# Exercises: High-Quality Classes

This document defines the **in-class exercises** assignments for the ["High-Quality Code" course @ Software University](https://softuni.bg/courses/high-quality-code).

## Large Project

Examine a large C# project. You can look in GitHub – search for the projects whose language is C#, sort them by number of forks (in descending order) and choose one of the first few projects. The project you have chosen to examine should contain at least 30 classes. Fill in the table below, documenting your findings.

Document anything you like (or don't like) in the code.

|  |  |  |
| --- | --- | --- |
| **Class / Method** | **Link** | **Notes** |
|  | <http://referencesource.microsoft.com/#mscorlib/system/io/file.cs> | Classes have separated their concerns, the abstraction levels are consistent… |
|  |  | Interfaces are small, each interface has its own responsibilities |
| … | … | … |

Some things to look for:

* OOP principles
* Validation and exception handling
* Correct abstraction levels

You can additionally look for things you have already learned:

* Code formatting
* Clear naming – variables, methods, namespaces, etc.
* Documentation and comments
* Variable usage
* Deep nesting of control structures
* Straight-line code
* Strong cohesion and loose coupling

## \* Design Patterns

Try to find any design patterns in the code you have already examined in Problem 1. Document your findings in the table below:

|  |  |  |
| --- | --- | --- |
| **Design pattern** | **Example** | **Description** |
| Singleton | <https://github.com/apache/log4net/blob/trunk/src/log4net/LogManager.cs> | The class **LogManager** uses Singleton because there must be only one instance of the logging manager per appication |
| … | … | … |