What is Clean Code?

Clean code is subjective and every developer has a personal take on it. There are some ideas that are considered best practice and what constitutes as clean code within the industry and community, but there is no definitive distinction. And I don’t think there ever will be.

After reading a few books on the topic, giving it some thought and delivering a couple of talks on the subject, if I had to summarize what clean code means in one sentence, I would say that for me:

*Clean code is code that is easy to understand and easy to change.*

Ok, that sounds nice, but what does it really mean? Let’s break that sentence apart and examine the individual points behind it.

Easy to understand means the code is easy to read, whether that reader is the original author of the code or somebody else. It’s meaning is clear so it minimizes the need for guesswork and possibility for misunderstandings. It is easy to understand on every level, specifically:

* It is easy to understand the execution flow of the entire application
* It is easy to understand how the different objects collaborate with each other
* It is easy to understand the role and responsibility of each class
* It is easy to understand what each method does
* It is easy to understand what is the purpose of each expression and variable

Easy to change means the code is easy to extend and refactor, and it’s easy to fix bugs in the codebase. This can be achieved if the person making the changes understands the code and also feels confident that the changes introduced in the code do not break any existing functionality. For the code to be easy to change:

* Classes and methods are small and only have single responsibility
* Classes have clear and concise public APIs
* Classes and methods are predictable and work as expected
* The code is easily testable and has unit tests (or it is easy to write the tests)
* Tests are easy to understand and easy to change

As I stated in the introduction, I plan to write a series of posts that cover these topics in more detail.

Why should you care about Clean Code?

As Robert C. Martin stated in his book [*Clean Code: A Handbook of Agile Software Craftsmanship*](http://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882), “Clean code is code that has been taken care of. Someone has taken the time to keep it simple and orderly. They have paid appropriate attention to details. They have cared.” But why should you care? What’s wrong with code that just works?

You should care because code is (almost) never written just once and then forgotten. Most of the time you, or someone else, need to work on the code. And to be able to work on it efficiently you need to understand the code.

And because people need to understand the code we write, we can say that the code we write is not intended only for the computer but also for humans.

*Programming is the art of telling another human what one wants the computer to do.  
— Donald Knuth*

If you write clean code, then you are helping your future self and your co-workers. You are reducing the cost of maintenance of the application you are writing. You are making it easier to estimate the time needed for new features. You are making it easier to fix bugs. You are making it more enjoyable to work on the code for many years to come. Essentially you are making the life easier for everyone involved in the project.

Now, I’m not saying you should get obsessed about clean code. Your code needs to provide value, so you can’t spend countless hours making it perfect. Clean code usually doesn’t happen on first try anyway, so you need to adopt a mindset that you will always strive to improve the code you are working on. You then need to decide when it is good enough and move on.

Conclusion

In this post I have tried to explain what clean code means to me and also hopefully convinced you that you should also care about clean code (in case you didn’t previously).

My writing productivity has not been that good lately, but hopefully I will get around to writing more posts soon. So check back if you are interested. Also any feedback and comments would be greatly appreciated, either here or Twitter.

To close things up, I would like to share a few more of my favourite quotes and tweets I have come across lately.