COP3330 Object Oriented Programming
Dr. Andrew Steinberg
Spring 2022

Programming Assignment 2

What is all this text about?

Max Points 100

Due: 2/20/2022 at 11:59pm

Background Story of this Assignment

You have been learning a lot about classes, objects, and Strings. At this point in the course, you should be able to notice the dots connected with how objects work in an Object-Oriented language. In this assignment we will focus primarily on the String class as it is one of the important classes in Java to understand (besides the Object class which we will see when we get to inheritance). In this assignment, you are going to analyze text documents using the String Class.

Start Early and see the TAs and ULAs! They are here to help you! Don't procrastinate!

Assignment Details

You are going to create a nice simple text analyzer that analyzes text files through String manipulation. String manipulation allows us to analyze works of great authors. In this assignment, you will be provided with a set of text documents (random text and not so much stories) to analyze. You are going to perform three types of analyzations. One you will count number of each letter used throughout the document. Second you will count the number of words in the document total. Third, you will count the number of words used in each sentence. You are going to use a variety of built in methods from the String Class. Please look at https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html for a complete of methods in the String Class. You can use additional methods in the String Class that weren't covered. Make sure to read the Java Docs! They are very useful!

The Class Name

The class you will implement will be called `TextAnalysis`.

The Class Constructor

You are going to implement one overloaded class constructor called `TextAnalysis`. The overloaded constructor has two parameters in the signature.

1. The first parameter is an integer data type called `limit`. This value is used in determining how many sentences are going to be stored into the object of the class. For example, a text document could have 90 sentences, but the limit is 32. That means only 32 sentences will be stored in the object.
2. The second parameter is an array of String Class objects called `data`. The array stores a sentence in each element.

You do not need to implement a default constructor for this class.

The Class Attributes

You do not need to implement a default constructor for this class.

The `TextAnalysis` Class has five attributes.

1. An attribute called `data`. This attribute is a String Class array that holds the text. Each element is a sentence.
2. An attribute called `alphabet`. This attribute is a primitive integer array that stores the number of times a letter occurs in data. Each element corresponds to a letter. For example 'a' would be at index 0, 'b' at index 1, 'c' at index 2 and so on until 'z'. You can assume that case sensitivity doesn't matter. That means 'A' and 'a' are considered the same.
3. An attribute called `wordsize`. This attribute is a primitive integer array that stores the size of each word. In simple terms it will store the size the of the words in of the respective text documents. Each index represents the size the of word. For example index 1 contains the number of occurrences one size words appear in the document. **Important:** You can assume that there will only be words that have no more than 15 letters.
4. An attribute called `wordcount`. This attribute is a primitive integer array that stores the size of the actual sentence. In simple terms, the number of words per sentence. Each index represents the total number of words in the sentence. For example, index 10 represents 10 words in a sentence. **Important:** You can assume that there will only be sentences that have no more than 30 words.
5. An attribute of type int called `limit`. This attribute stores the max number of sentences to observe for the class object. Each class object will have a different limit.

Make all attributes private! It is good practice in OOP! Since you are making private attributes, please consider creating accessor methods!

The Method Signatures

You are going to implement 5 user-defined methods.

```
public void display()
```

This non-static method will display the content of text document stored in the class object data attribute. Each sentence will be displayed on its own line.

```
public void tableDisplay()
```

This non-static method will display the content of the analyses done in the text stored for the respective class object in a simple table format.

```
public void letterAnalysis()
```

This non-static method will analyze the letter characters in the document. You do not need to worry about other characters used in sentences such as the comma, period, semicolon, quotes, etc… The method will count each respective character and store the information in `alphabet` array. This information will get displayed in `tableDisplay()`.

```
public void wordAnalysis()
```

This non-static overloaded method will analyze the words in the text document. This method will count the number of words used in each sentence. The result of each sentence will be stored in the `wordsize` attribute. Do not count the special characters as part of the words. Your program should also not count digits (101, 555, etc…). This information will get displayed in `tableDisplay()`.

