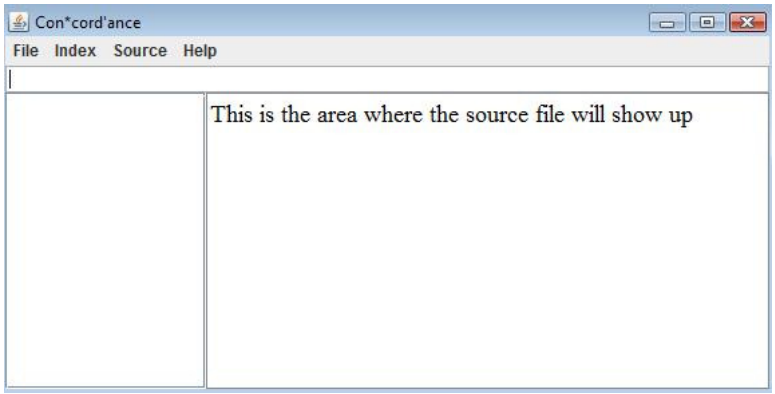
CS21120: Data Structures and Algorithm Analysis
Assignment 1 – Concordance

Stoyko Abrashev
Aberystwyth University
Computer Science
sta9@aber.ac.uk

1. INTRODUCTION

The program was a big challenge for me but I think I managed to do everything I had in mind.
Firstly I was doing the classes in my worksheets directory and I was working just with the system.out.println but after doing the three tasks in the assignment I figured that I can do nice GUI with everything working correctly and without errors.
I have used a lot of JOptionPanes to show the user messages about what he have to do in order to search a source with index words.
I wanted to highlight the line or the word itself in the line but couldn't do it. For my JTextArea there are methods like setcarotposition() and setselectionstart().Anyway I will investigate in the future and update the program components.



2. ALGORITHM

In the beginning I was wondering if the keys in the Hashtable has to be incremented integers or actual words to be keys.
Then I decided to use the integers for keys because they are unique and I can use loops easier and the other thing is line numbers data structure which is
LinkedList of Words. Every Word has it's own LinkedList of Integers (where I keep line numbers for every word). In Word class there is a very important method called ***checkLine(int line)*** *which checks whether the line is not already in lines linked list(if a word is repeating on a line more than one time it's line number is stored in lines just once).* ***toString()*** *in Word is returning a string of lines or a emptyString if* ***lines.isEmpty()*** *.*
My Algorithm to detect index words is in ***searchSource(File file)*** *method. I read in the file only* ***if (!mywords.isEmpty()).*** *Before the while (****hasNextLine()****) loop I have an int called* ***line*** *which is incremented by 1 every time. In the loop I read the line and* ***if (NextLine.equals(" "))*** *the loop* ***continue;*** *.The regex is* ***"\\W+"*** *a non-word character which occur one or more times. I have array of 50 strings which becomes = to* ***nextLine.split(regex).*** *Then for every string in the array(it will be just words* ***without any punctuation)*** *we check* ***if(hashtable.containsValue(array[i]).toLowerCase())*** *and if it is we go to* ***for loop*** *every element in hashtable(mywords). Then if*

mywords.get[loopnum].equals(array[i].toLowerCase()) *we have if inside it which checks*
if(linkl.get[loopnum].checkLine(line)==false *and only then* ***linkl.get[loopnum].addLine(line)*** *and* ***break;*** *the loop.*

I use JList for displaying the hashtable keys in the frame. The thing was difficult because when I sort the values with ***sortWords()*** *method I had to reset the ids of the JList and then the hashtable keys(and linkl linked list line numbers as well) had different ids then the list .So I made method* ***printLines()*** *which print each list item and loops through keys in hashtable to find match and of couse the id of the key in hashtable so I can print and the lines correctly because otherwise for every word there are different line numbers ☺(that was the first bug here). I put a lot of if() and else if statements to check whether user have to execute some code or not. The newest thing is that it prints all that exists in the file. I put if into* ***printLines()*** *and* ***printWords()*** *loops In some places I use Scanner and in some BufferedReader. For printing I will use and call* ***printLines()*** *which tells me only about the words in the text(of course I can use system.out.println for all words if I delete if statement ☺).*
When you click on some item in the Jlist in the text field at the top there is information only for this word. When you click on a word after searchSource() you can see the line numbers of of the word if exist or just position in the list and hashtable and item number
In GameFrame there are a lot of JmenuItems, Radio buttons, MenuListener, GamePanel(this), JTextArea where will be the source text and etc. This picture is after search the ***testsource.txt***

