REPORT

App :

This code is a Java program that counts the frequency of words in a given text file. It prompts the user to enter the name of the file they want to read, then uses a HashMap to store the words and their corresponding frequency. The code then uses a Scanner to read the file name, and a File object to create a File object representing the file. It then uses the RootFinder and FindFile classes to search for the file in various directories, including the system root, the user's home directory, and the current project directory. Once the file is found, the code uses a FileInputStream, DataInputStream, and BufferedReader to read the file line by line, and split each line into an array of words. The code then uses a for loop to iterate through the words, and adds them to the HashMap, incrementing their frequency if they already exist in the HashMap.
The program might be slow depending on the position of the .txt file the user enters. In order to make it a little faster, there are added shortcuts to the home directory and to the directory of the project /target/ to 1st search there and from the root of the system. Also, an added option to make it a little faster is to enter the absolute path of the file.

RootFinder :

This class is used to find the root directory of the system. The `findRoot()` method uses the `File.listRoots()` method to get all the root directories of the file system. It then concatenates all the root directories into a single string and splits it to get the first root directory. The first root directory is then returned as a `File` object.

FindFile :

This code is a Java class that defines a method that finds and returns the complete path of a file. The class has a method named "findFile" which takes two arguments: a string representing the name of the file to be found and a File object representing the directory to search in. The method recursively searches through the directory and its subdirectories looking for a file with the given name. If the file is found, the method returns the complete path of the file, otherwise, it returns the last directory searched.

The method first creates an array of File objects, "list", which stores the subdirectories and files of the directory passed as the second argument. It then checks if the list is empty or not. If it's not, the method runs a while loop to iterate

through the contents of the directory. Inside the loop, it checks if the current element is a directory or a file. If it's a directory, the method calls itself again, passing the directory as the second argument. If it's a file, the method checks if the file's name is the same as the name passed as the first argument. If it is, the method returns the complete path of the file and sets a flag variable i to 0, otherwise, it continues searching. Finally, it returns the complete path of the file.

HashMap :

This code is a Java class that defines a HashMap data structure that implements the Dictionary interface. The HashMap class uses an array to store key-value pairs, and a hash function to map keys to indices in the array. The class also includes methods to add, remove and get key-value pairs, as well as a method to check if a key is present in the HashMap.

The class has the following methods:

- put(K key, V value): This method adds a key-value pair to the HashMap. It uses the hash function to get the index in the array where the key-value pair should be stored. If the key already exists in the HashMap, the value is updated. If the HashMap is full, it throws an IllegalStateException.
- remove(K key): This method removes a key-value pair from the HashMap. It uses the hash function to get the index of the key in the array. If the key is not found in the HashMap, it returns null.
- get(K key): This method retrieves the value associated with a key from the HashMap. It uses the hash function to get the index of the key in the array. If the key is not found in the HashMap, it returns null.
- contains(K key): This method checks if a key is present in the HashMap. It calls the get method and returns true if the key is found, and false if the key is not found.
- checkResize(): This method checks if the HashMap needs to be resized (if the number of elements in the HashMap is greater than 75% of the size of the array) and resizes the HashMap if necessary.

It also has a private class Entry<K, V> that is used to store key-value pairs in the HashMap and an inner class HashFunction<K> that is used to hash the keys.

It also has some methods that are related to resizing the table, like expandTable and contractTable, to make the table bigger or smaller depending on the current number of elements stored in the table.

HashFunction :
The class `HashFunction` is a hash function for a generic type K. It uses the randomized technique of dot product hashing.

When an object of the class is created, it initializes an array of random bits with the size of b*u, where b is the size of the hash table and u is the size of the bit array of the key. It also creates an array of bits of the key.

The `hash` method takes a key as input, converts it into an array of bits using the `transformKeyToBits` method, and then performs the dot product between the array of random bits and the array of bits of the key. Finally, it calculates the new hash by taking the dot product mod 2 and converting the result to a decimal number.

The `updateTableSize` method updates the size of the table and creates a new array of random bits with the new size.

The `isPowerOfTwo` method is a helper method which checks if the table size is power of 2 or not.

Bibliography :