For example:

The irony of the situation wasn't lost on anyone in the room.

This sentence would store:

- Three 2 size words (*of, on, in*)
- Three 3 size words (*the, the, the*)
- Two 4 size words (*lost, room*)
- Two 5 size words (*irony, wasn't*)
- One 6 size word (*anyone*)
- One 9 size word (*situation*)

Note: You will be incrementing the values as you will read multiple sentences, but this is to show a small example.

```
public void wordAnalysis(int num)
```

This non-static overloaded method will analyze the words in the text document. This method will measure the length of each sentence. In simple terms you will count the number of words contained in a sentence. You can assume that all sentences will have at least one word. This method has one parameter of type int called num. The parameter `num` holds the number of sentences to be observed. For example, if `data` has 20 sentences and `num` is passed 7, then the first 7 sentences will be analyzed. The result of each sentence will be stored in the `wordcount` attribute. This information will get displayed in `tableDisplay()`. Make sure `num` is a positive number and that it does not exceed the `limit` attribute. If the condition is not satisfied, the message "`Exceeded and cannot produce an analysis on this component`" Should be displayed to the terminal.

COP3330 Object Oriented Programming
Dr. Andrew Steinberg
Spring 2022

The Provided Files

You were provided six files to assist you in this assignment.

1. A text file that shows a sample output of the program run in Eustis.
2. Another text file that shows a sample output formatted in a normal run.
3. 3 Text files that contains sentences to analyze.
4. A python script that will test and verify that your code output is correct.
5. A runner class that contains the main method.

Requirements

Your program must follow these requirements.

- The output must match exactly (this includes case sensitivity, white space, and even new lines). Any differences in the output will cause the grader script to say the output is not correct. Test with the script provided in order to receive potential full credit. Points will be deducted! Check out the sample text file.
- Make your class attributes private. It is good practice!
- Do not change the method signatures. Any changes to the method signatures will result in points being deducted.
- You are welcome to create additional helper methods as long as you do not remove the required methods that have been asked in this assignment.
- Do not make any changes to the Runner file that was provided for you. Any changes will result in points being deducted.
- Your code must work on Eustis. If it does not work on Eustis, points will be deducted and not changed as mentioned previously.
- Make sure you include a comment header. Check the assignment page of how it should. **It should be exactly the first line of your Java source file. See the assignment page for more info.**

The Rubric

Please see the assignment page in Webcourses for the Rubric of how the assignment will be evaluated.

Testing the Solution with the Python Script

Once you have completed the assignment, you will need to test it to make sure it matches Dr. Steinberg's sample output. In Eustis, make sure to upload the Python script, your Java Solution, the Runner file, and sample text output. I would highly recommend that you have a folder with those 6 files only.

Once all of those files are uploaded into a directory in Eustis, run the command "`python3 p2testscript.py`". The script will compile your Java source and execute it. You will then see the result in the form of a happy face or sad face. The happy face means your output was correct. The sad face means something was off with the output. Remember, the script is very

picky with white space and new lines. Make sure you do not add any extra trailing white space or new lines. Look at the sample text output.

Tips in Being Successful

Here are some tips and tricks that will help you with this assignment and make the experience enjoyable.

- Do not try to write out all the code and build it at the end to find syntax errors. For each new line of code written (my rule of thumb is 2-3 lines), build it to see if it compiles successfully. **It will go a long way!**
- After any successful build, run the code to see what happens and what current state you are at with the program writing so you know what to do next! If the program performs what you expected, you can then move onto the next step of the code writing. If you try to write everything at once and build it successfully to find out it doesn't work properly, you will get frustrated trying find out the logical error in your code! **Remember, logical errors are the hardest to fix and identify in a program!**
- Start the assignment early! Do not wait last minute (the day of) to begin the assignment.
- Ask questions! It's ok to ask questions. If there are any clarifications needed, please ask TAs/ULAs and the Instructor! We are here to help!!! You can also utilize the discussion board on Webcourses to share a general question about the program as long as it doesn't violate the academic dishonesty policy.