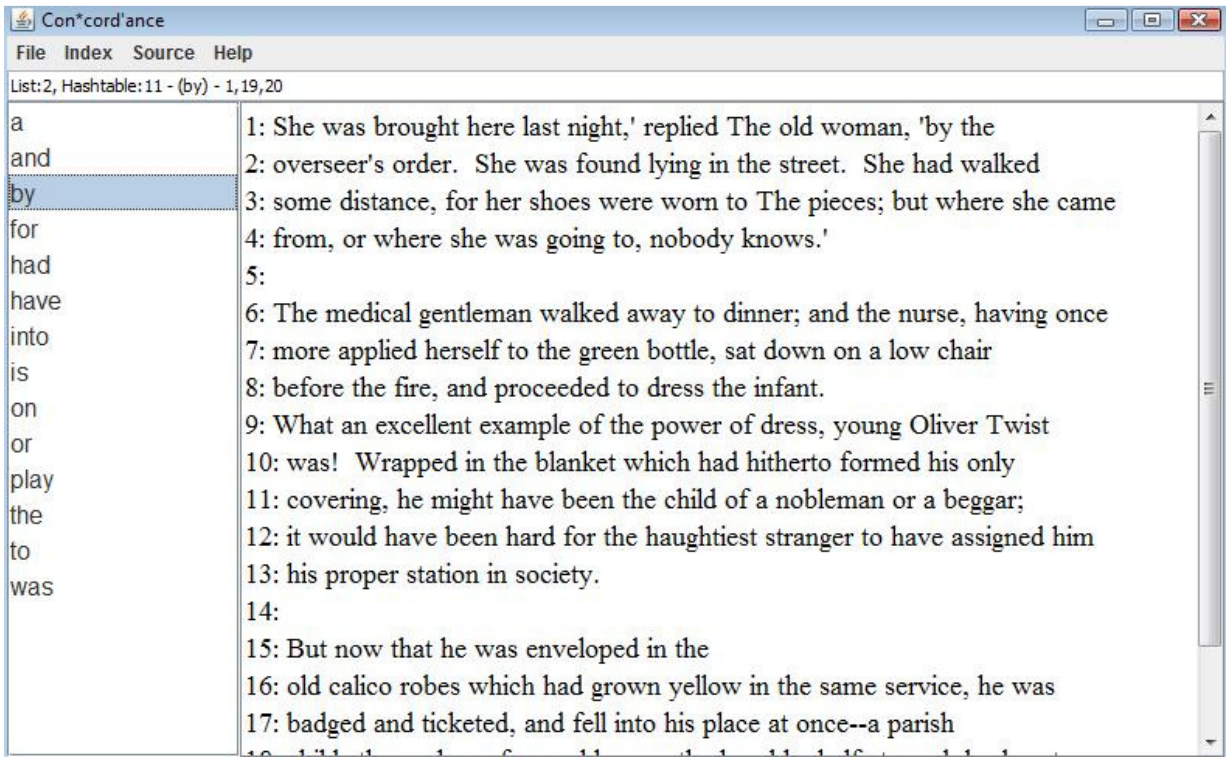
with the words from ***testindex.txt*** *(words are 12 but in the file are 14 ☺ "is" and "play" don't exist in the file.).*

```
List:0, Hash:10 :a - 7,11,17,18
List:1, Hash:6 :and - 6,8,17,19
List:2, Hash:11 :by - 1,19,20
List:3, Hash:12 :for - 3,12
List:4, Hash:13 :had - 2,10,16
List:5, Hash:8 :have - 11,12
List:6, Hash:9 :into - 17
List:8, Hash:7 :on - 7
List:9, Hash:5 :or - 4,11
List:11, Hash:0 :the - 1,2,3,6,7,8,9,10,11,12,15,16,18,19
List:12, Hash:4 :to - 3,4,6,7,8,12,18
List:13, Hash:2 :was - 1,2,4,10,15,16
```

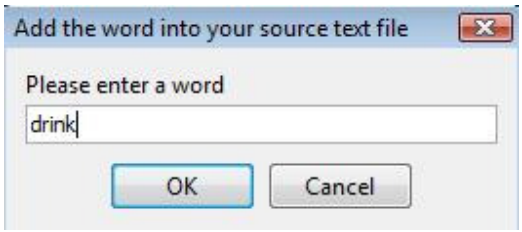
.

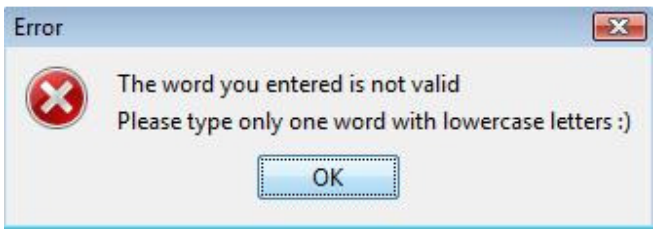
I put and three source files to choose between them. Oliver Twist, War of the Worlds and The Prince. There is a browseSource menu which allows you to choose file from any directory. There is and a menu ***AddtoYourIndex*** *where you can add words to yourindex.txt .*

The best thing is that here I use regular expressions again and you can enter only lowercase letter (one word). Here are some images of the things.