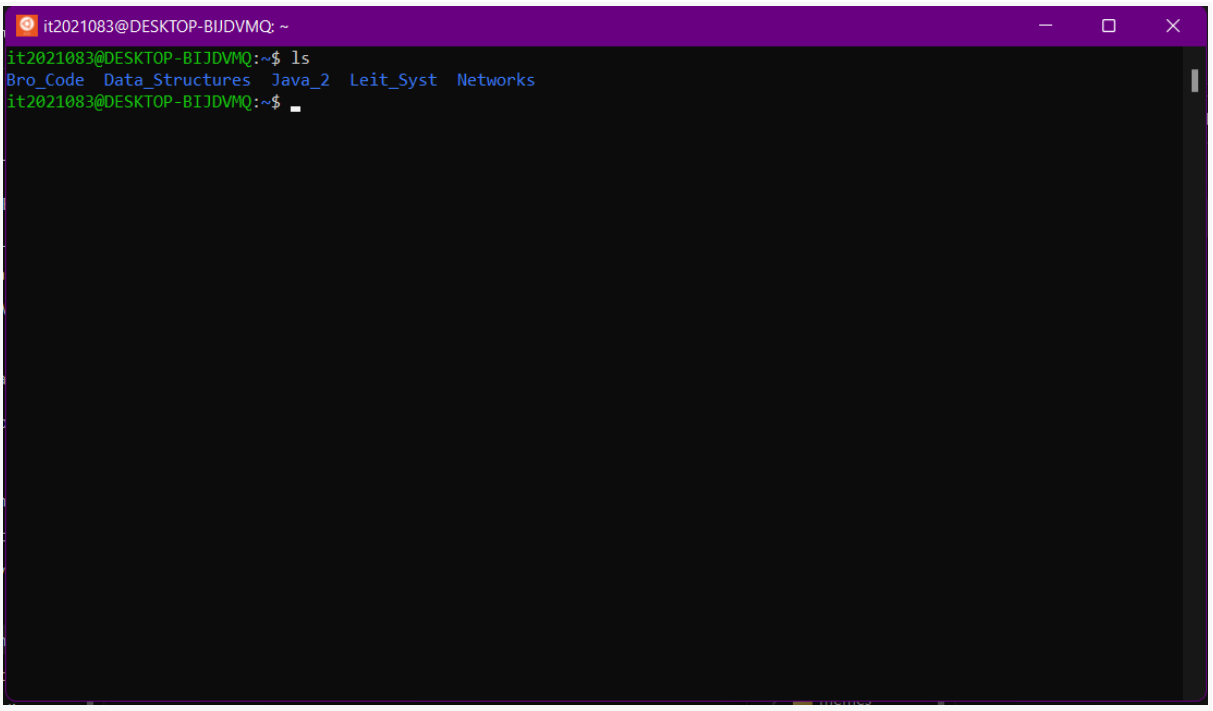
linux -How to make file search program in java -Stack Overflow

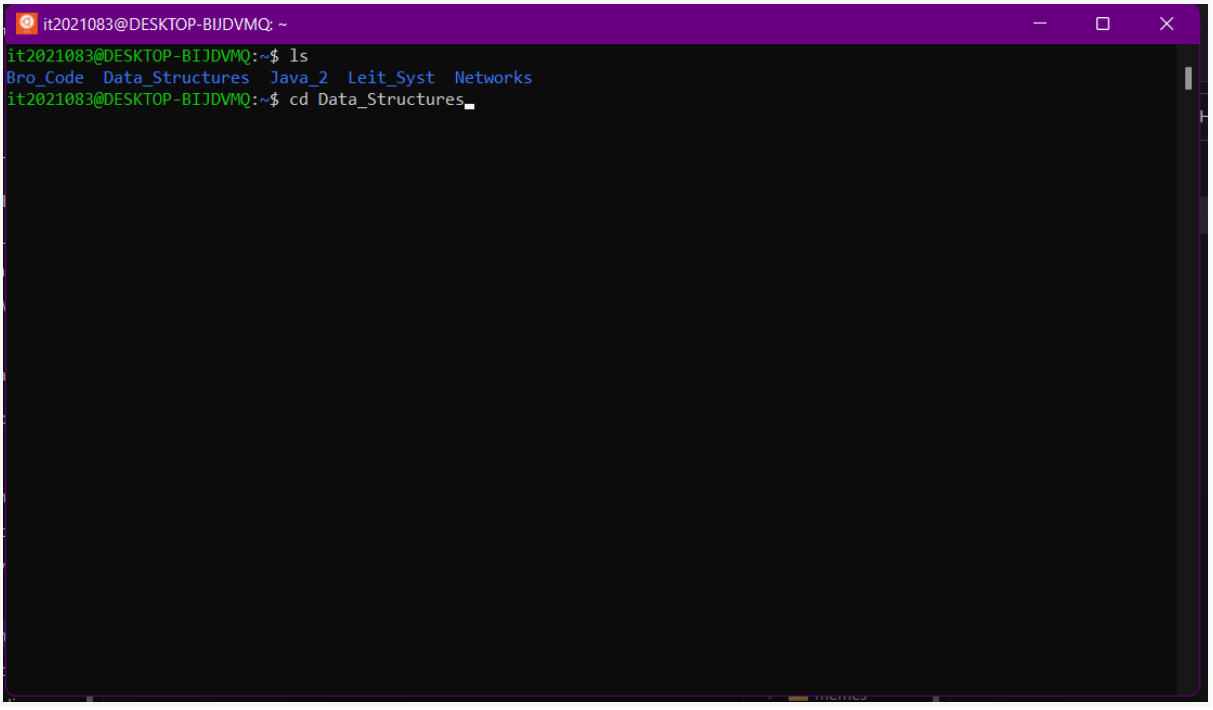
Java Read File Line by Line-JavaTutorial

