

Clean Code by Robert C. Martin

“Getting software to work and making software clean are two very different activities.”

What is clean code?

Clean code is code that is read like a well-written prose. It exhibits close attention to detail and it is focused and easy to understand.

Why is it so important to write clean code?

Most of what we do as developers is maintaining code that we have or someone else has already written. Making your code easy to understand is essential to making your code easy to maintain. Once you have written code that works, you have to go back and make it clean.

How do we write clean code?

Comments

1. Good code does not need a lot of comments to be understandable.
2. When you find yourself in a position where you need to write a comment, think it through and see whether there isn't some way to express yourself in code.

Functions

1. Keep functions small.
2. Functions should do ONLY one thing.
3. The name of the function should express exactly what the function does.

Classes

1. Keep classes small.
2. Classes should have ONLY one responsibility.
3. The name of a class should describe what responsibilities it fulfills.
4. Classes should be written in a way to make changing the code, adding and changing functionality easy without messing with the rest of the code.

Names

1. Choosing good names takes time but saves more than it takes.
2. Choose names that reveal intent.
3. Avoid the use of lower-case L or uppercase O in names. They get confused with one and zero.
4. Classes should have noun names.
5. Functions should have verb names.

Formatting

1. Choose a set of simple rules that govern the format of your code, and then consistently apply those rules.
2. Use blank lines to separate different aspects of your code.
3. Modules should be short. They should never go beyond a few hundred lines.
4. Within a module, the functions that are most important should come first.
5. Lines of code that are closely related should be closer together.
6. Variables should be declared as close to their usage as possible.
7. If one function calls the other, they should be vertically close. The caller should be above the callee.