Practice Lab 4

Given a set of English sentences, we would like to analyze the text to look for different patterns of usage of words - what words are used frequently, how many unique words are used. Whether words from a specified set are used, etc.

The added purpose of this exercise is to see how much of these operations can be accomplished using Collections and operations available on them, so that you, the programmer, do not have to work on detailed manipulation issues.

The input is a text string as well as a list of "reserved" words. Your program should output the following:

1. The unique words in the input:

    a. In the order they **appeared** in the input string
    b. In alphabetical order (normal **lexicographic** order)
    c. In order of increasing word **size**. That is all 1 letter words first, then 2 letter words, then 3 letter words etc. If there are multiple words with the same length, then these should be listed in lexicographic order.
    d. In order of increasing frequency of occurrence. If two words have the same frequency count, then they appear in the same order as the initial input

2. The count of the words in the input text that **start** with each letter of the alphabet
3. The usage of the **reserved** words in the input
    a. in the order they **appeared** in the input
    b. In order of **decreasing frequency** of occurrence in the input

The program should consider the following:

● You can assume there will be exactly one blank (whitespace) between words
● You will have to strip out any punctuation or special characters. Assume we have only the following punctuations in our input: . , ; : ‘ “ -
● All comparisons and output should consider only the lower-case versions of the text.

The code design should be on the following lines:
An **Analyzer** class that is provided the input string and a list of reserved words, and can be queried for each of the outputs needed above. All these interface methods should return an **ArrayList** (of the right type of element). Thus, any processing as well as use of other Collections should be hidden within Analyzer

The main/driver method creates an instance of Analyzer, reads in the input, initializes the Analyzer with this data, and queries it for each of the questions above, and prints out the elements of the list received. No printing should be done from the Analyzer class, and no processing/computation should be done in the main class.

You are strongly encouraged to use existing methods in the String class and in the Collections framework. Part of this is to identify the best Collections type to use and invoking the appropriate methods of those objects. In fact, with the use of the appropriate bulk methods of these classes, you can avoid having to explicitly iterate through the collections. In case you need to explicitly iterate through a collection, use an Iterator to do so.

Some of the available methods/classes you can look at:
● String class: split, replaceAll, toLowerCase etc.
● Conversion from arrays to lists and vice versa Arrays.asList and toArray for collections
● Set, SortedSet, HashMap, ArrayList,
● Methods on collections like retainAll, removeAll apart from the more common ones, and appropriate use of sort using Comparators (This will be covered in the next lecture)
● Initializing one type of Collection from another.

The first line of the input contains a list of reserved words. (You can assume max one line of reserved words). The rest of the input in the set of words to be analyzed

Sample Input:
groups collections elements
A collection — sometimes called a container — is simply an object that groups multiple elements into a single unit. A collections framework helps manipulate collections, creating sets of elements etc.

Output
1a: a collection sometimes called container is simply an object that groups multiple elements into single unit collections framework helps manipulate creating sets of etc
1b: <above words in sorted order>
1c: a an is of etc into sets that unit <and so on>
1d: collection sometimes called container is simply an object that groups multiple into single unit framework helps manipulate creating sets of etc elements collections a
2a:
a 5
c 6
e 3
f 1
g 1
h 1
i 2
m 2
o 2
s 4
t 1
u 1
3a: groups elements collections
3b: elements collections groups