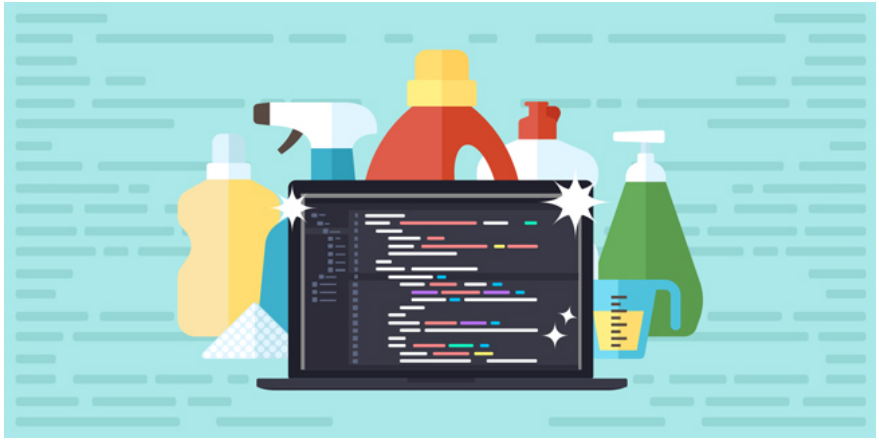
9 tips to help you write 'more delicious' code

Writing code is obviously the first task if you are a software developer. However, have you ever thought your code was really good?

Writing code is obviously the first task if you are a software developer. However, have you ever thought your code was really good?

Keeping the code 'clean', clear and easy to read is a good habit but not all developers are interested. It's a pity, because the clean code will really change and affect many things in your project.

So what's the clean code - Clean Code?



Clean Code understands wall meaning clean code, clear code, easy to read.

As for form, Clean Code is shown as follows:

1. How to present code: how to align, indent using tabs, space . so that it is easy to read, easy to see.
2. How to name variables, functions and classes according to the general rules when programming?
3. How to distribute the amount of code (number of lines of code in the file, number of lines of code in 1 method .)

About the content to consider:

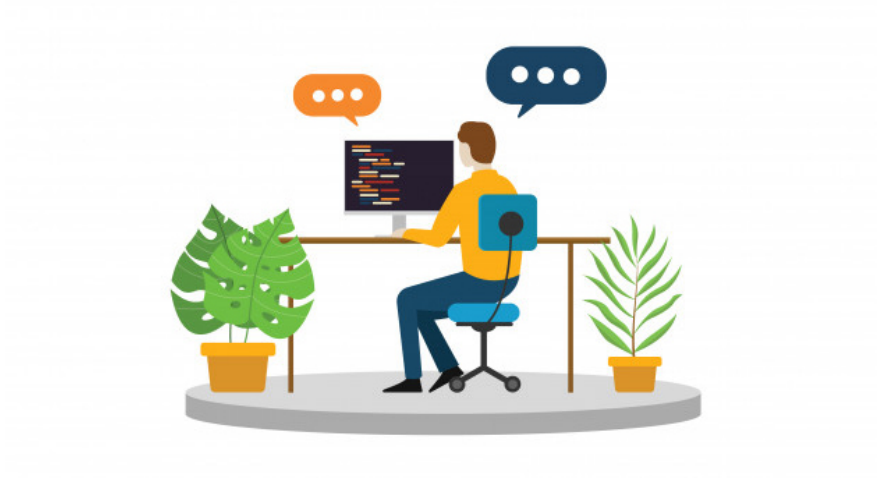
1. How to name functions, variable names must be easy to read and understand.
2. Determine when to write a comment for the code, when not.
3. Design of object structure construction (Object) data is easy to use and easy to expand.
4. Design reasonable Exception handling.

5. Is the ability to maintain and expand the code good?
6. .

It's simple to listen, but it's not easy.

So with this article, Quantrimang will list **9 tips to help you write "clean and better" code** . Invites you to read the track!

Name meaningful



Please ask: what do objects, variables, classes, functions mean? There are many scholarly definitions of these concepts, but when thinking holistically, it can be said that this is the link between the developers and the basic logic of the application you are writing.

If you use ambiguous names, do not describe the variables, classes, and functions, you are manually hiding the application logic and making it difficult for you to update and later. application editing. This also leads co-workers to read your code later on in a labyrinth without seeing the way back.

> *"I'm not a great programmer; I'm just a good programmer with great habits." - Kent Beck*

What does a variable named abc mean? Be always! You might need to reread the entire code to understand what abc means !? In general, reading this type of code will be extremely hard. Instead of wanting to render the customer list with the variable *list1*, use *listCustomer*. Because *list1* is a meaningless, confusing and messy name to read. The same is true for naming classes or functions.

So, in summary, the variable name is required to answer 3 big questions:

1. **WHAT:** What is it?
2. **WHY:** What is it for?
3. **HOW:** How to use it?

A few more seconds to think of a standard name, save hours later.

Functions and classes should be written in a short and unique way

Have you ever encountered a long function hundreds or even thousands of lines yet? If so, you must imagine how painful it would be if we had to browse, read, or edit such functions. Comments will probably help a bit, but only at a very limited level.

