# CLEAN CODE

Ciobanu Andrei-Alexandru 2.2

## Characteristics of clean code

1) It should be elegant and pleasing to read.

2) Runs all tests

3) Contains no duplication

4) Minimize the number of entities (classes, methods, functions)

## Why should we write clean code ?

1) **Better use of your time**

The first beneficiary of clean code is the programmer himself. After you work for a long period of time on a project, it's easy to forget things about the code. Clean code helps you understand faster what you did.

2) **Easier debugging**

Whether you write dirty or clean code, bugs are inevitable. But clean code will make the debugging process a lot faster.

3) **Easier for new team members**

Using clean code principles helps to get a new programmer onboard. He can directly jump into it. This reduces the time for training and the time it takes for the new programmer to get started on the project.

4) **You'll feel good**

Writing clean code will make you more confident in sharing your code to others and will make the programming itself more enjoyable.

## How to write clean code?

### 1) **Naming**

Clean code makes use of intention revealing name. Choosing good names takes time but saves more than it takes. The name of a variable, function or class should answer all the questions. It should tell you why it exists, what it does and how it is used. If a name requires a comment, then the name does not reveal its intent.

For example- int d ; // elapsed time in days.

We should choose a name that specifies what is being measured and the unit of that measurement.

A better name would be : int elapsedTime( or even elapsedTimeInDays if it is necessary) ;

Classes and object should have noun or noun phrases names like Customer, Person, Car, Account.

Methods should have verb or verb phrase names like deletePage, save, calculatePay.

A function shouldn’t have more than 3 arguments. Keep the number as low as possible. Also, the body of the function should be kept small. It Is better to have more small function than to have a big and dirty one. It will be even easier to understand with good naming of the functions.

**2) Comments**

If you are writing comments to prove your point, you are doing an error. Ideally, comments are not required at all. If your code needs commenting, you are doing something wrong. The code should explain everything. Comments should be an option only when It's impossible for the code to explain itself through clean code principles.

# Bibliography

https://simpleprogrammer.com/clean-code-principles-better-programmer/

https://medium.com/mindorks/how-to-write-clean-code-lessons-learnt-from-the-clean-code-robert-c-martin-9ffc7aef870c

https://x-team.com/blog/principles-clean-code/

https://medium.com/coding-skills/clean-code-101-meaningful-names-and-functions-bf450456d90c