Clean code

The main purpose of refactoring is to fight technical debt. It transforms a mess into clean code and simple design.

ice! But what's clean code, anyway? Here are some of its features:

Clean code is obvious for other programmers.

And I'm not talking about super sophisticated algorithms. Poor variable naming, bloated classes and methods, magic numbers -you name it- all of that makes code sloppy and difficult to grasp.

Clean code doesn't contain duplication.

Each time you have to make a change in a duplicate code, you have to remember to make the same change to every instance. This increases the cognitive load and slows down the progress.

Clean code contains a minimal number of classes and other moving parts.

Less code is less stuff to keep in your head. Less code is less maintenance. Less code is fewer bugs. Code is liability, keep it short and simple.

Clean code passes all tests.

You know your code is dirty when only 95% of your tests passed. You know you're screwed when you test coverage is 0%.

Clean code is easier and cheaper to maintain!