CIS573 HW1 Analysis

Shuangchen Shen

September 15, 2017

Q1

I have four classes in total(Main, Document, Decrypt, Encrypt): Document Class is used to build the frequency model and process all the files; Encrypt Class is used to encrypt the files using Caesar cipher; Decrypt Class is used to decrypt the files; Main Class is used to control the entire program. The four specific and distinct ways in which the design of my classes is good are:

1. If I want to change the format of the input file, I only need to modify the methods inside the Document Class instead of modifying the entire program.

2. If I want to change the way of encrypting, I only need to modify the methods inside the Encrypt Class without affecting the rest parts of the program.

3. If I want to change the way of decrypting, I only need to modify the methods inside the Decrypt Class without changing the other parts of the program.

4. If I want to modify the logic of the program, I only need to modify the methods inside the Main Class. Also, I can repeatedly call the methods in other classes as long as they are public.

In conclusion, every class in my program represents a clear, single concept and thus the dependency between classes is largely decreased, which means changing one class won't require much changes to other classes.

Q2

I found my codes easy to read and understand(not because I am the one who wrote it) because of four specific ways:

1. The names for my variables, methods and Classes are meaningful and thus the functions for those variables, methods and classes can be revealed by their names. In short, by just glancing through the names for methods and classes, people can have a rough idea of what my program is about.

2. I used method overloading to avoid unnecessary naming. Within the same class, I have several methods sharing the same name but with different parameter declarations. In this way, I can also benefit from calling a similar method for different types of data.

3. For every method, its length doesn't exceed 30 lines and each line is not very long and thus easy for people to read and understand.

4. I used uniform indentation throughout the program and it's easier for people to read and understand. Also, the structure of my program can be nicely shown by indentation.

Q3

My code is efficient in terms of execution time and/or memory usage because of three specific ways:

1. I used an array to store the frequency of each character. By using this array, it's not only fast to locate each character but also efficient in memory usage.

2. I used HashMap in decrypt to pair the original character and the encrypted character so that it's fast to transform between this two values.

3. I used StringBuffer instead of String because StringBuffer is faster than String in performing concatenations.

Q4

I used JUNIT test to test my program. In my program, Decrypt Class, Document Class and also Encrypt Class are tested. So basically, each class(except the main class) has its test class. The inputs to use for testing are randomly chosen because in real life you might encounter different kinds of inputs and thus it's beneficial to test using arbitrary inputs. My tests covered most of the methods in my program so I think I could stop testing. On a scale of 0-100%, I am 80% sure that my code is correct because my code has about 80% accuracy in average.