Homework 4

Code Tree Design Document

Student:

Lillith Chute

Teacher:

Clark Freifeld

Course:

CS 5010

# Design Summary

1. There are a few operation types that is required for this application:
   1. Encoding: This doesn’t require the use of a code tree, so this process relies on using *Map*.
   2. Decoding: The requirement for this is to use a code tree.
   3. Huffman algorithm: This requirement is simply to implement the algorithm, but it is not necessary for the application itself to use this based on my understanding of the requirements. To prove this out, I used the Declaration of Independence in a test case to validate that it does, in fact, produce a prefix code.
2. Because the fundamental principle here is to use a tree, I created an interface to use as the basis for operations of the code tree.
3. The cipher program is basically an encoding and decoding program, so I also created an interface to handle all of the cipher program operations.
4. Based on the code tree interface, I created a basic tree that handles the decoding operations.
5. I created a driver program that runs the application and allows users to decode and encode text either from the console and/or from files. Further, the results will be displayed either on the console or written to a file per the user’s preference.

# Test Plan

Do the basics:

1. Make sure the tests are not in the same package as the code.
2. Test that the constructors for the various classes work to included error checking and validations.
3. Testing getter and setter methods are generally straightforward.
4. Check code coverage to make sure as many lines of code have been checked as possible.
5. As a note, I test private methods by making them public if they are complex and then setting them back to private. There is probably a better way, but that’s the quick and dirty way for me at the moment.

## Difficulties

N/A

## Notes

1. I tested the functionality that would be available to the user with a few additional internal methods tested for assurance.
2. The Huffman algorithm is validated through testing in the test suite.

## UML Diagram

