wordCounter

a Java program to print a text count of words
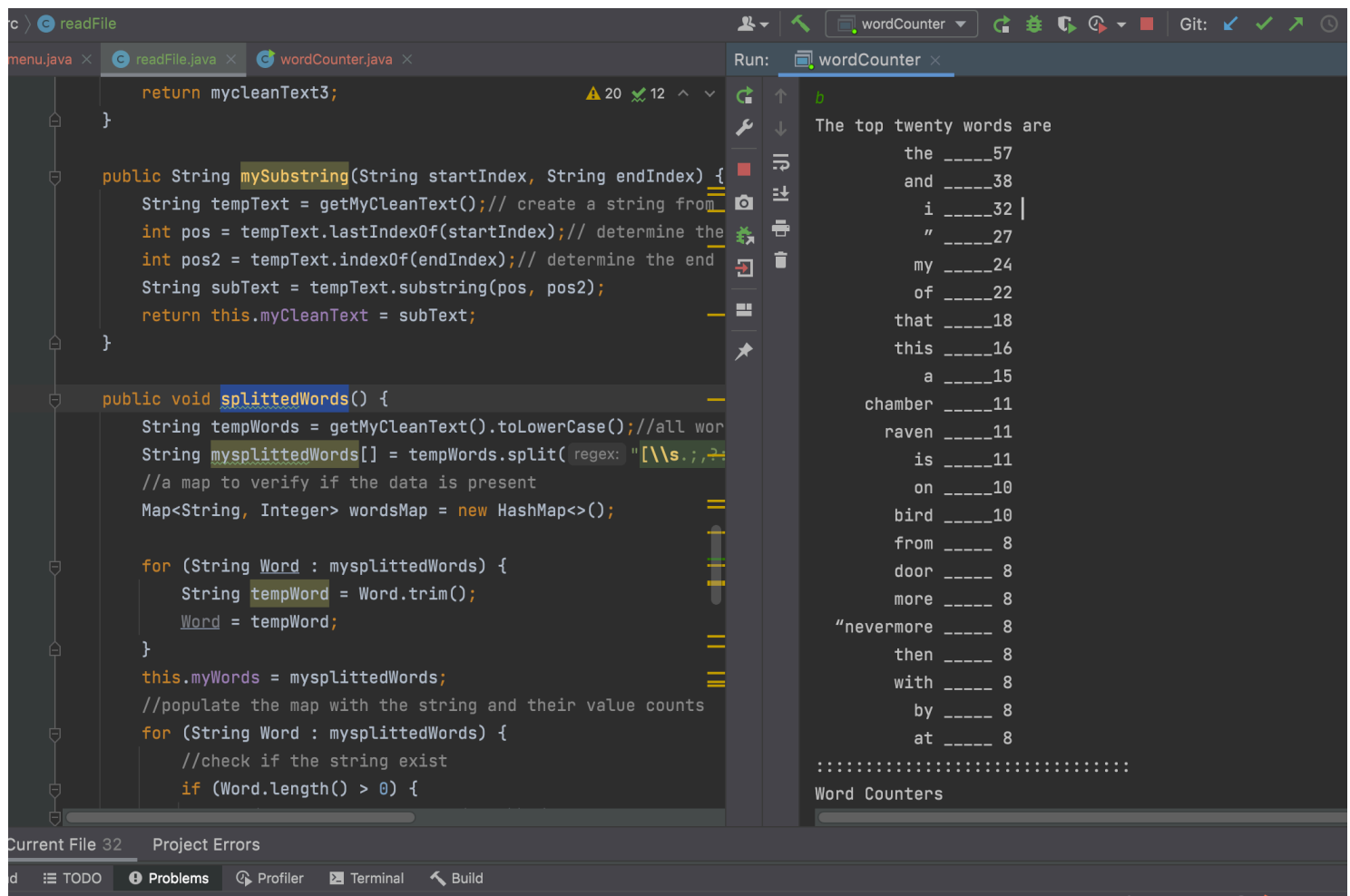
Gregory Lauture
CEN 3024C
September 22, 2021

The premise of the program is to load a file, in the requested case an html file from an url, skip the html tags, then identify the word in the file. Next step is ti have some functions that help create a list of the words and have this list sorted by descending count values. Finally to print the 20 most encountered words. The output is requested to be sorted in pairs.

My solution was first to have a function to read the html file from a web address. The result is then saved in a string variable origText. Another function is applied to the text to return a string myCleanText with all html tags and extra space removed. This is followed by a function to look for the words splittedWords, with two major use, first to save all the words encountered in a map, next to use the map to populate and Arraylist of pair with the words with its corresponding total encounters. Final results are saved into two Arrayslist, one with the total descending count, the other with the descending top twenty count. I made a supplement check to include any word who has the same encountered value as the last member of the top twenty to include potential runoff. Finally a function to print the list of count for all words, or for the top twenty only. Base features are included in the readFile class, there is also a menu file and the wordCounter to call the menu.

I join the GitHub link https://github.com/gnl2021/gnlprojects/tree/main/wordsCount/src

On the next page is the screen shots for the top twenty words encountered for the file provided The Raven.

Thanks for reading my document.

```java
        return mycleanText3;
    }

    public String mySubstring(String startIndex, String endIndex) {
        String tempText = getMyCleanText();// create a string from
        int pos = tempText.lastIndexOf(startIndex);// determine the
        int pos2 = tempText.indexOf(endIndex);// determine the end
        String subText = tempText.substring(pos, pos2);
        return this.myCleanText = subText;
    }

    public void splittedWords() {
        String tempWords = getMyCleanText().toLowerCase();//all wor
        String mysplittedWords[] = tempWords.split( regex: "[\\s.;,
        //a map to verify if the data is present
        Map<String, Integer> wordsMap = new HashMap<>();

        for (String Word : mysplittedWords) {
            String tempWord = Word.trim();
            Word = tempWord;
        }
        this.myWords = mysplittedWords;
        //populate the map with the string and their value counts
        for (String Word : mysplittedWords) {
            //check if the string exist
            if (Word.length() > 0) {
```

Run: wordCounter ×

```
b
The top twenty words are
         the _____57
         and _____38
           i _____32 |
         ”  _____27
          my _____24
          of _____22
        that _____18
        this _____16
           a _____15
     chamber _____11
       raven _____11
          is _____11
          on _____10
        bird _____10
        from _____ 8
        door _____ 8
        more _____ 8
  “nevermore _____ 8
        then _____ 8
        with _____ 8
          by _____ 8
          at _____ 8
::::::::::::::::::::::::::::::::
Word Counters
```

Current File 32   Project Errors

≡ TODO   ● Problems   Profiler   Terminal   Build

wordCounter   3