

Assignment: WordCount

Course: CSE2130 - File Structures and Exception Handling

Error - Reading lines and putting them into an array list would put a bunch of blank spaces into the array list.



```
4      "o", "p", "q", "r", "s", "t",
5      "u", "v", "w", "x", "y", "z" });
6
7      String word = "";
8
9      ArrayList<String> words = new ArrayList<String>();
10
11      ArrayList text = new ArrayList();
12
13      //USE ARRAY LIST
14
15      //Create the file
16      File textFile = new File("../Chapter11/src/SkillBuilders/instructions.txt");
17
18      try
19      {
20          //
21          in = new FileReader(textFile);
22          readfile = new BufferedReader(in);
23
24          while ((lineText = readfile.readLine()) != null)
25          {
26              for(int i = 0; i < 26 ; i++)
27              {
28                  if (lineText.contains(alphabet[i]))
29                  {
30                      word += lineText;
31                  }
32                  else
33                  {
34                      words.add(word);
35                      word = "";
36                  }
37              }
38          }
39      }
40      catch (IOException e)
41      {
42          e.printStackTrace();
43      }
44      readfile.close();
45  }
```

Problems @ Javadoc Console X Git Staging

minated> WordCount [Java Application] C:\Program Files\Eclipse\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw.exe (Oct 8, 2024 2:17:21 PM - 2:17:22 PM) [pid: 16900]

....., Create a Assignment application that reads and then displays the contents of a file containing

Error - Character and String issue, couldn't do methods with the two types together.

```

String[] alphabet = {"a", "b", "c", "d", "e", "f", "g",
                    "h", "i", "j", "k", "l", "m", "n",
                    "o", "p", "q", "r", "s", "t",
                    "u", "v", "w", "x", "y", "z"};

String word = "";

ArrayList<String> words = new ArrayList<String>();

ArrayList text = new ArrayList();

//USE ARRAY LIST

//Create the file
File textFile = new File("../Chapter11/src/SkillBuilders/instructions.txt");

try
{
    //
    in = new FileReader(textFile);
    readFile = new BufferedReader(in);

    while ((lineText = readFile.readLine()) != null)
    {
        for(int i = 0; i < 26 ; i++)
        {
            for(int y = 0; y < lineText.length(); y++)
            {
                if (lineText.charAt(y).contains(alphabet[i])) // char and string issue
                {
                    word += lineText; // Its reading the whole line have to make it read only characters
                }
                else
                {
                    words.add(word);
                    word = "";
                }
            }
        }
    }
}

```

Solution to the above errors - Make the file reader read line then read each character of each line. **Error** - This led to more errors since it was separated by letters and you couldn't define the words so I got rid of it.

The screenshot shows the Eclipse IDE with several tabs open: MyFile_P1.java, MyFile_P2.java, TestFile, Assignment.java, instruction..., and WordCount.java. The WordCount.java tab is active, displaying the following Java code:

```
36 in = new FileReader(textFile);
37 readFile = new BufferedReader(in);
38
39 while ((line = readFile.readLine()) != null)
40 {
41     for(int i = 0; i < line.length(); i++)
42     {
43
44         for(char a : alphabet) {
45
46             if(line.charAt(i) == a)
47             {
48
49                 word += line.charAt(i);
50                 System.out.println(word);
51
52             }
53             else
54             {
55
56                 words.add(word);
57                 word = "";
58
59             }
60
61         }
62
63     }
64 }
```

The console output at the bottom shows the words "b", "o", "b", "w", "e", "n", "t", "o", "n", "a", "w", "a", "l", "k", "h", "e", "w", "e", "n", "t", "t", "o", "t", "h", "e", "p".

Solution (Final Product) - Fixes new line spacing issue with the replace method, gets rid of null, and correctly splits the string into words based off of an array of non characters.

```

23
24 //Create the file
25 File textFile = new File("../Chapter11/src/Mastery.
26
27 try
28 {
29
30     in = new FileReader(textFile);
31     readFile = new BufferedReader(in);
32
33     while ((line = readFile.readLine()) != null)
34     {
35
36         lines += line;
37         lines.replace('\n', ' ');
38         lines += " " + readFile.readLine();
39
40     }
41
42     lines = lines.replace("null", " ");
43
44
45     String[] words = lines.split(nonChars);
46
47     for (String s : words)
48     {
49

```

Error - Output of number of words is incorrect. It's counting only the words of the second file line.

```
18 //Declaration
19 FileReader in;
20 BufferedReader readFile;
21 double wordLength = 0, avgLength = 0, numWords = 0;
22 String line = "", word = "";
23
24 String nonChars = "['1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '-', '_', '=', '
25
26 //Create the file
27 File textFile = new File("../Chapter11/src/Mastery/findreplace.txt");
28
29 try
30 {
31
32     in = new FileReader(textFile);
33     readFile = new BufferedReader(in);
34
35     while ((line = readFile.readLine()) != null)
36     {
37
38         String[] words = line.split(nonChars);
39
40         for (String s : words)
41         {
42             System.out.println(s);
43
44             wordLength += s.length();
45
46         }
47
48         numWords = words.length;
49
50     }
51     avgLength = wordLength / numWords;
52
53     System.out.println("The amount of words is: " + numWords);
54     System.out.println("The average length of the words is: " + wordLength / numWords);
55     readFile.close();
56     in.close();
57 }
```