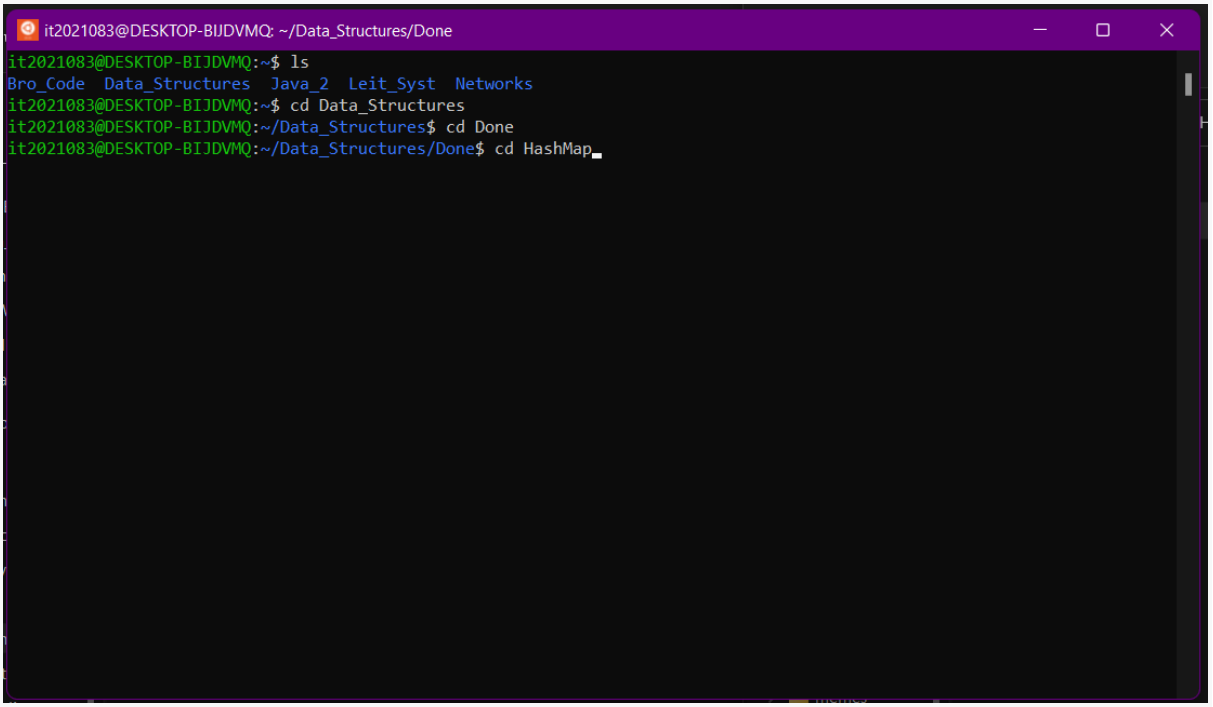
https://stackoverflow.com/questions/2397188/how-to-create-own-hashmap-in-java

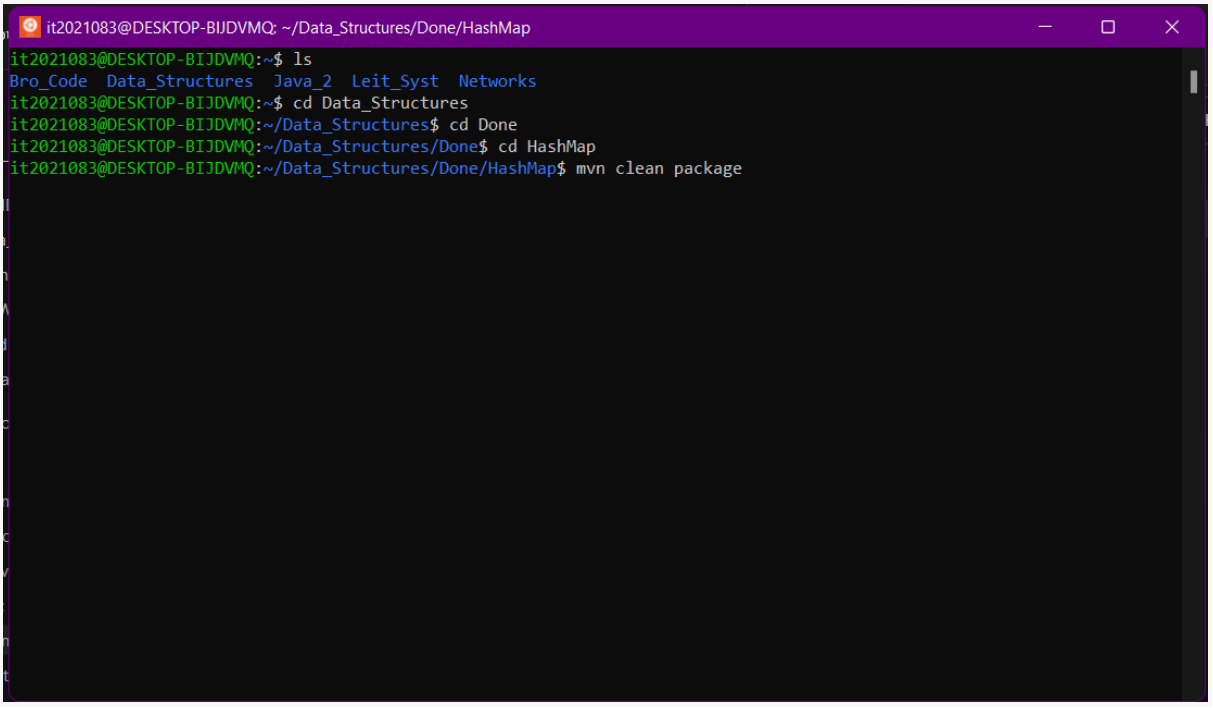
https://www.quora.com/How-do-I-create-my-own-hashing-function-for-use-in-a-look-up-table

