Assignment 3 – Creating a Dictionary Using Map
i.e. Specifically Using a **TreeMap** Data Structure
CST8130 – Data Structures

Useful Resources:
See TreeMap as well as course materials on Sets and Maps.

Submission and Demo are required to get a full mark.
Due date is set on Brightspace

Problem Description:
In this assignment, you will write a program which builds a tree "dictionary" of words that were found in a piece of text and keep track of how many occurrences there were of each of the words. It is the last assignment in this course.
Create an Eclipse project named **Lname_Fname_Assign#3** containing two classes namely:

- **Assign3Test** - will host the main method and the menu
- **Dictionary** - will host the data processing methods

You may need to place the text file as the word source file in the project directory.

Basic Requirements:
1. You must have a menu in your main method of the test class with the following options:

• Add words to the dictionary from file
• Search for a word and show how many times it occurred in the text
• Display the number of unique words in the dictionary
• Display the number of all words in the dictionary including duplicates
• Reset dictionary
• Ignore definite and indefinite articles (a, an, the). The default setting is false.
• Exit

2. You must use the collection class TreeMap to implement this assignment. A description of the methods available in that class can be found in the Java documentation (https://docs.oracle.com/javase/8/docs/api/java/util/TreeMap.html).
Efficient use of the class will factor in the assignment marking.

3. I have given you a Raven.txt file with Edgar Allan Poe poem. (Note you may get slightly different numbers if your editing of the word is different than mine – that is okay).

Hint: you might consider putting all the "words" that you read into either uppercase or lowercase to make sure that the words "there" and "There" count as the same.

Hint: you might collect all the "words" regardless of "Ignore definite and indefinite articles" flag setting and only modify the displayed action result for which the flag could have any impact.

To calculate number of all words you go through all tree nodes and sum up the number of all words concurrency.

Expected Output (user input is shown after each menu option selected)

```
*****************************************
DICTIONARY
*****************************************
1. Add words to the Dictionary from file
2. Search a word in the Dictionary
3. Display number of unique words in the Dictionary
4. Display number of all words in the Dictionary
5. Reset Dictionary
6. Ignore definite and indefinite articles (false)
7. Exit

Enter your option: a
Input Mismatch Exception while reading user's option from main menu



*****************************************
DICTIONARY
*****************************************
1. Add words to the Dictionary from file
2. Search a word in the Dictionary
3. Display number of unique words in the Dictionary
4. Display number of all words in the Dictionary
5. Reset Dictionary
6. Ignore definite and indefinite articles (false)
7. Exit

Enter your option: 3
Dictionary has 0 unique words

*****************************************
DICTIONARY
*****************************************
1. Add words to the Dictionary from file
2. Search a word in the Dictionary
3. Display number of unique words in the Dictionary
4. Display number of all words in the Dictionary
5. Reset Dictionary
6. Ignore definite and indefinite articles (false)
7. Exit

Enter your option: 4
Dictionary has 0 words

*****************************************
DICTIONARY
*****************************************
1. Add words to the Dictionary from file
2. Search a word in the Dictionary
3. Display number of unique words in the Dictionary
4. Display number of all words in the Dictionary
5. Reset Dictionary
6. Ignore definite and indefinite articles (false)
7. Exit
```

```
Enter your option: 2
Enter the word you want to search: test
test occurs 0 times

*****************************************
DICTIONARY
*****************************************
1. Add words to the Dictionary from file
2. Search a word in the Dictionary
3. Display number of unique words in the Dictionary
4. Display number of all words in the Dictionary
5. Reset Dictionary
6. Ignore definite and indefinite articles (false)
7. Exit

Enter your option: 6
Ignore definite and indefinite articles has been set to true

*****************************************
DICTIONARY
*****************************************
1. Add words to the Dictionary from file
2. Search a word in the Dictionary
3. Display number of unique words in the Dictionary
4. Display number of all words in the Dictionary
5. Reset Dictionary
6. Ignore definite and indefinite articles (true)
7. Exit

Enter your option: 1
Enter the word you want to search: beguiling     (you may choose to make this option 1 to open a submenu with
                                                  1. read from keyboard,  2. read from file, 3 return to main menu )
Program has read the file

*****************************************
DICTIONARY
*****************************************
1. Add words to the Dictionary from file
2. Search a word in the Dictionary
3. Display number of unique words in the Dictionary
4. Display number of all words in the Dictionary
5. Reset Dictionary
6. Ignore definite and indefinite articles (true)
7. Exit

Enter your option: 5
Program has removed all the words

*****************************************
DICTIONARY
*****************************************
```

```
1. Add words to the Dictionary from file
2. Search a word in the Dictionary
3. Display number of unique words in the Dictionary
4. Display number of all words in the Dictionary
5. Reset Dictionary
6. Ignore definite and indefinite articles (true)
7. Exit

Enter your option: 7
Good bye.... hope to see you soon
```

Submission:

You must submit zipped folder named LastnameFirstName_Assign3 by the due date and time, containing: To avoid submission missteps and preserve the File I/O pathnames, you may simply zip and submit the eclipse project directory. This must contain:

- all source code – i.e. .java files
- text file to be read.
- Javadoc for all classes, field names and methods
- Your test plan that follows the structure of test plan template provided

Failure to provide any of the above will have an adverse effect on your grade for this assignment.