Problems @ Javadoc Console X Git Staging

<terminated> WordCount [Java Application] C:\Program Files\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32

bob
went
on
a
walk
he
went
to
the
park
The amount of words is: 5.0
The average length of the words is: 5.8

Solution - Introduced a second variable to add to the first line and continue reading the rest of the file.txt, line and lines are the variable names

Error - Number of words is wrong and the lines read is wrong, it's reading the first line of the file twice.

```
30 {
31
32     in = new FileReader(textFile);
33     readFile = new BufferedReader(in);
34
35     while ((line = readFile.readLine()) != null)
36     {
37
38         lines += line;
39
40         String[] words = lines.split(nonChars);
41
42         for (String s : words)
43         {
44             System.out.println(s);
45
46             wordLength += s.length();
47
48
49         }
50
51         numWords = words.length;
52
53     }
54     avgLength = wordLength / numWords;
55
56     System.out.println("The amount of words is: " + numWords);
57     System.out.println("The average length of the words is: " + wordLength / numWords);
58     readFile.close();
59     in.close();
60 }
```

Problems @ Javadoc Console X Git Staging

<terminated> WordCount [Java Application] C:\Program Files\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe

bob
went
on
a
walk
bob
went
on
a
walkhe
went
to
the
park
The amount of words is: 9.0
The average length of the words is: 4.777777777777778

Solution - Had to make the string in the while loop and separate it outside of the loop.

```
23
24 //Create the file
25 File textFile = new File("../Chapter11/src/Mastery.
26
27 try
28 {
29
30     in = new FileReader(textFile);
31     readFile = new BufferedReader(in);
32
33     while ((line = readFile.readLine()) != null)
34     {
35
36         lines += line;
37         lines.replace('\n', ' ');
38         lines += " " + readFile.readLine();
39
40     }
41
42     lines = lines.replace("null", " ");
43
44
45     String[] words = lines.split(nonChars);
46
47     for (String s : words)
48     {
49
```

Error - Does not separate words from the newLine

```

32 in = new FileReader(textFile);
33 readFile = new BufferedReader(in);
34
35 while ((line = readFile.readLine()) != null)
36 {
37     lines += line;
38
39     String[] words = lines.split(nonChars);
40
41     for (String s : words)
42     {
43         System.out.println(s);
44
45         wordLength += s.length();
46
47     }
48
49     numWords = words.length;
50
51 }
52
53 length = wordLength / numWords;
54
55 em.out.println("The amount of words is: " + numWords);
56 em.out.println("The average length of the words is: " + wordLength / numWords);

```

Problems @ Javadoc Console X Git Staging

<terminated> WordCount [Java Application] C:\Program Files\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hot

went
on
a
walk
bob
went
on
a
walkhe
went
to
the
park
The amount of words is: 9.0
The average length of the words is: 4.777777777777778

New line

Solution - Every time the file reader read a new line it would add a space to the string to separate the new lines. I also used the replace method to replace /n with ' '


```

23
24 String nonChars = "[ '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '-', '_', '=', '+', '[', '{',
25
26 //Create the file
27 File textFile = new File("../Chapter11/src/Mastery/findreplace.txt");
28
29 try
30 {
31
32     in = new FileReader(textFile);
33     readFile = new BufferedReader(in);
34
35     while ((line = readFile.readLine()) != null)
36     {
37
38         lines += " " + readFile.readLine();
39         //lines = lines.replace('\n', ' ');
40
41         String[] words = lines.split(nonChars);
42
43         for (String s : words)
44         {
45
46
47
48             System.out.println(s);
49             numWords += 1;
50             wordLength += s.length();
51
52         }
53     }
54 }
55
56
57
58
59 avgLength = wordLength / numWords;
60
61 System.out.println("The amount of words is: " + numWords);
62 System.out.println("The average length of the words is: " + wordLength / numWords);
63 System.out.println(wordLength);
64 readFile.close();
65 in.close();
66
67 //Catching errors
68 }

```

```

23
24 //Create the file
25 File textFile = new File("../Chapter11/src/Mastery.
26
27 try
28 {
29
30     in = new FileReader(textFile);
31     readFile = new BufferedReader(in);
32
33     while ((line = readFile.readLine()) != null)
34     {
35
36         lines += line;
37         lines.replace('\n', ' ');
38         lines += " " + readFile.readLine();
39
40     }
41
42     lines = lines.replace("null", " ");
43
44
45     String[] words = lines.split(nonChars);
46
47     for (String s : words)
48     {
49

```

Error - Ugly decimal output

```

bob went to the mall he saw bill bill had a cat named c o l l e r
The amount of words is: 19.0
The average length of the words is: 2.473684210526316

```

```

words is: " + numWords);
length of the words is: " + wordLength / numWords);

```

Solution - Decimal Format

```

bob went to the mall he saw bill bill had a cat named c o l l e r
The amount of words is: 19.0
The average length of the words is: 2.47

```

```

is: " + numWords);
of the words is: " + DecimalFormat.format(wordLength / numWords));

```