Chapter 1 - Clean Code

This Book is about good programming. It's about how to write good code, and how to transform bad code into good code.

The code represents the detail of the requirements and the details cannot be ignored or abstracted. We may create languages that are closer to the requirements. We can create tools that help us parse and assemble those requirements into formal structures. But we will never eliminate necessary precision.

Why write bad code?

- Are you in a rush?
- Do you try to go "fast"?
- Do not you have time to do a good job?
- Are you tired of work in the same program/module?
- Does your Boss push you to finish soon?

The previous arguments could create a swamp of senseless code.

If you say "I will back to fix it later" you could fall in the LeBlanc's law "Later equals never"

You are a professional and the code is your responsibility. Let's analyze the following anecdote:

What if you were a doctor and had a patient who demanded that you stop all the silly hand-washing in preparation for surgery because it was taking too much time? Clearly the patient is the boss; and yet the doctor should absolutely refuse to comply. Why? Because the doctor knows more than the patient about the risks of disease and infection. It would be unprofessional (never mind criminal) for the doctor to comply with the patient.
So too it is unprofessional for programmers to bend to the will of managers who don't understand the risks of making messes.

Maybe sometime you think in go fast to make the deadline. The only way to go fast is to keep the code as clean as possible at all times.

What is Clean Code?

Each experimented programmer has his/her own definition of clean code, but something is clear, a clean code is a code that you can read easily. The clean code is code that has been taken care of.

In his book Uncle Bob says the next:

Consider this book a description of the Object Mentor School of Clean Code. The techniques and teachings within are the way that we practice our art. We are willing to claim that if you follow these teachings, you will enjoy the benefits that we have enjoyed, and you will learn to write code that is clean and professional. But don't make the mistake of thinking that we are somehow "right" in any absolute sense. There are other schools and other masters that have just as much claim to professionalism as we. It would behoove you to learn from them as well.

The boy Scout Rule

It's not enough to write the code well. The code has to be kept clean over time. We have all seen code rot and degrade as time passes. So we must take an active role in preventing this degradation.

It's a good practice apply the Boy Scout Rule

Always leave the campground cleaner than you found it.