Clean Code
Chpt 1

CS342 – Fall 2017

20 Essential Software Development Books to Read – Dzone.com

- https://apple.news/AiqtCuHXEOlu49_pZTV6yag

1. Refactoring: Improving the Design of Existing Code by Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts, Erich Gamma
4. Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch
9. Head First Design Patterns: A Brain-Friendly Guide by Eric Freeman, Bert Bates, Kathy Sierra, Elisabeth Robson
14. Agile Testing: A Practical Guide for Testers and Agile Teams by Lisa Crispin, Janet Gregory
16. Agile Software Development: Principles, Patterns, and Practices by Robert C. Martin

Clean Code

- Bad code can bring a company down.
- We wade through bad code, code that impedes our understanding.
- Writing good code takes energy and thoughtful deliberation.
- LeBlanc’s law: Later equals never
- Complexity Slows Done Development
- Lehman’s Laws of Software Evolution
    - "Continuing Change" — an E-type system must be continually adapted or it becomes progressively less satisfactory
    - "Increasing Complexity" — as an E-type system evolves, its complexity increases unless work is done to maintain or reduce it

What is Clean Code?

- Bjarne Stroustrup, Inventor of C++

- 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.

What is Clean Code?

- Grady Booch, author of Object Oriented Analysis and Design with Applications

- Clean code is simple and direct. Clean code reads like well-written prose. Clean code never obscures the designer’s intent but rather is full of crisp abstractions and straightforward lines of control.

What is Clean Code?

- “Big” Dave Thomas, godfather of Eclipse

- Clean code can be read, and enhanced by a developer other than its original author. It has unit and acceptance tests. It has meaningful names. It provides one way rather than many ways for doing one thing. It has minimal dependencies, which are explicitly defined, and provides a clear and minimal API. Code should be literate since depending on the language, not all necessary information can be expressed clearly in code alone.

What is Clean Code?

- Ron Jeffries, author of Extreme Programming Installed

- Consider Beck’s rules of simple code. In priority order, simple code:
    1. Runs all the tests;
    2. Contains no duplication;
    3. Expresses all the design ideas that are in the system;
    4. Minimizes the number of entities such as classes, methods, functions, and the like.

What is Clean Code?

- Robert Martin, author of Clean Code

- The ratio of time spent reading vs. writing is well over 10:1. We are constantly reading old code as part of the effort to write new code. Because this ratio is so high, we want the reading of code to be easy, even if it makes the writing harder. Of course there’s no way to write code without reading it, so making it easy to read actually makes it easier to write

What is Clean Code?

- The Boy Scout Rule:
  - Leave the campground cleaner than you found it.

- Applying that idea to coding.
- Imagine if over time code always got better!
- Cleaning can be done a little at a time.

- Clean Coding is at art. An artist must continually practice.
  - How do you get to Carnegie Hall?