Kevin Walsh

Cs 361

Exploration 1.2: Clean Code Intro & Meaningful Names

10/06/2023

1. WhoTF is Robert Martin (investigate online) and why might we consider listening to what he has to say?

Robert martin is widely respected in the software industry. He had also been a editor of the C++ report and currently writes for the software development magazine.

1. In the Introduction, Martin describes two components to craftsmanship. What are they? Do you disagree? He also describes the structure of the book. What is the structure of the book, and what can we expect?

The two parts are knowledge and work I agree with both of these. The book has a structure where it teaches you material then tests you on the material that it taught you. It is not a “feel good book” and to follow code. There are three parts teach princibles, practice, then pay off.

1. There is currently some trendy buzz about how programmers will be out of jobs because AI will create programs; there is also increasing fanfare around "no code" systems. Dig into these topics quickly online. Notice what Uncle Bob's stance is on these topics. Describe whether you agree or disagree with whether programmers will become obsolete, and why.

I agree with uncle bob I think that as long as code is the only way that is as well specified to deal with the requirements of the problem you are dealing with then people who speak that language are going to be required.

1. Identify and summarize (briefly) five main points from Chapter 1. Identify the one that you disagree with or agree with the least and explain why.

What is bad code- what makes code now and, in the future, problems that may arise with wading.

The total of owning your mess -when a code is bad then you can no longer build on that mess and collapses.

The grand redesign- sometimes it is better to completely redesign is necessary but it takes time and resources, and you still have to implement the changes that are continuously added to the program you are redesigning.

The primal conundrum-

Your manager wants you to push out code as fast as possible but that might make you need to push out bad code which is not good so there is give and take.

What is clean code- there are many points but elegant, readability, tests, care, no duplications stood out to me.

1. Chapter 2 gives you 16 (!) tips about names. Identify one of these suggestions that you have violated in your previous code, and describe what you did, why you did it, why it was a poor choice, and what you might do in the future when you're in a similar situation.

Single word/ letter names.

I would use the first letter of the ting I was using for variable names. One that sticks out to me was for colors I would you r for red b for blue and so on. But then I got to the point where I needed to implement black later on and ran into a ton of trouble. Also naming arrays as arr1 arr2 and keeping track of what was stored in each array. I mostly did this because I was lazy and did not want to spent the extra 10 seconds thinking of a good name for what I was working on. That is why when I write code now I try to be as discptive as possible when ever naming variables, arrays, classes, functions and even placeholders as possible so when I am looking back at the code I do not have to comment every time I use them.