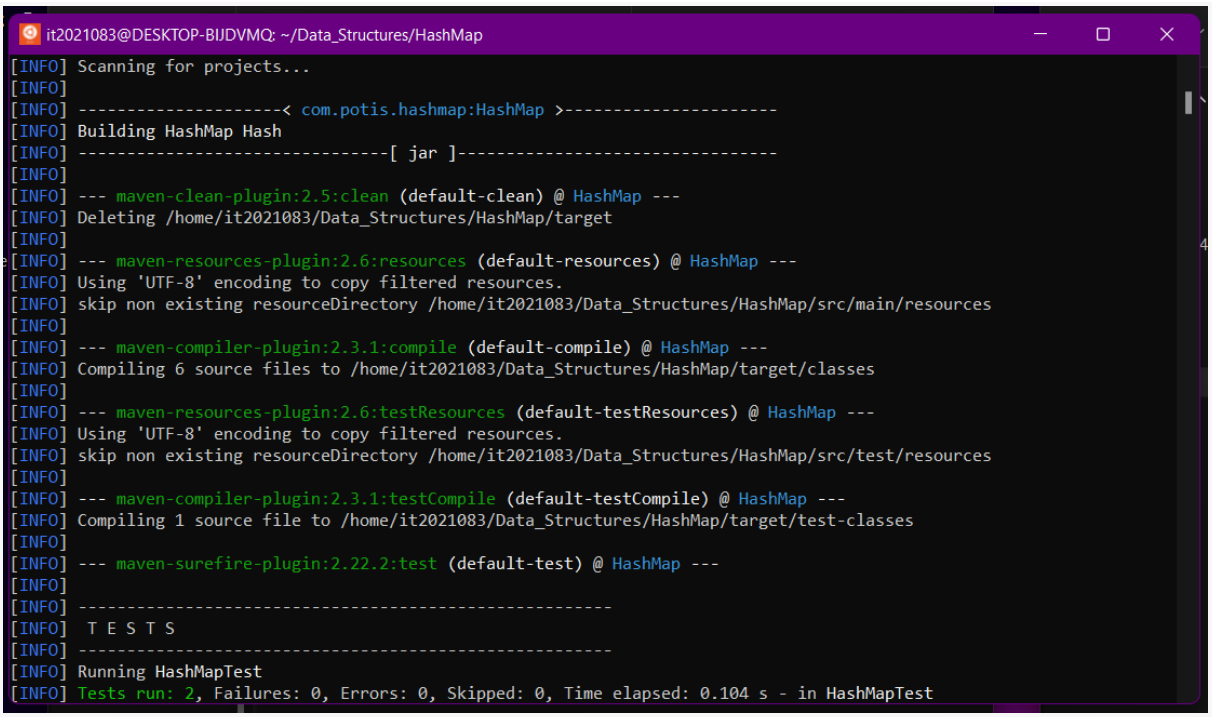
Example of use :

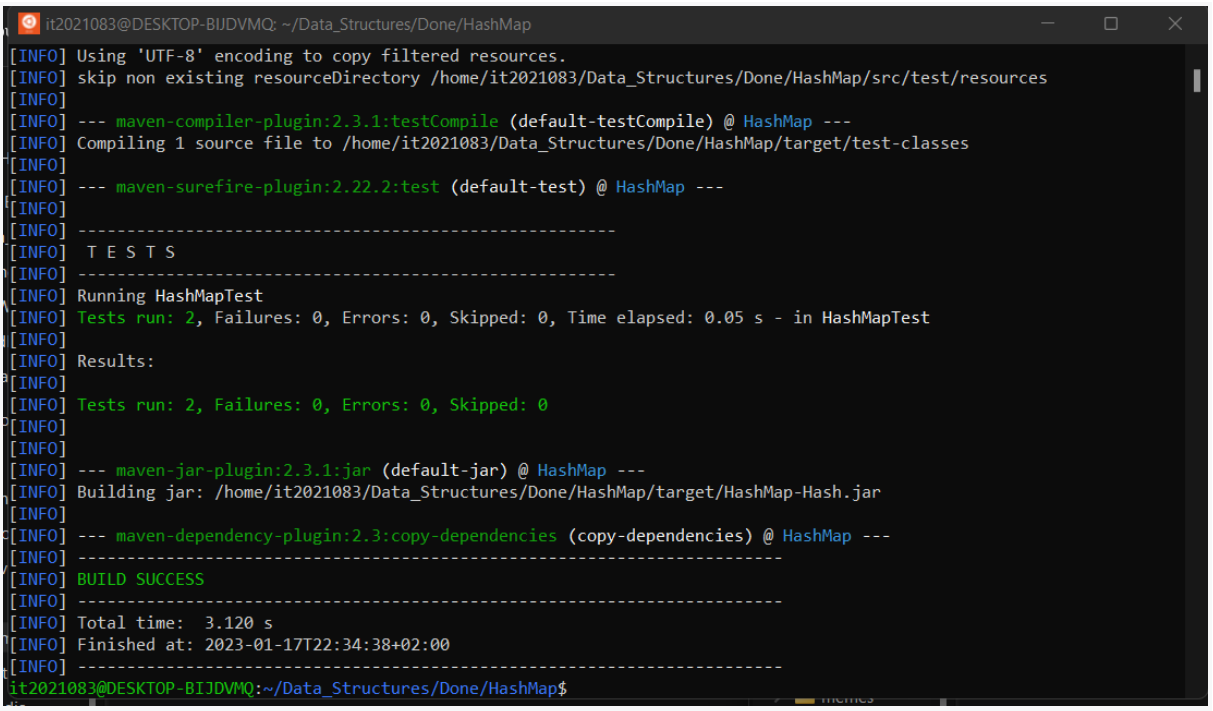

it2021083@DESKTOP-BIJDVMQ: ~
it2021083@DESKTOP-BIJDVMQ:~$ ls
Bro_Code Data_Structures Java_2 Leit_Syst Networks
it2021083@DESKTOP-BIJDVMQ:~$

