

Alex Eschenauer
Program 2 (WordAnalyzer) Bug Log

BUGS:

1. Bug 1: `getWord()` not returning word
 - a. JUnitTest: `testSettingAndGettingWord()`
 - b. Location of Bug in Code: Return Statement of `getWord()` method
 - c. Description: it returns null, instead of returning the word.
 - d. How you fixed it (If Applicable): I changed null to word
2. Bug 2: `firstMultipleCharacter()` returning wrong character
 - a. JUnitTest: `testFirstMultipleCharacter()`
 - b. Location of Bug in Code: `firstMultipleCharacter()` if-statement
 - c. Description: It said that if the result was greater than or equal to zero it would work, which means no matter what it is always the first character.
 - d. How you fixed it (If Applicable): I deleted the equal symbol and kept it as greater than.
3. Bug 3: `RepeatedCharacterGroupCount()` bug
 - a. JUnitTest: `testCharacterGroupCount()`
 - b. Location of Bug in Code: `RepeatedCharacterGroupCount()` method
 - c. Description: If you have a word such as "aaaaaaaaaa" with no other letters, it should still be one group and it returns zero instead.
 - d. How you fixed it (If Applicable): I added three things, all of which are commented in the code. The first was a count variable, which counts how many times something is repeated. If count is equal to the length of the word, then result adds one. That fixes it.
4. Bug 4: `Palindrome()` only outputting true
 - a. JUnitTest: `testPalindrome()`
 - b. Location of Bug in Code: `Palindrome()` Method
 - c. Description: It only outputs true, no matter what.
 - d. How you fixed it (If Applicable): I added in code so that it can tell if it is not a palindrome, and return false accordingly. There was not even the word false anywhere in the method before hand.
5. Bug 5: `firstRepeatedCharacter()` not working if there are no repeated characters
 - a. JUnitTest: `testFirstRepeatedCharacter()`
 - b. Location of Bug in Code: `firstRepeatedCharacter()` Method
 - c. Description: It would never display anything outside of the for loop. So that means it would never return result if there were no repeated characters.
 - d. How you fixed it (If Applicable): I fixed it by adding a counter named count, set to integer 1. I then put `count++` in the for loop to allow it to add up. Finally I added an if statement saying that if count equals `word.length()`, then it would return 0. I tested it and it now works for both scenarios.

And that is all the bugs I could find. I'm sure there are probably more, but at least I found one for each method.