So, the function must be as short as possible. An ideal function should not write more than 20 lines. In principle, if the function is too long, break it down into sub-functions, each sub-function does one thing. The name of the function must clearly show its unique effect.

For example, a complex function like *GetCreditScore ()* can be divided into many smaller help functions like *GetCreditReports (), ApplyCreditHistoryAge ()* and *FilterOutstandingMarks ()* .

Delete unnecessary code

Bad habits of retaining some redundant code are one of the things that every programmer has experienced. Surely you will be familiar with this scenario: you want to fix or optimize a code, but after commenting and writing a new code - and even though it works, you still retain the old code. over there.

Over time these comments are no longer needed and will cause your source file to be messy. Therefore, when there is a redundant code, yes or no, we need to remove from the source code to keep your application cleaner.

Avoid shortening the code too much

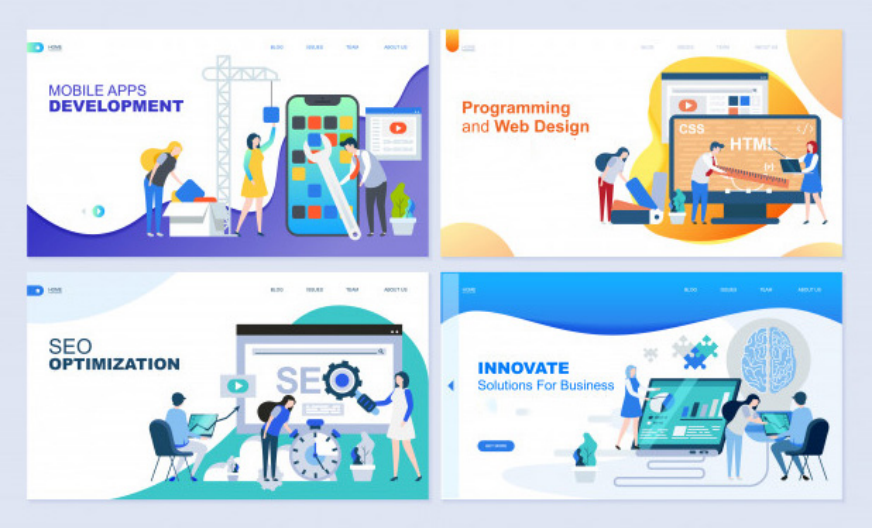
There are many programmers who want to combine *clean code - clean code* with *clever code - smart code* , such as how to put 10 lines into one way, it will look cleaner. This will definitely help the code take up less space on the screen, but is it really easy to understand?

The answer is sometimes possible, but most claim that it's not easy to understand at all!

Often, writing 'smart' code is often to express my personal self, or implicitly show that 'my mind is smart so I can think of it'. There are also cases where writers enjoy the challenge of making a difference, but accidentally make it difficult for readers to understand the code later.

So, to write clean code, please put your ego back, optimize the code so that the next person working with the application can understand. Even after a while, users of this code are still friends, and you do not remember what the shortcuts used in this short code of yours mean, it is possible! It's weird if you say that you can't understand what you used to do?

Select the architecture that is suitable for the project



There are many different models and architectures that you can use to create your project. But keep in mind that it is necessary to choose the right architecture for your needs, not choose the best one.

For example :

1. Model-View-Controller (MVC) model is very popular in web development today because it helps keep your code organized and designed so as to minimize maintenance.
2. Similarly, the Entity-Component-System (ECS) model is very popular in game development because it makes game data modules easier to maintain.

Master the individual nuances of the language used

One of the difficulties in mastering a new programming language is learning nuances that separate it from all other languages. These nuances may be the difference between a dirty code, complex and clean code, easy to maintain.

Consider Python, Java and JavaScript, these 3 languages ??have very clear differences. While Python stands out with short and simple lines of code, Java is explicit in a fairly lengthy way. So every programming language has its own nuances, before you want to clean code, you have to learn and understand them!

Learn clean code from Masters in the industry

If you want to write clean code, it's best to 'index' to see what the clean code looks like and try to understand why clean code is like that. There is no better way to do this than studying the source files of the preceding ones, especially those of the subjects that are considered Master in the industry.

Of course you can't go to Microsoft headquarters and sneak peek at their projects, but you can always consult the famous open source projects. Try with great projects on Github!

One of the reasons that open source projects are so welcome is to help others learn a lot from their code.

Write a reasonable comment

Reasonable, appropriate comment writing is the oldest advice in the programming world. In fact, since the introduction of the comment, the 'newbie' often tends to abuse excessive commenting - describing things that don't need to be described, lack of perspective on a "logical comment" meaning what.

This is a small rule: comments exist to explain why a code exists and not what the code actually does. If the code is written clean enough, it explains itself like what it does - the comment should only shed some light on the intention behind why it was written.

Refactor, Refactor, Refactor

Code refactoring is an editing activity that makes source code easier to read, more organized, and can have better architecture / structure but does not change the behavior of the system in terms of functionality.

Simply understanding, refactor (refactoring, improving the source code) is a miraculous way to clean code. You can see this is the most important tip if you want your code to be 'clean'.

It's over, if you are learning and want to write clean code, you can try it. If you have any tips, welcome to share with Quantrimang.