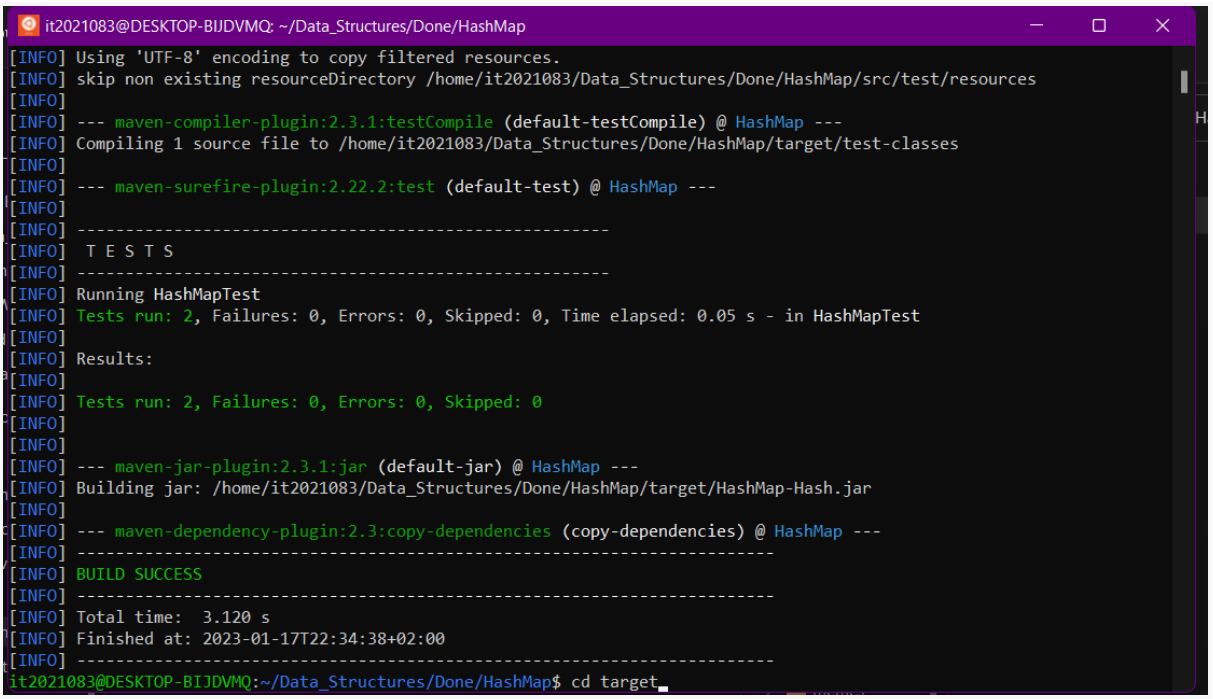

it2021083@DESKTOP-BIJDVMQ: ~
it2021083@DESKTOP-BIJDVMQ:~$ ls
Bro_Code Data_Structures Java_2 Leit_Syst Networks
it2021083@DESKTOP-BIJDVMQ:~$ cd Data_Structures

