

Chapter one of clean code to Uncle Bob

1. "Working mess is better than nothing" I think I have fallen into that many times in the past where I did write messy code & just figured if it works then I wouldn't touch it. Of course I knew it was bad practise but laziness gets the best of us.
2. It is programmers' job to defend their code meaning that you will have to make sure that everything is clean to be able to go fast.
3. People may think that messy code is faster but It isn't the case and no proof needed, Most programmers know it is just easier.
4. Code-sense --> Feeling if the code is clean or bad.
5. Bjarne stroustrup on clean code --> I like my code to be elegant and efficient. The logic should be straightforward to make it hard for bugs to hide, the dependencies minimal to ease maintenance, error handling complete according to an articulated strategy, and performance close to optimal so as not to tempt people to make the code messy with unprincipled optimizations. Clean code does one thing well.
6. You can think of bad code (metaphorically) as a house with broken windows (nobody cares about it) so others stop caring also. So the result is more broken windows. all starts with one window.
7. CLEAN CODE IS --FOCUSED-- each function, each class, each module exposes a single-minded attitude. undistracted/unpolluted by surrounding details.
8. Clean code can be read, enhanced by a developer other than its original author.
9. Leave code that gives the impression that you care deeply about the craft that when someone else try to improve it , they are led back to where you are. They will appreciate your code that way.
10. If a method do more than one thing, you should brake it resulting in one method that says more clearly what it does, and some submethods saying how it is done.
11. Leave the campground cleaner than you found it.