MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

IVAN FRANKO NATIONAL UNIVERSITY OF LVIV

Department
of Foreign Languages
for Sciences

SUMMARY

Clean C++- Sustainable Software Development Patterns and Best Practices with C++ 17

by Stephan Roth
Apress Media LLC, New York, 2017, 1-72 pp.

Submitted by
the student of the Faculty of
Applied Mathematics and
Informatics
O. Kravets

Lviv-2021

In the passage I've read, I learned a lot of new and instructive things. One of the main things with are discussed in this passage is proper software testing. That is, writing the correct unit tests. Also, I paid special attention to, is that it is not the correct or incomplete testing of the software can lead to very unpleasant consequences.

The author also depicted software testing in the form of a pyramid. Simple unit tests were depicted below. This is because a lot of unit tests are needed to test the program, but it is fast and cheap. But at the top of the pyramid was an automated GUI test. This test is expensive to develop and very slow to run, but fully tests the program for all scenarios.

The author also told about the advantages of such a programming language as C ++. One of them is that in this programming language you can easily perform difficult projects and low-level programming.

The first chapter deals with the external and internal quality of the code. In second chapter the author describe why he think that Unit Tests, especially, are indispensable to ensure a fundamental level of excellent quality in software. In third chapter, Stephan Roth introduce the most important and fundamental principles of well-designed and wellcrafted software. What is special about these principles is that they are not tied to certain programming paradigms or programming languages. Some of them are not even specific to software development. These advices are given for you to internalize. In fourth chapter author describe the general basics of clean C++. These are sometimes universal things that are often programming language independent. For example, giving a good name is essential in all programming languages.

This book is a must for every developer, software architect, or team leader who is interested in good C++ code, and thus also wants to save development costs. If you want to teach yourself about writing clean C++, Clean C++ is exactly what you need. It is written to help C++ developers of all skill levels and shows by example how to write understandable, flexible, maintainable, and efficient C++ code. Even if you are a seasoned C++ developer, there are nuggets and data points in this book that you will find useful in your work.

If you don't take care with your code, you can produce a large, messy, and unmaintainable beast in any programming language. However, C++ projects in particular are prone to be messy and tend to slip into bad habits. Lots of C++ code that is written today looks as if it was written in the 1980s.

It seems that C++ developers have been forgotten by those who preach Software Craftsmanship and Clean Code principles. The Web is full of bad, but apparently very fast and highly optimized C++ code examples, with cruel syntax that completely ignores elementary principles of good design and well-written code. This book will explain how to avoid this scenario and how to get the most out of your C++ code. You'll find your coding becomes more efficient and, importantly, more fun.