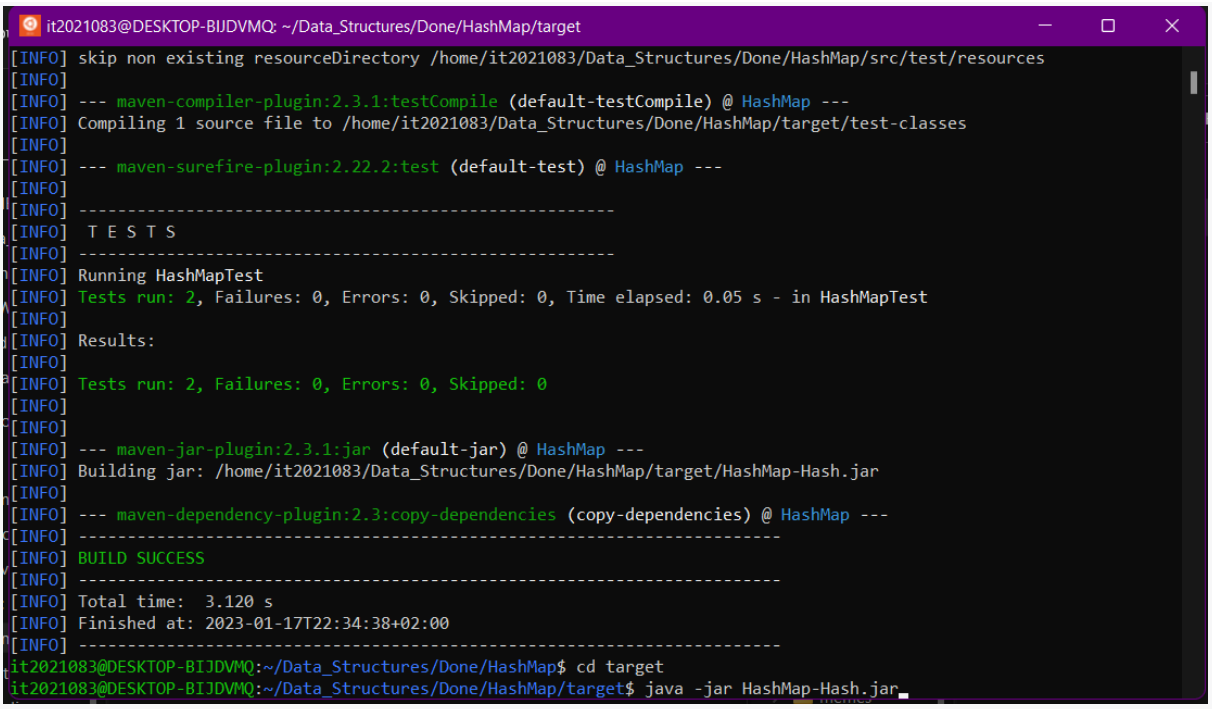

it2021083@DESKTOP-BIJDVMQ: ~/Data_Structures/Done
it2021083@DESKTOP-BIJDVMQ:~$ ls
Bro_Code Data_Structures Java_2 Leit_Syst Networks
it2021083@DESKTOP-BIJDVMQ:~$ cd Data_Structures
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures$ cd Done
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done$ cd HashMap

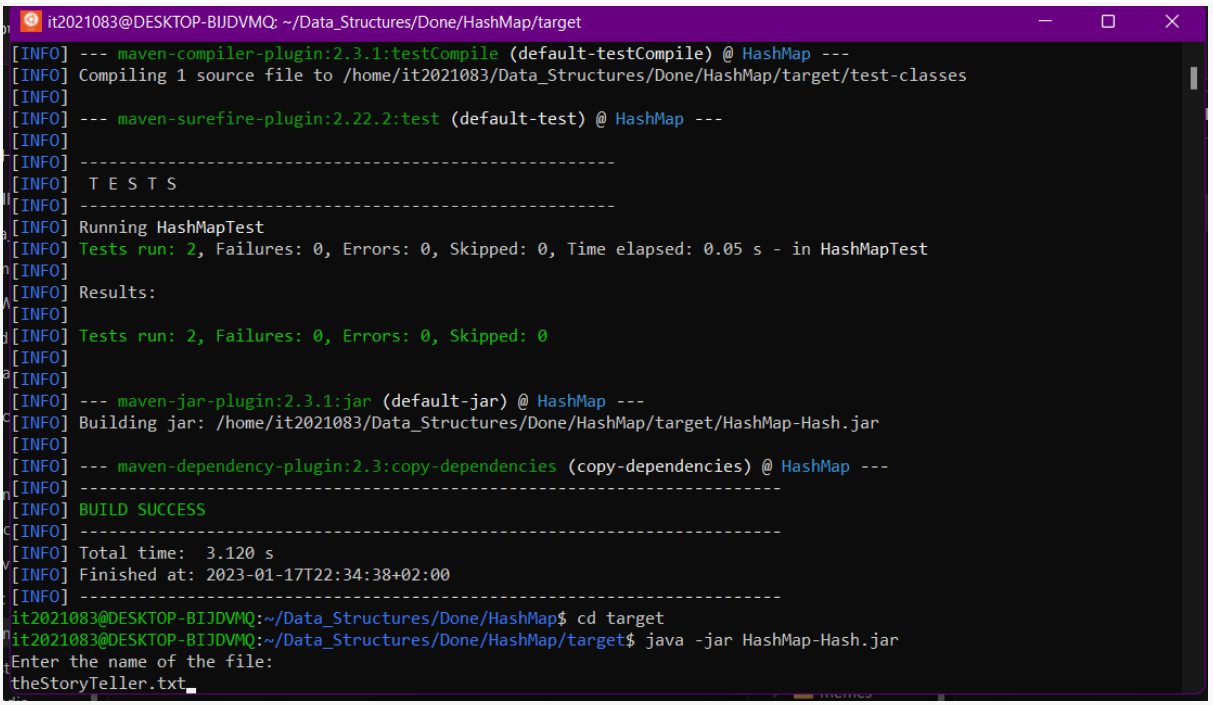

it2021083@DESKTOP-BIJDVMQ: ~/Data_Structures/Done/HashMap
it2021083@DESKTOP-BIJDVMQ:~$ ls
Bro_Code Data_Structures Java_2 Leit_Syst Networks
it2021083@DESKTOP-BIJDVMQ:~$ cd Data_Structures
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures$ cd Done
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done$ cd HashMap
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done/HashMap$ mvn clean package

