CPSC501 Refactoring Report

October 9, 2015

From just a quick glance at the code, we can instantly pick up on several “bad smells”. The entire program is written inside of main and it is much too long to keep track of. It is also very hard to test. We can also see a couple of vaguely declared variables, and undesirable switch statements.

The first thing I did was rename the vaguely declared variables with something easier to understand. I used *Rename Method*.

The second thing I did was to pull out the very large for loops and create functions. This will make the main code easier to read. I used *Extract Method*.

The third thing I was to ‘switch’ from switch statements to if/else statements. With switch statements, it is difficult to catch errors with input. Although it’s not technically a refactoring method, switch statements are rare in good OO code.

Uhoh! Another quite obvious ‘bad smell’ has appeared. We have created 4 new methods with vague undescriptive names! I used *Rename Method* again.

The class is still looking stuffy, so I applied *Extract Class* and moved the drawing functions to its own class Draw.java. I also moved the printing of banners to its own class Print.java.

Another bad smell that has appeared is the use of public variables. I set them all to private, then created setters and getters for Draw.java and Print.java using the refactor method *Self Encapsulate Field*.

Now that we have some functions that actually modify values, we can do some Junit testing. It is difficult to test for the output of the entire program, as it is a loop that prints lines. However, we can test the setter and getter methods to make sure they are working as intended.

To increase functionality of draw, and knowing that we could add more classes in the future, I split up the different shapes into their own methods using the refactor method *Extract Subclass* where they extend draw.