

Ben Gaudreau

Professor Levine

COS 420

April 9, 2025

SINGLETONS

After creating a Referee class, the process of having an AI tool refactor it to fit the Singleton design pattern was trivially simple. In addition to that, the AI-generated code followed the template shown in lecture. Given the popularity of the Gang of Four book, it's no surprise to me that their work would find its way into a LLM's corpus of training data as people discuss, present, and share such information online.

As an addendum, I'd like to mention that my current Referee class, although created, has not yet been implemented into my greater Bulldog program. Several events in my personal and academic life have restricted the amount of effort I can put towards refactoring and updating the existing code. Such a need for refactorization was noted in my previous report, and it is due to time restrictions that I have not yet been able to make those changes. So while the greater program as a whole has not functionally changed from the previous iteration to this one, all the work necessary for the completion of this assignment (insofar as studying the efficacy of AI tools in assisting programming is concerned) has been finished.

UNIT TESTING

Modifying the program to facilitate testing of randomness was another rather simple task, but it did require diving a bit into Java's InputStreams. I was able to extract a superclass of the Dice class from the previous iteration using Eclipse's built-in refactoring functions, which I then extended to a new FakeRandom class. From this point on I had gotten ahead of myself and completed the remainder

of the class by hand. But I later asked ChatGPT-4 to solve the problem to see what other methods I could have employed. Instead of reading from a stream, the AI tool decided to parse directly from a String into an array which could be later read in much the same way. I was more surprised by its approach to handling values outside the bounds of its Dice.

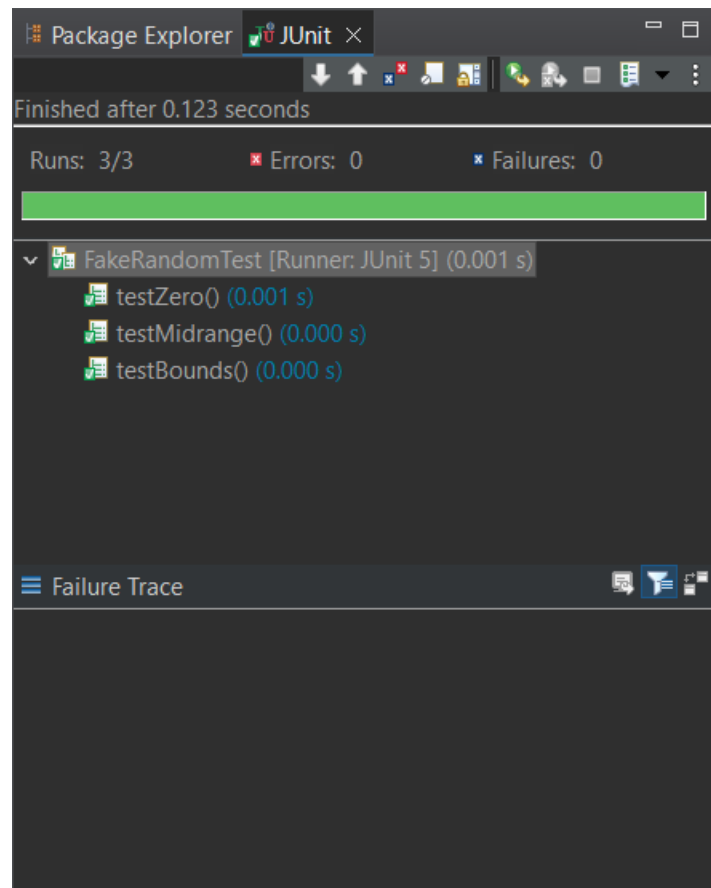
Instead of checking each value on the fly, the AI tool validated each and every value after they were read into the array. Anything outside the bounds would throw an error. In my program, I handled this issue by fitting the erroneous value into the bounds via modulus, and printing a warning containing both the input value and the value being returned by roll(). In the context of this project, I think either implementation is acceptable.

The tests themselves were written by hand, given that only a few tests were required for ensuring the new SevenPlayer class functioned properly. I found it rather difficult to come up with good prompts to generate a test suite similar to what we had come up with in lecture, and some of the tests it did generate were of rather poor quality. In one example, ChatGPT-4 wrote a test to ensure that SevenPlayer could stop on its first roll. However, when playing with a six-sided die, it is mathematically impossible to reach the threshold of 7 points with just one roll. Then again, I had never expected a generative AI to be able to read in a program and learn enough about it to design a full and proper test suite for it.

I believe that the tech industry is making a massive mistake by leaning into such tools in their present state to the extent that they are. I personally don't think that I could ever trust an AI tool to ensure the complete security and stability of a program designed for human use. Even though human supervision and review of AI-generated test suites are likely already being done, the mere existence of AI tools writing unit tests guarantees that some will cut corners, blindly trusting such tools to carry out one of the most vital processes in software development. The CrowdStrike outages from last year

illustrate just one of the many disasters that can arise when software is not properly tested before being released to the public.

Tangent aside, here are the JUnit test results from my (hand-written) test suite:



CONCLUSION

I don't have a whole lot more to say in regards to this iteration. As stated in the original assignment, much of the work was more similar to a self-guided lab than a full-scale programming project. However, each of these smaller explorations have helped me tremendously to better understand and form more robust opinions on the use of artificial intelligence in the myriad industries surrounding computer science. This field evolves at such a rapid pace, and with all the hype surrounding the latest

and largest trends in tech, it can be difficult to slow down and really scrutinize just what exactly we're implementing into and forcing upon the massive and ever-connected chunk of humanity that is now dependent on a technology that did not even exist one hundred years ago.