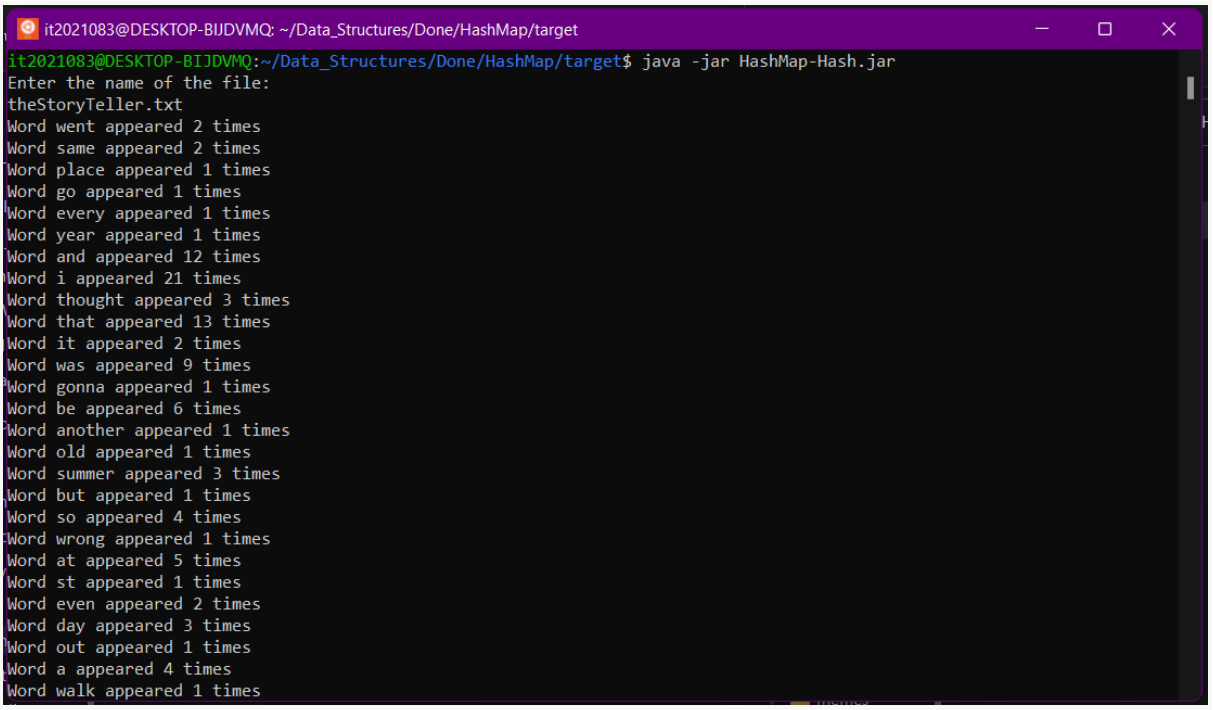

it2021083@DESKTOP-BIJDVMQ: ~/Data_Structures/HashMap
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< com.potis.hashmap:HashMap >----------------------
[INFO] Building HashMap Hash
[INFO] --------------------------------[jar]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ HashMap ---
[INFO] Deleting /home/it2021083/Data_Structures/HashMap/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ HashMap ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/it2021083/Data_Structures/HashMap/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.1:compile (default-compile) @ HashMap ---
[INFO] Compiling 6 source files to /home/it2021083/Data_Structures/HashMap/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ HashMap ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/it2021083/Data_Structures/HashMap/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.1:testCompile (default-testCompile) @ HashMap ---
[INFO] Compiling 1 source file to /home/it2021083/Data_Structures/HashMap/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ HashMap ---
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running HashMapTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.104 s - in HashMapTest

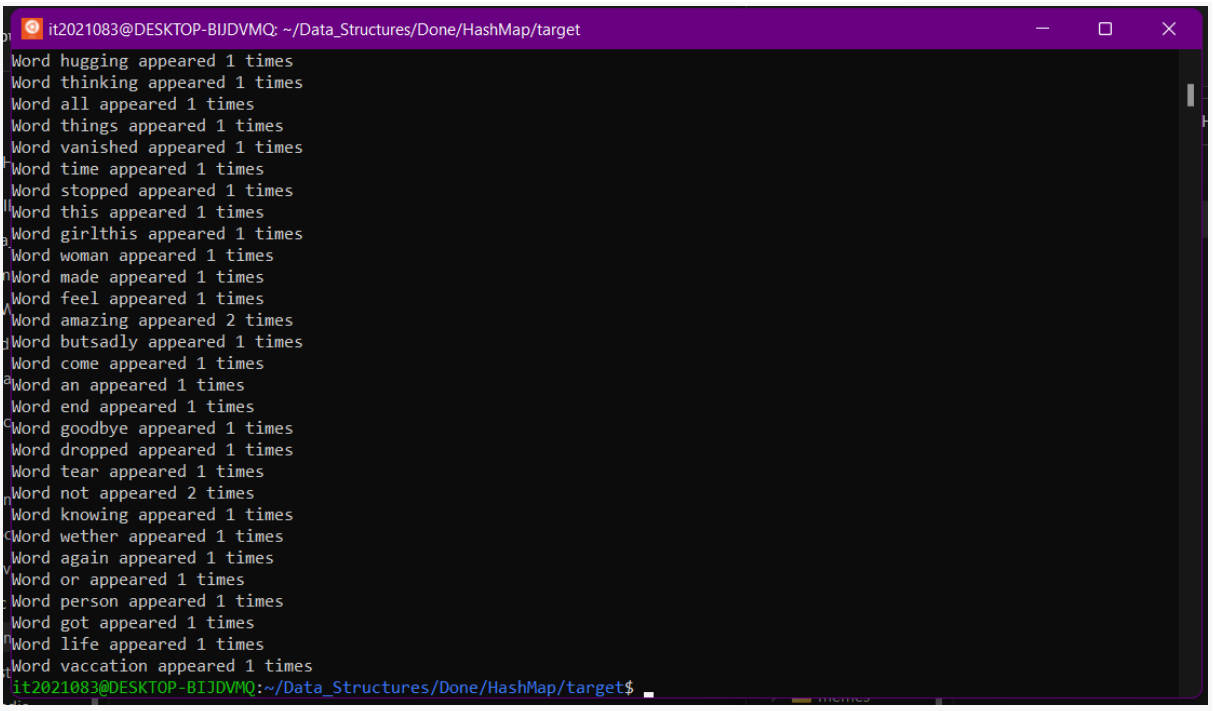

it2021083@DESKTOP-BIJDVMQ: ~/Data_Structures/Done/HashMap
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/it2021083/Data_Structures/Done/HashMap/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.1:testCompile (default-testCompile) @ HashMap ---
[INFO] Compiling 1 source file to /home/it2021083/Data_Structures/Done/HashMap/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ HashMap ---
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running HashMapTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.05 s - in HashMapTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ HashMap ---
[INFO] Building jar: /home/it2021083/Data_Structures/Done/HashMap/target/HashMap-Hash.jar
[INFO]
[INFO] --- maven-dependency-plugin:2.3:copy-dependencies (copy-dependencies) @ HashMap ---
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 3.120 s
[INFO] Finished at: 2023-01-17T22:34:38+02:00
[INFO] --
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done/HashMap$


it2021083@DESKTOP-BIJDVMQ: ~/Data_Structures/Done/HashMap
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/it2021083/Data_Structures/Done/HashMap/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.1:testCompile (default-testCompile) @ HashMap ---
[INFO] Compiling 1 source file to /home/it2021083/Data_Structures/Done/HashMap/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ HashMap ---
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running HashMapTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.05 s - in HashMapTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ HashMap ---
[INFO] Building jar: /home/it2021083/Data_Structures/Done/HashMap/target/HashMap-Hash.jar
[INFO]
[INFO] --- maven-dependency-plugin:2.3:copy-dependencies (copy-dependencies) @ HashMap ---
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 3.120 s
[INFO] Finished at: 2023-01-17T22:34:38+02:00
[INFO] --
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done/HashMap$ cd target


it2021083@DESKTOP-BIJDVMQ: ~/Data_Structures/Done/HashMap/target
[INFO] skip non existing resourceDirectory /home/it2021083/Data_Structures/Done/HashMap/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.1:testCompile (default-testCompile) @ HashMap ---
[INFO] Compiling 1 source file to /home/it2021083/Data_Structures/Done/HashMap/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ HashMap ---
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running HashMapTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.05 s - in HashMapTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ HashMap ---
[INFO] Building jar: /home/it2021083/Data_Structures/Done/HashMap/target/HashMap-Hash.jar
[INFO]
[INFO] --- maven-dependency-plugin:2.3:copy-dependencies (copy-dependencies) @ HashMap ---
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 3.120 s
[INFO] Finished at: 2023-01-17T22:34:38+02:00
[INFO] --
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done/HashMap$ cd target
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done/HashMap/target$ java -jar HashMap-Hash.jar


it2021083@DESKTOP-BIJDVMQ: ~/Data_Structures/Done/HashMap/target
[INFO] --- maven-compiler-plugin:2.3.1:testCompile (default-testCompile) @ HashMap ---
[INFO] Compiling 1 source file to /home/it2021083/Data_Structures/Done/HashMap/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ HashMap ---
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running HashMapTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.05 s - in HashMapTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ HashMap ---
[INFO] Building jar: /home/it2021083/Data_Structures/Done/HashMap/target/HashMap-Hash.jar
[INFO]
[INFO] --- maven-dependency-plugin:2.3:copy-dependencies (copy-dependencies) @ HashMap ---
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 3.120 s
[INFO] Finished at: 2023-01-17T22:34:38+02:00
[INFO] --
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done/HashMap$ cd target
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done/HashMap/target$ java -jar HashMap-Hash.jar
Enter the name of the file:
theStoryTeller.txt


it2021083@DESKTOP-BIJDVMQ: ~/Data_Structures/Done/HashMap/target
it2021083@DESKTOP-BIJDVMQ:~/Data_Structures/Done/HashMap/target$ java -jar HashMap-Hash.jar
Enter the name of the file:
theStoryTeller.txt
Word went appeared 2 times
Word same appeared 2 times
Word place appeared 1 times
Word go appeared 1 times
Word every appeared 1 times
Word year appeared 1 times
Word and appeared 12 times
Word i appeared 21 times
Word thought appeared 3 times
Word that appeared 13 times
Word it appeared 2 times
Word was appeared 9 times
Word gonna appeared 1 times
Word be appeared 6 times
Word another appeared 1 times
Word old appeared 1 times
Word summer appeared 3 times
Word but appeared 1 times
Word so appeared 4 times
Word wrong appeared 1 times
Word at appeared 5 times
Word st appeared 1 times
Word even appeared 2 times
Word day appeared 3 times
Word out appeared 1 times
Word a appeared 4 times
Word walk appeared 1 times



Το theStoryTeller.txt αρχείο που χρησιμοποιείται στο παράδειγμα της εφαρμογής θα βρίσκεται στο αρχείο .zip

Κατά τη διάρκεια της λήψης των εικόνων για την επίδειξη της εφαρμογής έγινε διαγραφή του αχρηστου υποκαταλόγου done/ (Έχει διαγραφεί σε μία από τις εικόνες(5η εικόνα) αλλά στις υπόλοιπες υπάρχει κανονικά) που όμως δεν έχει καμία λειτουργικότητα.