2021

JAVA WORD COUNTER

YAZEED ALZUGHAIBI

Table of Contents

Introduction:

The best way to learn something and deeply understand it is implementation. In this homework the requirements were to write a Java program that takes a text file and counts the number of words in it and the number of occurrences for each word to be written in an output text file. This homework is required in order to get the best understanding of mapping in java or programming in general. To achieve the goal of this assignment, the concept of mapping or HashMaps will be the main focus for counting and sorting. Throughout this brief report, the main topics to be discussed are; some background information about mapping and how it works, a quick dive into the simple algorithm and finally discuss the finding & results.

Background:

Before mentioning anything about the Java written program, some concepts need to be clarified. One important concept about this program are HashMaps; HashMap is a built-in Java class that implements Map interface Figure(1). It is an object that maps keys to their specific values figure(2), maps cannot have duplicate keys thus having two different keys pointing to the same value. It can be treated same as a dictionary. The Map interface holds three collection views, which permits a map's contents or elements to be demonstrated as a set of keys, a collection of values or a set of keys to value mappings. However, the order of a map is defined by the order the items are put in but some map implementations such as TreeMap class can sort the items with respect to the keys of a map automatically, others like the HashMap do not sort the items.[1]

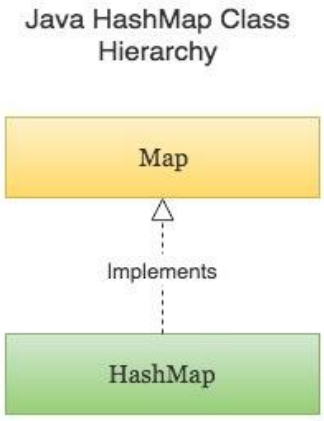

Figure 1: Map tree figure

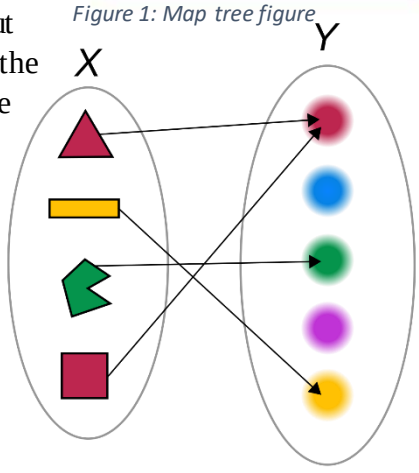

Figure 2: mapping example

The WordCounter:

```
import java.io.FileReader; //Import th
import java.io.FileNotFoundException;
import java.io.BufferedReader; //Impor
import java.io.FileWriter; //Import th
import java.nio.file.Path; //Import th
import java.nio.file.Paths; //Import t
import java.util.*; //Import this clas
import java.util.Map.Entry; //Import t
import java.lang.*; //Import this clas
```

Figure 3: imported classes

First thing to be seen in the class are the imported classes. The main ones to mention are; "FileReader" which reads the input file, "FileNotFoundException" which is for throwing an exception if the input file name was not found, "BufferedReader" which allows the program to read the input file line by line from the FileReader, "FileWriter" which allows us to create an output text file and write on it, "Path & Paths" to permit the program to access text files within its directory, "Java.util" is to be able to use arrays and array lists for sorting purposes, "Map.Entry" to ease the converting of the map to an arraylist and finally "Java.lang" to use class wrappers such as "Integer" objects. Figure (3)

Main Method:

To put it simply, the class starts by creating needed objects such as Path, FileWriter and BufferedReader. Create a HashMap for the words and their number of occurrences, receive and read through the input text file, discard of unneeded empty lines and punctuation marks. Separate the words using the ".split" function and store them in the Map using ".put" and increment their counters. After reading through every line and counting the occurrence of every word, count the number of words by simply counting all the values of the keys in the map. Then starts the sorting of the words, using the Collections Class from "Java.util" to sort them in descending or ascending order (the user can choose while its descending on default if the user doesn't choose). After storing the sorted map in a list, another map is created to store the sorted words in and lastly print that map in an output text file. The whole main method has one big try and catch with the catch pointed at "FileNotFoundException" just in case the input file does not exist.

To run this program, the user should their command prompt window, go the directory of the program; also make sure the input file is in the same directory as the program itself. Start by writing; <Javac WordCounter.java> after that write; <java WordCounter (input file name).txt (output file name).txt (the order needed from the user; "asc" for ascending and "des" for descending)>. After that there shou;d be an output file in the same directory for the user to look at. The command prompt window should look like figure(4).

```
C:\Users\      \workspace\EE367\src>javac WordCounter.java

C:\Users\      \workspace\EE367\src>java WordCounter big.txt output.txt asc
Output file has been generated with the name: output.txt
```

Figure 4: Command prompt window

Results and Findings:

In this section od the report, the results are shown, and the findings are discussed. The begin with, when the input file was "big.txt" the results were that; the number of words counted were: 1115031 words, the most mentioned word was "the" and it was mentioned (80030) times. I personally faced issues with the default arguments for running the program did not work as expected, I was only able to make it run by writing "asc" or "des" otherwise it won't run. Figure(5)

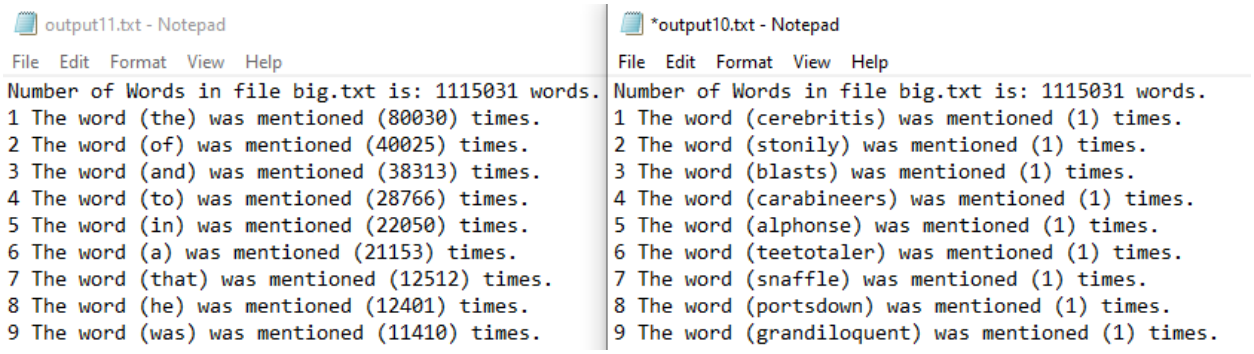

Figure 5: output text files

Discussion:

At the end, this was a great experience and a refreshing return to java programing and how great it is. In this homework I was able to take an input file, read through it and process it, count the number of words in it and the occurrences of every word in it. Like I mentioned before the issues I faced were about the default arguments for running the program and the user had to choose and write ascending or descending, otherwise everything went great and it all works well as expected. After submitting this work I will get back to the program and work on it until I achieve the required goals from this homework.

References:

1- Mapping in java
https://docs.oracle.com/javase/8/docs/api/java/util/Map.html