

This final exam was quite fun. Although I am a little nervous about my implementations, I am overall satisfied about how it turned out. I made many decisions and over time, wondered if it was a good idea or not. An example was the use of enumerations, because while the idea did sound good (at least in my head), I found that it actually wasn't that useful in the end. If I had started the project with the idea of enumerations in mind, maybe that would have been a different story.

Another decision I had trouble dealing with, was the actual formatting of all the inputs. I wasn't sure if my program should reinforce certain formats (for example, YYYY-DD-MM). It would have taken too much time to enforce how the user should enter inputs. I needed an exact format for years to calculate the salaries of employees, and I'm not sure whether it was a good idea to just let the user do whatever they wanted. This was a major decision in this project, and I actually ended up not enforcing a format (besides the year, technically). I imagined that since this was at a company, and I was building a software for an organization, they would (hopefully) train people to use my software, and if they didn't, it would simply be the user's fault for not following the instructions. Of course, that leaves the software open for vulnerabilities and malicious employees but, if it works, it works.

Because of that though, the program is actually very fragile. There are some times when I tried to protect the program from bad usage, but for the most part, in some situations, it could end up breaking very quickly with one typo.

Speaking of challenges, I didn't like how unorganized my code was. I was really in a crush for time, so I couldn't optimise it the way I wanted to. For instance, the switch statement made me create a new temporary variable after every case, despite each case being completely unrelated to each other. I knew this before the exam, but it was still annoying. I ended up making a ton of temporary variables, and I regret that I left such a large mess behind. I don't like the way that "if" statements look in long lists though. I'm glad I didn't use if statements for the print menu.

Another major decision I had was whether or not I should send an entire food item or ask the user for quantities to send. I decided it would make more sense to send in quantities, because if somebody donated 1000 apples, it might be absurd to send those 1000 apples to a donee.

I gave it my best effort. It has been a while since I last programmed, but I enjoyed it. I'm really proud of the use of abstract classes though. I didn't need to rewrite the same variables, getters, and setters in the child classes. I was thinking about using a "Human" interface since there were common methods and variables (name and SSN for instance) but I decided not to.

Another thing I could have done was declare SSN as a constant, which, in hindsight, I really should have. But I preferred using zero parameter constructors for some reason.

I also liked organizing some classes into different package source folders. It just makes the directory cleaner and tidy for me.