**Technical Design Document**

**Name:** Chauncy Kent

**Date Created:** 10/2/2025

**Program Description:**

Contains the ParagraphAnalyzer class and the main function. The main function creates an instance of the ParagraphAnalyzer class and calls the appropriate methods in the correct order to save a paragraph from the user, where each sentence starts with a number, divide it into a list of sentences, count the number of sentences, and display each identified sentence as well as the total number of sentences counted.

**Functions used in the Program:**

1. **Function Name:** main

**Description:** Creates a ParagraphAnalyzer object and calls the paragraph collection, analysis, and display methods in the correct order.

**Parameters:** none

**Variables:** none

**Logical Steps:**

1. Creates a ParagraphAnalyzer object.
2. Calls the get\_paragraph method.
3. Calls the analyze\_paragraph method.
4. Calls the count\_sentences method.
5. Calls the display\_results method.

**Returns:** none

2. **Function Name:** get\_paragraph

**Description:** Prompts the user for a paragraph where each sentence begins with a number and saves it to the self.paragraph attribute.

**Parameters:** self

**Variables:** self.paragraph

**Logical Steps:**

1. Prompts the user for a paragraph where each sentence starts with a number.
2. Saves it to self.paragraph.

**Returns:** none

3. **Function Name:** analyze\_paragraph

**Description:** Defines a pattern that recognizes a sentence as starting with a number and ending with a '.', '!', or '?' followed by either a space and a number, or the end of the string. Then searches the paragraph to find all qualifying substrings and adds them to a list, saving the list as the value of self.sentence\_list.

**Parameters:** self

**Variables:** pattern, self.sentence\_list, self.paragraph

**Logical Steps:**

1. Defines the pattern that identifies what the findall method is looking for.
2. Calls re.findall using pattern to find all the sentences in the paragraph.
3. Assigns the created list as the value of self.sentence\_list.

**Returns:** none

4. **Function Name:** count\_sentences

**Description:** Finds the length of self.sentence\_list and assigns it as the value of self.sentence\_count.

**Parameters:** self

**Variables:** self.sentence\_count, self.sentence\_list

**Logical Steps:**

1. Finds the length of self.sentence\_list.
2. Assigns it as the value of self.sentence\_count.

**Returns:** none

5. **Function Name:** display\_results

**Description:** Calls a helper method to 'clean' the formatting of the sentences in self.sentence\_list then loops through the list, printing each sentence. Finally, prints a blank line followed by the total number of sentences.

**Parameters:** self

**Variables:** self.sentence\_list, self.sentence\_count

**Logical Steps:**

1. Calls the \_clean\_sentences method.
2. Prints a blank line.
3. Prints “The sentences in the paragraph are:”.
4. Prints each sentence in self.sentence\_list.
5. Prints a blank line.
6. Prints “The number of sentences in the paragraph is:”.
7. Prints the value of self.sentence\_count.

**Returns:** none

6. **Function Name:** \_clean\_sentences

**Description:** Loops through the sentences in self.sentence\_list and 'cleans' them, removing extra white space, new line characters, and other 'junk'.

**Parameters:** self

**Variables:** self.sentence\_list

**Logical Steps:**

1. Loops through each sentence in self.sentence\_list.
2. Calls .strip on each sentence.
3. Assigns the ‘clean’ sentence to the same index position in self.sentence\_list.

**Returns:** none

**Logical Steps:**

1. main
2. get\_paragraph
3. analyze\_paragraph
4. count\_sentences
5. display\_results
6. \_clean\_sentences

**Link to your repository:** <https://github.com/ChauncyJKent/COP2373/activity>

**Output Screenshot:**

Input and output:

A black background with many small colored lines

AI-generated content may be incorrect.

Just output (for size):

A screen shot of a computer screen

AI-generated content may be incorrect.