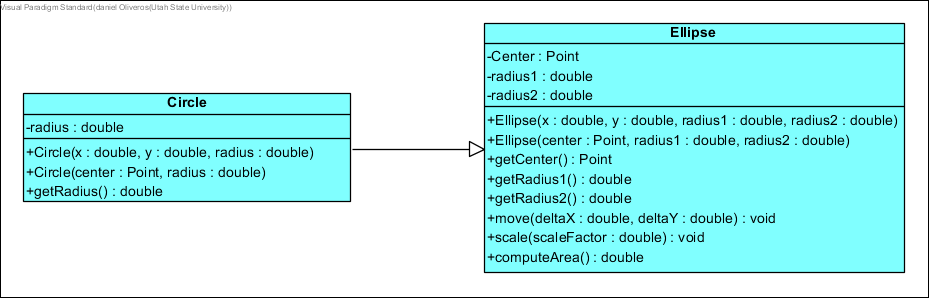
Daniel Oliveros

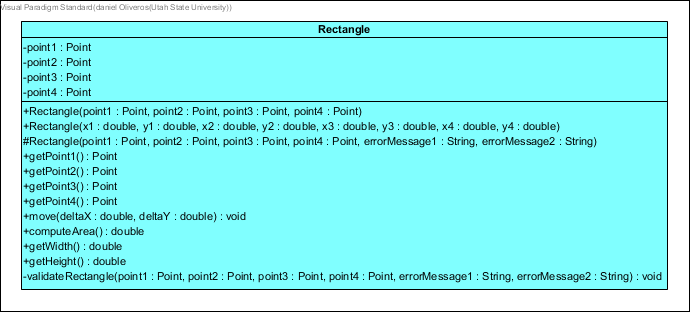
CS 5700

Assignment 1 – Abstraction and Modularity

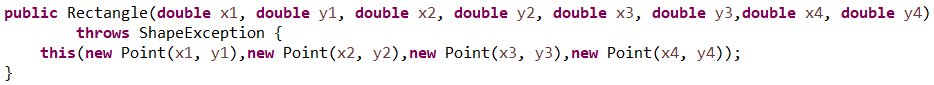
I originally set up most of my validation for shapes working correctly within the Validator class, after meeting to discuss my design earlier today, I worked on redesigning that part of my project to be more modular, which also helped my design in general. Through running tests across my classes, I have also been able to achieve as high a coverage I can really go for. Some of the classes aren’t covered fully, but this is more a quirk of the way in which I’m calculating functions like getWidth in the rectangle class, where I surround my code with a try-catch block so I don’t have to state the “exceptions” this function can throw, in reality, this function can never throw these exceptions as the validators for rectangle check for things working correctly in the first place.

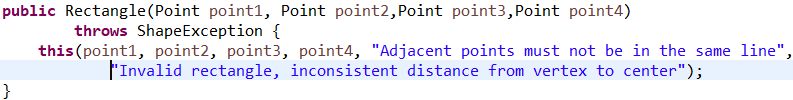


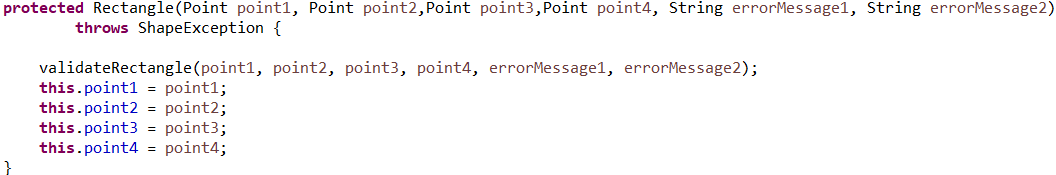
Starting out with the Circle class, seeing as it’s a special case of the Ellipse class, I made it inherit from Ellipse, allowing me to use all the methods from that class, and generally just giving me an easier time with maintaining and testing the class. I did a similar thing with square, however, I also had to add an extra validator to Square as it needed to also check width and height were equal.



Giving the Rectangle class a second look, I also worked on implementing some other general programming practices into it. I overloaded the constructor multiple times for different purposes, and ensured that they were using each other, having most of the code within a single constructor to make it much more maintainable.







I made my main constructor for the class protected, this is because it deals with error messages, and I only wanted these to be specified by either the original class or the classes that inherit from it.

When testing, an interesting insight that came for me was that I could never get full coverage on my program. This was due to a workaround I had to do to get around some issues with my getWidth function. My implementation for it used Lines for some calculations, and because of this they could throw exceptions. So I wrapped it in a try-catch block as I knew it was impossible for them to reach that situation. My unit tests don’t have full coverage of my Rectangle class due to this situation, but I’m fine with that since these pieces of code are never expected to be reached in the first place anyway.