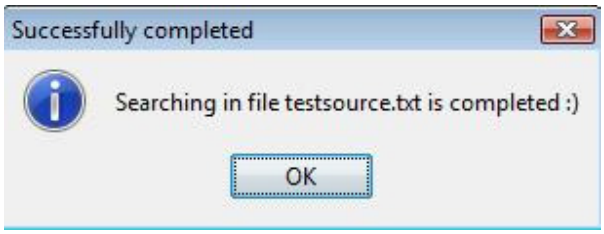

Con*cord'ance
File Index Source Help
List:2, Hashtable:11 - (by) - 1,19,20
a
and
by
for
had
have
into
is
on
or
play
the
to
was
1: She was brought here last night,' replied The old woman, 'by the
2: overseer's order. She was found lying in the street. She had walked
3: some distance, for her shoes were worn to The pieces; but where she came
4: from, or where she was going to, nobody knows.'
5:
6: The medical gentleman walked away to dinner; and the nurse, having once
7: more applied herself to the green bottle, sat down on a low chair
8: before the fire, and proceeded to dress the infant.
9: What an excellent example of the power of dress, young Oliver Twist
10: was! Wrapped in the blanket which had hitherto formed his only
11: covering, he might have been the child of a nobleman or a beggar;
12: it would have been hard for the haughtiest stranger to have assigned him
13: his proper station in society.
14:
15: But now that he was enveloped in the
16: old calico robes which had grown yellow in the same service, he was
17: badged and ticketed, and fell into his place at once--a parish

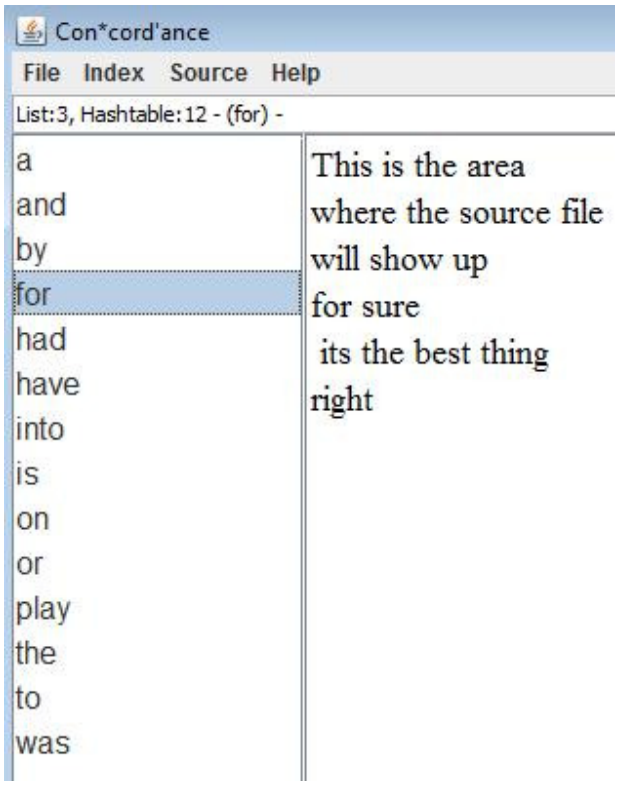

Successfully updated
The Index file is updated :)
OK


Add the word into your source text file
Please enter a word
drink
OK
Cancel

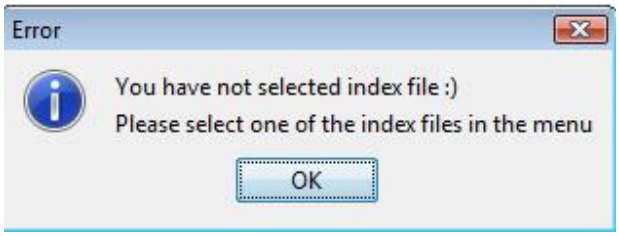

Error
The word you entered is not valid
Please type only one word with lowercase letters :)
OK

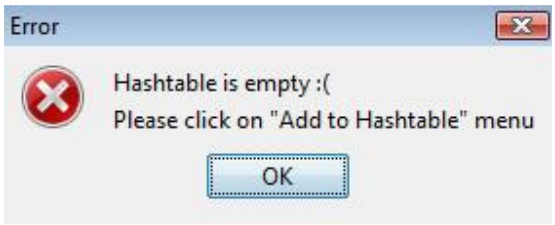

Successfully deleted
All the data in lists and hashtables and lines linked list is deleted :(
OK

