Server Framework

Peter Levinsky IT, Roskilde

01.02.2021

Next two week
Working with a framework for TCP-servers

- Class library
- Template/abstract server class
- Framework = comments
- Soft-closedown of server
- Tracing and Logging
- Configuration of the server

Template/abstract server class

Design Pattern

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

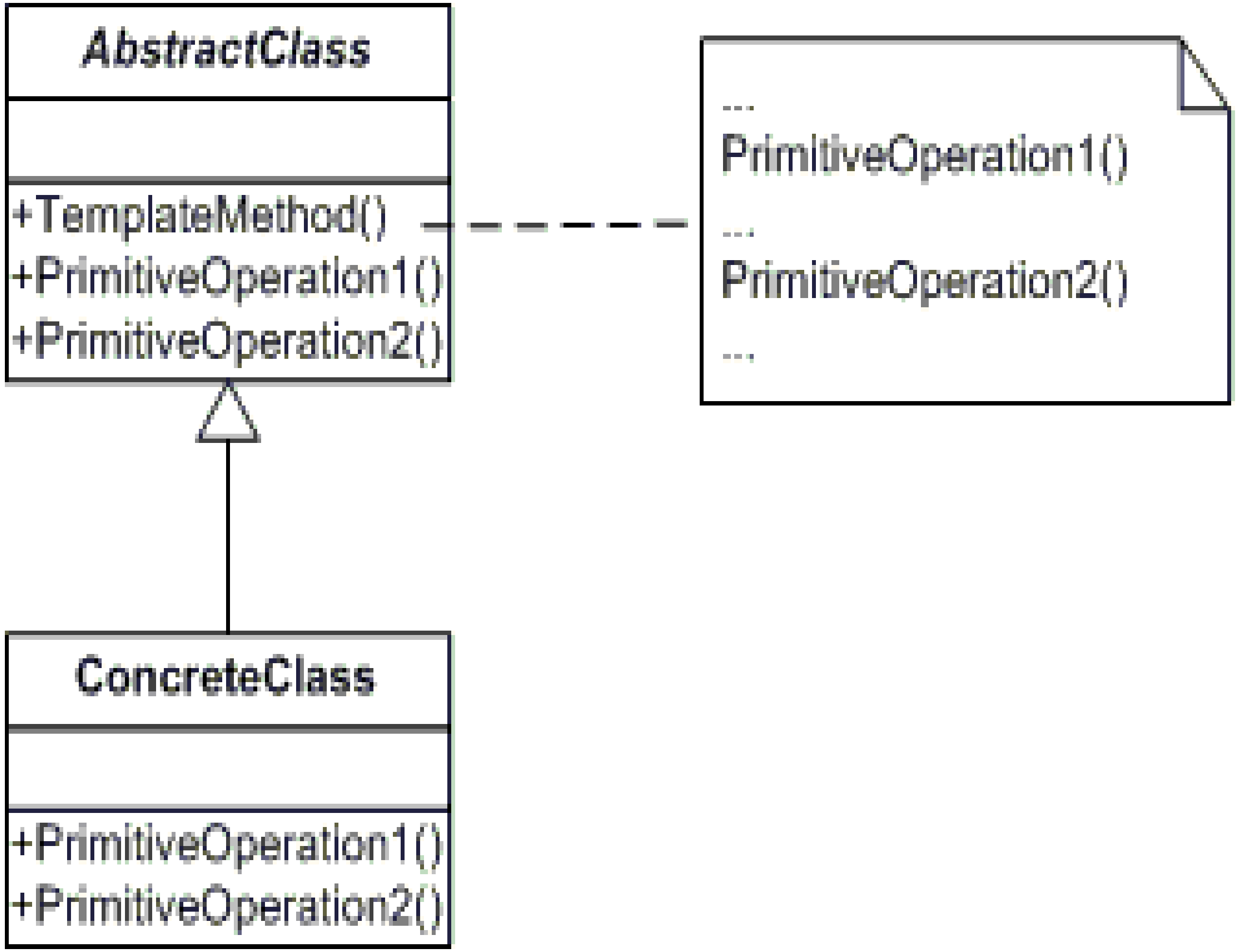
They are described as

- **Name**: Can talk of them ~ like Singleton / Observer
- **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

https://github.com/RO21F-VF-ASWC/ClassDemoTemplate

Soft Closing the Server

Until now Brute Force shutdown.
Like Soft Shutdown.

1. Introduce a bool field Stop – initial false
2. Let the while-loop condition be this field (not always true)
3. Implement another method – which is a new server e.g. listen at port ServerPort + 1
4. Start this stop-server in a separate thread (task)
5. Within the while-loop before AcceptTcpClient – ask if any connection is pending otherwise wait e.g. 2 sec.

Demo

Very basic – A simple Server

https://github.com/RO21F-VF-ASWC/ClassDemoTemplate

Commenting code

Type	Example	Purpose
Single Line	// this I a comment	For maintenance programmers
Multiple Lines	/* * This is a comment …. */	For maintenance programmers
Structured XML comments	/// <summary> /// Here comes some text /// </summary> /// <returns>Text of return type</returns>	For Documentation of code for programmers to use the classes; Typical API's
Special Variation for Visual Studio	//todo //hack	For maintenance programmers

XML-Comments

```
/// <summary>
/// return a multiplication of x and the number within n
/// </summary>
/// <param name="x">The value to be multiplied</param>
/// <param name="n">The value of the figure to be multiplied must be '2' or '3'</param>
///<exception cref="System.ArgumentNullException">Thrown when n is null or empty</exception>
///<exception cref="System.ArgumentException">Thrown when n is not '2' or '3'</exception>
/// <returns>The value x multiplied by two or three depending on the value in n</returns>
public int SomeMethod(int x, String n)
 {
     if (string.IsNullOrWhiteSpace(n)) throw new ArgumentNullException("n must have a value but was null or empty");
     if (! (n=="2" || n=="3")) throw new ArgumentException("Only '2' or '3' is supported but was " + n);

     switch (n)
     {
         case "2" : return x * 2;
         case "3" : return x * 3;
     }

     throw new NotImplementedException("");
 }
```

Doxygen – Make homepage of documentation

Download and install Doxygen

Generate Config file
…> Doxygen –g "nameOfConfigFile"

Generate documentation
…> Doxygen "nameOfConfigFile"

Show documentation
In …<projectFolder>/html/index.html

Demo

Very basic – a simple Library with a (nonsens method)

https://github.com/RO21F-VF-ASWC/ClassDemoTemplate