

Server Framework – 1

(template)

Peter Levinsky IT, Roskilde

03.02.2025

Next two week

Working with a framework for TCP-servers

This week:

- Class library
- Template/abstract server class
- Framework = comments
- Soft-closedown of server

Next Week

- Tracing and Logging
- Configuration of the server using XML

Template/abstract server class

Design Pattern

- Summary of good programming experience – Best practise
- Terminology among programmers

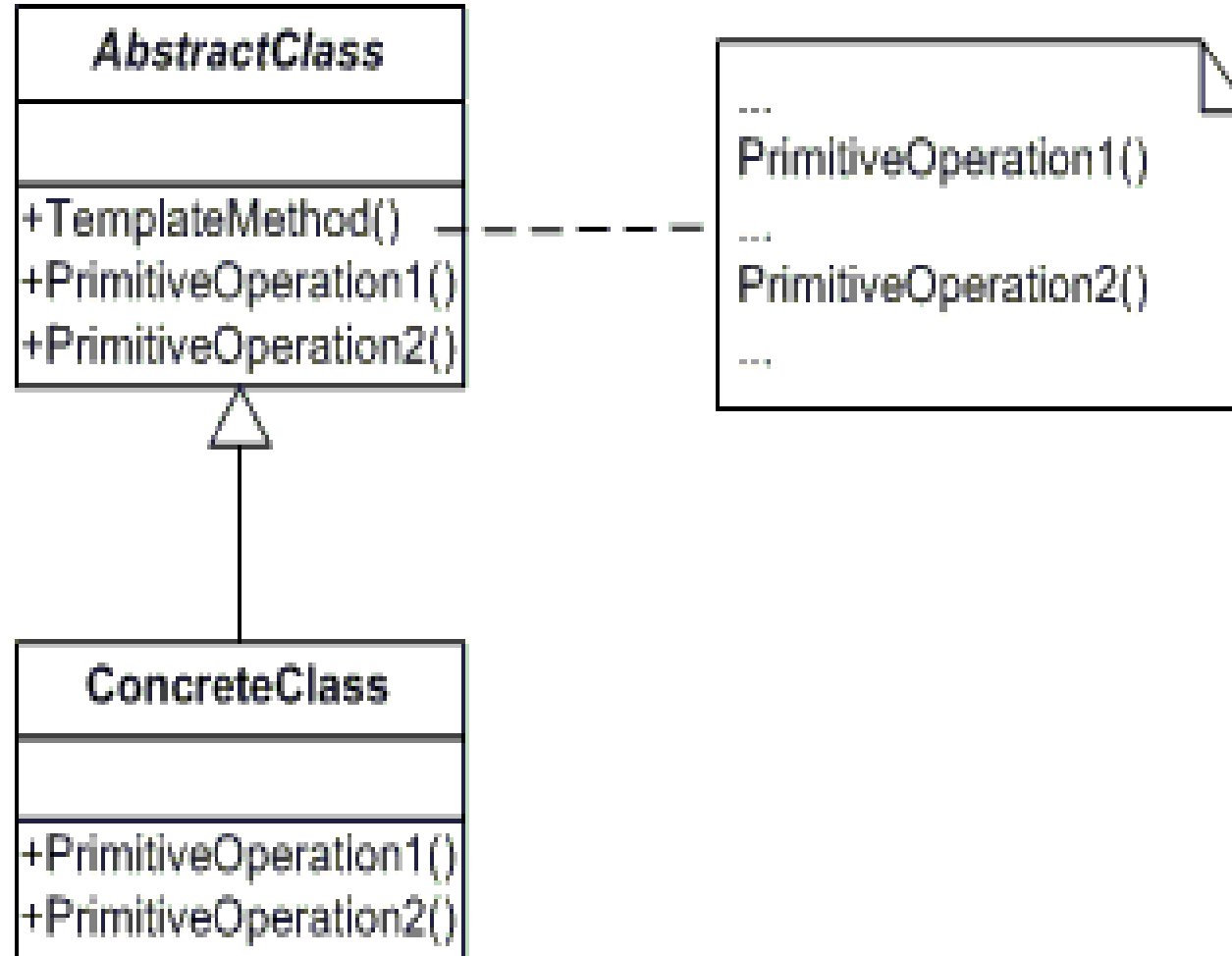
They are described as

- **Name:** Can talk of them ~ like Singleton
- **Problem:** What problem will they solve?
- **Solution:** A design solution – not a programming language specific solution – although OOP

Template/abstract server class

Name: Template

Problem: Most of the algorithm are the same –
Only a small part vary



Demo

Very basic - The template design Pattern

Simple TCP-Server in C#

```
public void Start()
{
    TcpListener listener = new TcpListener(IPAddress.Any, PORT);
    listener.Start();

    while (true)
    {
        TcpClient client = listener.AcceptTcpClient();

        Task.Run(() =>
        {
            TcpClient tmpClient = client;
            DoOneClient(client);
        });
    }
}
```

```
private void DoOneClient(TcpClient sock)
{
    using (StreamReader sr = new StreamReader(sock.GetStream()))
    using (StreamWriter sw = new StreamWriter(sock.GetStream()))
    {
        sw.AutoFlush = true;
        Console.WriteLine("Handle one client");

        // simple echo
        String? s = sr.ReadLine();
        sw.WriteLine(s);
    }
}
```

Your turn



Thursday – Comments / soft closing