* First check the length of the string line. If it is not zero, word count and sentences count is increased by one.
  + Because word count is calculated by using spaces. So, total number of words are equal to total number of spaces +1
  + In this code last character of not be executed. Because of that, if string length is not zero, the sentence count is by one, before counting sentences.
* As well as check if last element is space or not. If it is a space the last element is removed.
* By checking that word count can be calculated correctly (According to my logic).
  + Assume that space is done by mistake. There is only one space in the end points.
* Then, convert string to char array and, after that check each element in the array by using for loop.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* When counting spaces, it always checks, if next element is space or not. If it is space, the word count (numberOfWords) variable is not increased. If next element is not space word count (numberOfWords) variable is increased.
* By checking this next element multi spaces can be neglected. If element is space and next is also space, then word count (numberOfWords) variable is not increased until next element is not space.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* After that number of sentences are counted by using full stop, question mark, exclamation mark and semi colon.
  + Assumed sentences are end only using those functions.
* As well as it should be a space after that function mark. Otherwise, the sentence count (numberOfSentences variable) is not increased.
  + Because of otherwise url and some other special cases are considered as one sentence.
* Let’s imagine there is name with initials, in such cases there are “.” Marks in those places. But code is not recognized it is not a name. So, I check second previous number and if it is not a space then sentence count (numberOfSentences variable) is increased.
  + Assumed there is not ending words with one character.
  + If first sentence is started name with initials there should be an exception when checking second previous number. To avoid that index of previous second large number should be positive or zero. Otherwise sentence count (numberOfSentences variable) not increased.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* The other requirement is to calculate the phrases in the text document. To do that count all double quotations in documents (In here opening double quotation and closing double quotation are same).
  + Then number of said phrases are equal to half from total number of Phrases variable value (total number of double quotations).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* I assumed paragraphs are considered as more than one sentence. Because otherwise, address lines, dates, topics are calculated as paragraphs.
  + So, variable is created as temp and that temp variable is equal to number of sentences in the line.
  + Then created a logic, if there are one more sentence in the line, that line can be considered as paragraph. (temp variable should be more than one)
* Finally, those total word count is assign to 1st of the return array, 2nd element represents the number of paragraphs, 3rd element represents the number of sentences.
* Number of phrases are equal to half of double quotation number. So, 4th element represents the number of phrases is in the document.
* As well as it is needed to find average number of words in sentences. It is assigned as the 5th element of the array.
  + To do that total number of words should be divided by total number of sentences.
  + In this code if 1st line is empty, null point exception is thrown. To avoid that, that condition is written inside if condition.
* Finally, array is returned.
* To do that I used only basic concept of java. But there are modern features are in java like, random access file method, nio library and using regular expression library (regex). Using those methods, it can be done easily.