Abiola Gabriel Olofin

CS 150-02

Professor Falconi

01/27/2020

Lab 1 Report

Introduction

In this lab, we were asked to solve two problems. The second problem was to write a program that reads a file from the current working directory and writes it to another file in the current working directory – in order words, it is taking information from one file, dissecting the information, and that data will be printed out into another file to read. The goal was to have a program that allows us to read and write into different text files. Before starting the assignment, I assumed that the project would be creating a program that gave information and printed out to the BlueJ terminal, but it actually needed to be printed out into a different text file.

Approach

When I was designing the program, I only created one class that ran the entire program called ReadWrite. I used the try and catch method in order to make sure that whenever there was an error in the text file input, it would throw an exception and notify me that there was an error. I also imported the scanner, PrintWriter, and FileReader API. In the ReadWrite class, I created a method called “run()” which runs the try and catch that I would use. The method “run()” is a void method so it won’t return anything but at the end it does create a new text file in the folder I am currently in. First thing, I did was to create 2 scanners that would allow to me to make one scanner scan a line and the other scan the words in each line. Second thing I did was create a new Scanner object that would allow the program to take the scan the text file and read it as well and the FileReader object was masked as a Scanner object. I also used while loops to loop through both the text file as a whole, each line in the text file and each word in each line.

Methods

While writing the code, I ran each part of the method in order to make sure I was on the track and that it compiled correctly. This included using print statements to map where the computer had gone when run the program. This allowed me to keep track of the computer’s movements and whether there were any errors in the code. After knowing the code was complete, I tested it multiple times with different text files that I created and see whether it would give me a text file with the accurate information about the text file it was reading from. I also gave it a fake text file to see whether it would print out the exception which it did.

Data

A picture containing table

Description automatically generated

Conclusion

Overall, this program was ßsuccessful, and it taught me how to read a text file and write the

information I wanted from that text file to a new text file.

References

Scanner API

PrintWriter API

FileReader API

Try and Catch: <http://docs.oracle.com/javase/tutorial/essential/exceptions/handling.html>