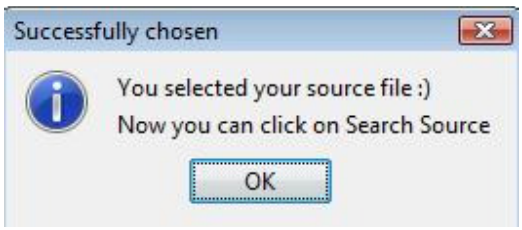

Successfully completed
Searching in file testsource.txt is completed :)
OK


Con*cord'ance
File Index Source Help
List:3, Hashtable:12 - (for) -
a
and
by
for
had
have
into
is
on
or
play
the
to
was
This is the area
where the source file
will show up
for sure
its the best thing
right


Successfully added
Words from yourindex.txt are added to a hashtable successfully :)
OK



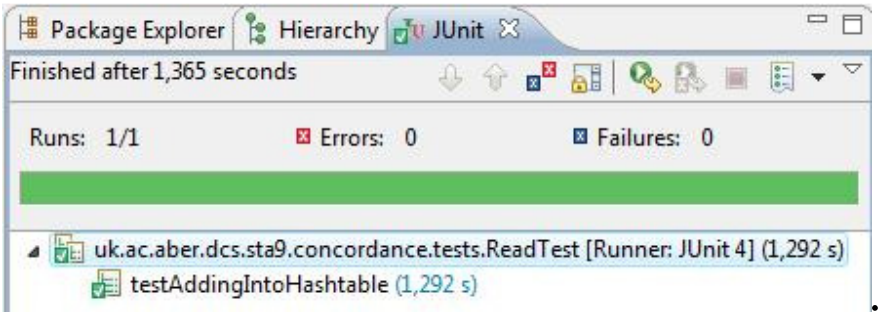




3. *TESTING*

I did two test classes and three test methods.

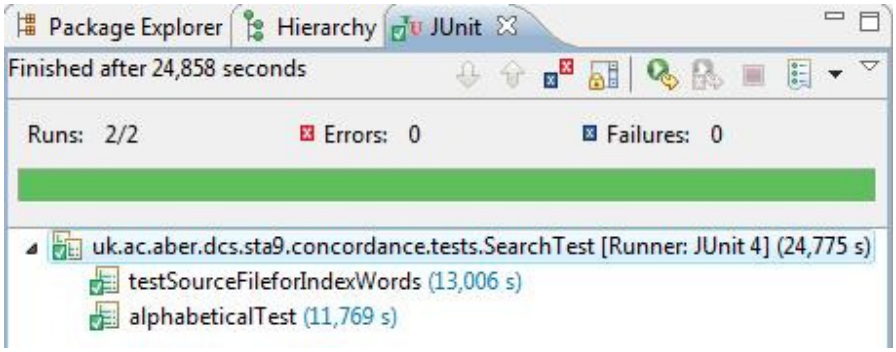
First one ***testAddingIntoHashtable()*** *add 5 words to* ***yourindex*** *and checks if they are added successfully*



. Result:

```
List:0, Hash:2 :down - Doesn't exist i
List:1, Hash:1 :grown - Doesn't exist
List:2, Hash:0 :night - Doesn't exist
List:3, Hash:4 :seek - Doesn't exist i
List:4, Hash:3 :sly - Doesn't exist in
```

Second is ***testSourceFileforIndexWords() –*** *it addsfrom testindex the words to hashtable and the* ***searchintext.*** *Then it checks for line numbers of words if they are correct. After that adds two more words from yourindex and test the line numbers of words again. Third is checking for alphabetical order where we add our index file and search in source text .*



Result here I call both methods first printLines() and then printWords():

```
List:0, Hash:6 :child - 11,18
List:1, Hash:2 :down - 7
List:2, Hash:1 :grown - 16
List:3, Hash:0 :night - 1
List:4, Hash:5 :old - 1,16
List:5, Hash:4 :seek - Doesn't exist in the source file or you have not searched for index word in the source
List:6, Hash:3 :sly - Doesn't exist in the source file or you have not searched for index word in the source t
List:0, Hash:6 :child - 11,18
List:1, Hash:2 :down - 7
List:2, Hash:1 :grown - 16
List:3, Hash:0 :night - 1
List:4, Hash:5 :old - 1,16
List:5, Hash:4 :seek - Doesn't exist in the source file or you have not searched for index word in the source
List:6, Hash:3 :sly - Doesn't exist in the source file or you have not searched for index word in the source t
0: night-1
1: grown-16
2: down-7
3: sly-Doesn't exist in the source file or you have not searched for index word in the source text yet
4: seek-Doesn't exist in the source file or you have not searched for index word in the source text yet
5: old-1,16
6: child-11,18
```