / client