



Chongqing University of Technology

《Information Retrieval》

Big Work

Academic Term: 2020

Class:

Student ID: 62017010082

Student Name: 尹莱姆

Teacher: Tan Shu qui

Score:

Table of Contents

1 Overview	3
2 Way to Run the Project	3
2.1 Windows	3
3 Project Description	5
3.1 Reading Big File Method	5
3.2 Make Directory And 10 File	7
3.3 Write to File Method	9
3.4 Reading 10Doc file	11
4 Result	15
5 Summary	16

1 Overview

An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines. In information retrieval a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy. An object is an entity that is represented by information in a content collection or database. User queries are matched against the database information. However, as opposed to classical SQL queries of a database, in information retrieval the results returned may or may not match the query, so results are typically ranked. This ranking of results is a key difference of information retrieval searching compared to database searching. Depending on the application the data objects may be, for example, text documents, images, audio, mind map or videos. Often the documents themselves are not kept or stored directly in the IR system, but are instead represented in the system by document surrogates or metadata. Most IR systems compute a numeric score on how well each object in the database matches the query, and rank the objects according to this value. The top ranking objects are then shown to the user. The process may then be iterated if the user wishes to refine the query.

2 Way to Run the Project

2.1 Windows

Windows and Linux are not same Operating system so everything is depended on the Operating system but java program is not operating depended, finally you can easily run the projects by using same method in windows and Linux. In Linux has another different way to run the project it will be discussed on

Linux part. First of all unzip the file which I sent you before. Keep the file in Local Disk(C :) Named BigWork.

Drivers	6/21/2020 12:51 AM	File folder
Final	7/1/2020 5:18 AM	File folder
Intel	6/13/2020 9:25 AM	File folder
My Drivers	6/21/2020 12:54 AM	File folder
PerfLogs	3/19/2019 10:52 AM	File folder
Program Files	6/27/2020 6:32 AM	File folder
Program Files (x86)	6/21/2020 12:54 AM	File folder
SQL2019	6/14/2020 11:15 PM	File folder
TOMPCAT	6/15/2020 7:00 AM	File folder
Users	3/19/2019 6:05 PM	File folder
Windows	6/30/2020 11:00 AM	File folder

Figure 2-1: keep project on right destination

In second step open the IDE by which one you will run the project. I have used here Intillj Ide and import the project from Local Disk(C :) named Final and run the project.

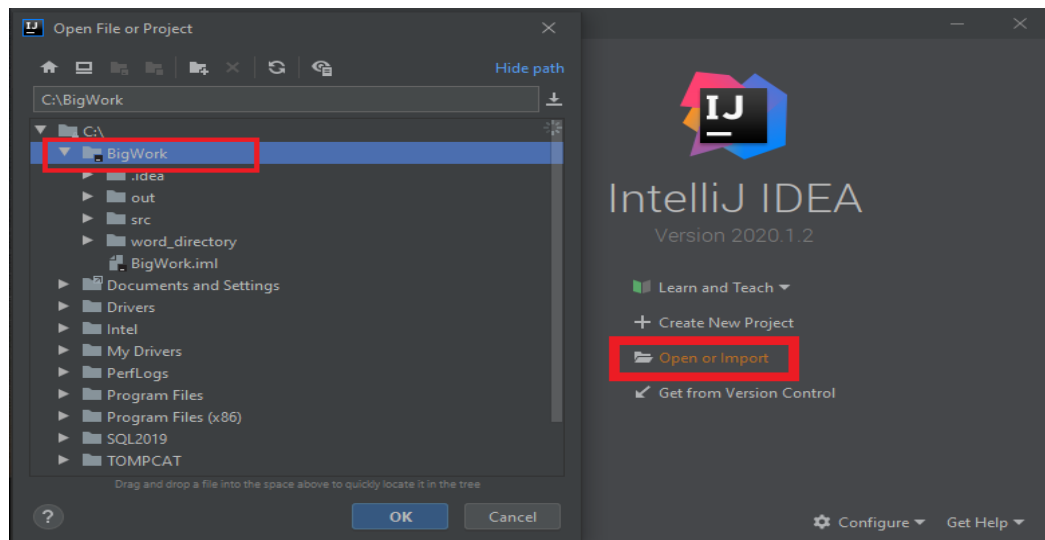


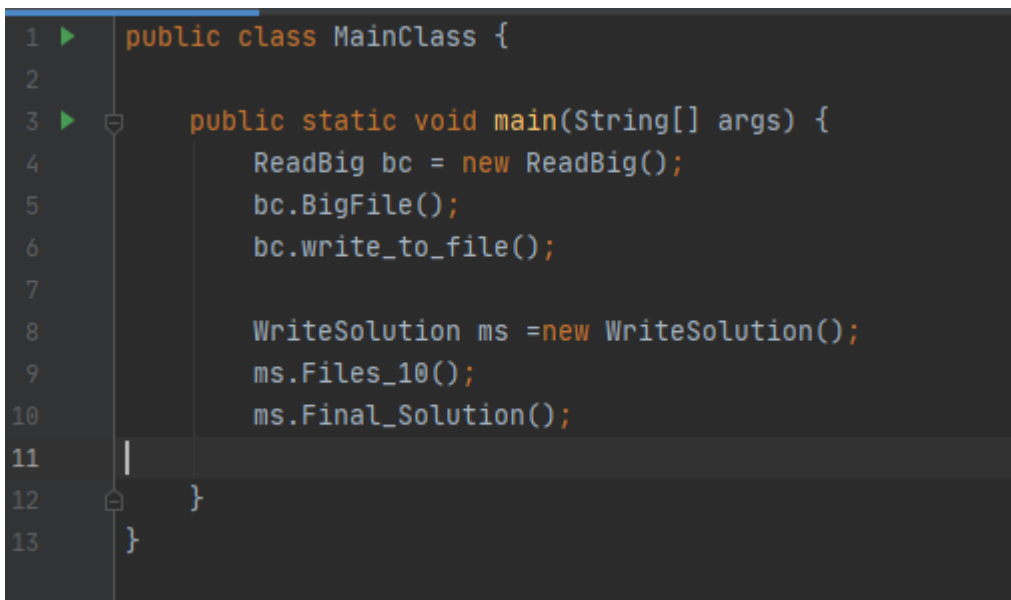
Figure 2-2: Open project

Finally you have ran the project properly and considered the output.

3 Project Description

Mainly the project has been done in four part there are four different method is

- ❖ write_to_file();
- ❖ Files_10();
- ❖ Final_Solution();

A screenshot of a code editor showing the main method of a Java class named MainClass. The code is as follows:

```
1 public class MainClass {  
2  
3     public static void main(String[] args) {  
4         ReadBig bc = new ReadBig();  
5         bc.BigFile();  
6         bc.write_to_file();  
7  
8         WriteSolution ms = new WriteSolution();  
9         ms.Files_10();  
10        ms.Final_Solution();  
11  
12    }  
13 }
```

Figure 3-1: Main Method

3.1 Reading Big File Method

In the Reading Big File method this method used for read large file where we putted 3000 word's from this file I have read 3000 word's and putted the word's int the collection named BigList . In this method just done the count words and put word's into the collection.

```

15 public void BigFile(){
16     try{
17         File file;
18         Scanner in;
19         file = new File( pathname: "C:/Final/word_directory/Large_Document.txt");
20         in = new Scanner(file);
21         while(in.hasNext()){
22             String s = in.next();
23             BigList.add(s);
24             words_count ++;
25         }
26         System.out.println("total "+words_count+" words in the Large Document");
27     }
28     catch (FileNotFoundException e){
29         System.out.println("File not Found ");
30     }
31     catch (IOException e) {
32         e.printStackTrace();
33     }
34 }
35 }
36 }
37 }

```

Figure 3-2: Read Big File method

```

public void BigFile(){
    try{
        File file;
        Scanner in;
        file = new File("C:/Final/word_directory/Large_Document.txt");
        in = new Scanner(file);
        while(in.hasNext()){
            String s = in.next();
            BigList.add(s);
            words_count ++;
        }
        System.out.println("total "+words_count+" words in the Large Document");
    }
}

```

```

    }
    catch (FileNotFoundException e){
        System.out.println("File not Found ");
    }
    catch (IOException e) {
        e.printStackTrace();
    }
}

```

3.2 Make Directory And 10 File

This is the second method to solve the project. The method has been used for making directory called name is 10Doc and create 10 empty file for keep randomly 500 words from the collection. Already have been kept 3000 words in collection. Now it's time to put randomly 500 words from the collection. The algorithm is for the this method is following

Declare 10Doc folder

Check the folder is exists or not

If not exists then make folder named 10Doc

Lop: 10 for creating 10 file

End

```

public void Files_10(){
    File folder;
    try{
        folder = new File( pathname: "C:/Final/word_directory/10Doc");
        if(!folder.exists()){
            folder.mkdir();
            for(int i=1; i<=10; i++){
                folder = new File( pathname: "C:/Final/word_directory/10Doc/file"+i+".txt");
                folder.createNewFile();
            }
        }
    }catch (Exception e){
        System.out.println("problem in making 10 documents block ");
    }
}

```

Figure 3-3: make Directory method

```

public void Files_10(){
    File folder;
    try{
        folder = new File("C:/Final/word_directory/10Doc");
        if(!folder.exists()){
            folder.mkdir();
            for(int i=1; i<=10; i++){
                folder = new File("C:/Final/word_directory/10Doc/file"+i+".txt");
                folder.createNewFile();
            }
        }
    }

    }catch (Exception e){
        System.out.println("problem in making 10 documents block ");
    }
}

```


3.3 Write to File Method

This is the method for write the words 500 words from the collection keep said that the 500 words will be randomly from the collection. previously I have solved the others two method where I did reading big file and make a directory if not exist and create 10 file by using for loop for keep 500 words from collection. Here is write to file method is work for put 500 words.

```
38 public void write_to_file(){
39
40
41     try{
42         File file;
43         FileWriter fr;
44         Writer output;
45         for(int i=1; i<=10; i++){
46             file = new File( pathname: "C:/Final/word_directory/10Doc/file"+i+".txt");
47             fr = new FileWriter(file);
48             output =new BufferedWriter(fr);
49
50             int a =(int) Math.floor(Math.random() * (2000 - 500) + 500);
51             int b = a - 50;
52             for(int j =b; j<=a; j++){
53
54                 output.write( str: BigList.get(j).toString()+" ");
55                 String str = (String) BigList.get(j);
56
57                 for (int d=0; d<str.length(); d++) {
58
59                     if(str.length()==1)
60                         break;
61                     else if (str.charAt(d) == '.') {
62                         output.write( str: "\n");
```

Figure 3-4: method for write to file

```
public void write_to_file(){
    try{
        File file;
        FileWriter fr;
        Writer output;
        for(int i=1; i<=10; i++){
```

```

file = new File("C:/Final/word_directory/10Doc/file"+i+".txt");
fr = new FileWriter(file);
output =new BufferedWriter(fr);

int a =(int) Math.floor(Math.random() * (2000 - 500) + 500);
int b = a - 50;
for(int j =b; j<=a; j++){

    output.write(BigList.get(j).toString()+" ");
    String str = (String) BigList.get(j);

    for (int d=0; d<str.length(); d++) {

        if(str.length()==1)
            break;
        else if (str.charAt(d) == '.') {
            output.write("\n");
        }
    }
}
output.close();
}

System.out.println("has been inputted in the 10 documents ");
}catch (Exception e){
    System.out.println("problem in write to file");
}

```

}

3.4 Reading 10Doc file

This is the important method for the projects . In this method has finished the important things. First of all has been read 10 file one by one and find the absolute output. There is 8 loop firstly used a loop for reading 10 file every time after reading the file the output will be shown below and the loop is going till unfinished the reading 10 file. Secondly counted the number of line and put words to a collection for the first file.

```
58     mainarray.add(s);
59     p++;
60 }
61 wordCount = wordCount + words.length;
62 int repeat = 0;
63 int position = 0;
64 for(int g = 0; g<=line_no; g++){
65     for(int x = 0; x<=mainarray.size()-1; x++){
66
67         for(int f = 0; f<x; f++){
68             String i = mainarray.get(x).toString();
69             if(i.equals(mainarray.get(f))){
70                 x++;
71             }
72         }
73         String s = mainarray.get(x).toString();
74         for(int e = x+1; e<mainarray.size()-1; e++){
75
76             if(s.equals(mainarray.get(e))){
77                 repeat++;
78                 position = e;
79             }
80         }
81         System.out.println(mainarray.get(x)+"
82         repeat =0;
83         position = 0;
84     }
```

Figure 3-5: first part of method four

Thirdly for every line and in 4th loop for checking the word is already read if read then break the loop for this time and going to the next loop where is checked the occurrences and the position of occurrence and all the result puts to some different

array-list for printing the result together . And for printing the result together has been used another loop. The algorithm for this method is following

```
while((line = reader.readLine())!=null){ //(line = reader.readLine())!=null
    tendoc = new ArrayList();
    line_no++;
    String[] words = line.split( regex: " ");
    //String words = in.next();
    int h =0;
    while(h <=words.length-1){
        String s = in.next();
        tendoc.add(s);
        h++;
    }
    wordCount = wordCount + words.length;
    int repeat = 0;
    int position =0;

    for(int g = 1; g<=line_no; g++){ //line_no
        for(int w = 0; w<=tendoc.size()-1; w++){ //tendoc.size()-1
            position = 0;
            int f = 0;
            for(f = 0; f<w; f++){
                if(f==w)
                    break;
                String q = tendoc.get(f).toString();
                String i = tendoc.get(w).toString();
                if(i.equals(q)){
                    //tendoc.add(w,tendoc.get(w+1));
                    break;
                }
            }
        }
    }
}
```

Figure 3-6: last part of method four

```
public void Final_Solution(){
    try{
        File file;
        FileInputStream fileStream;
        InputStreamReader input ;
        BufferedReader reader ;
        Scanner in;
        ArrayList mainarray;
```

```

for(int v=1; v<=10; v++){

    file = new File("C:/Final/word_directory/10Doc/file"+v+".txt");
    fileStream = new FileInputStream(file);
    input = new InputStreamReader(fileStream);
    reader = new BufferedReader(input);
    in = new Scanner(file);

    int line_no = 0;
    int wordCount = 0;

    String line;
    System.out.println("file "+v+" line Number, repetition times, repetition
position ");
    System.out.println("          line    repetition_times    position");

    while((line = reader.readLine()) != null){
        mainarray = new ArrayList();
        line_no++;
        String[] words = line.split(" ");
        int p=0;
        while(p <=words.length-1){
            String s = in.next();
            mainarray.add(s);
            p++;
        }
    }
}

```

```

wordCount = wordCount + words.length;
int repeat = 0;
int position = 0;
for(int g = 0; g<=line_no; g++){
    for(int x = 0; x<=mainarray.size()-1; x++){

        for(int f = 0; f<x; f++){
            String i = mainarray.get(x).toString();
            if(i.equals(mainarray.get(f))){
                x++;
            }
        }
        String s = mainarray.get(x).toString();
        for(int e = x+1; e<mainarray.size()-1; e++){

            if(s.equals(mainarray.get(e))){
                repeat++;
                position = e;
            }
        }
        System.out.println(mainarray.get(x)+"          ["+line_no+"
"+repeat+"          "+position+"]");
        repeat = 0;
        position = 0;
    }
}
}

```

```

        System.out.println("total line "+line_no);
        System.out.println("total "+wordCount+" words");
    }
} catch (Exception e){
    System.out.println("problem in reading 10 doc file");
}
}

```

4 Result

At the last solved the problem with a absolute solution. In the output window there are many more output but the output is separate for a specific file. For the first file1 in the output window there has been said that is the file1. And for every file has a specific output with mentioned the file name. Another thing in the output window is first is word which is occurred for several time after that the line number which line's word is happening occurrences and then mentioned the occurrences times or repetition times finally the position has mentioned that you can find the word in the file easily.

```

"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" "-javaagent:C:\Prog
total 3000 words in the Large Document
has been inputted in the 10 documents
file 1 line Number, repetition times, repetition position
      line      repetition_times      position
donec      [1      0      0]
sagittis    [1      0      0]
condimentum      [1      0      0]
malesuada    [1      0      0]
iaculis.     [1      0      0]
donec      [1      0      0]
sagittis    [1      0      0]
condimentum      [1      0      0]
malesuada    [1      0      0]
iaculis.     [1      0      0]
Laoreet     [2      0      0]
nostra      [2      0      0]
ante.       [2      0      0]
Laoreet     [2      0      0]
nostra      [2      0      0]
ante.       [2      0      0]

```

Figure 4-1: output for file1

5 Summary

Programming is the process of designing and building an executable computer program to accomplish a specific computing result. Programming involves tasks such as: analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms in a chosen programming language (commonly referred to as coding). Finally success to solve the problem. There are so much trouble to finish the Big Work. Some times has gave up that is not possible to solve it but finally success.