Design Change Report

Xi Tang

@01887891

1. DRY (Don’t repeat yourself)

(1) In order to avoid duplicate code. I wrote a config.py to store them as public final constants.

文本

描述已自动生成

congif.py 1

(2) Different tasks entail code to decide whether a string is a word. Thus, I wrote a separate method: is\_word(short\_seqs):

文本

描述已自动生成文本

描述已自动生成文本

描述已自动生成

By the same token, I wrote has\_seperators(text) to decide if a string contains seperators, find\_sep\_index(text) to find the index of the separator in a string, word\_split(text) to find all the words in a text, and text\_break(text) to make all the words and separators elements of an array. I can reuse these methods if new requirements need.

2. Delegation Principles

The above methods also satisfy the Delegation Principles.

3. Single Responsibility Principle (SRP)

Every method written has only one functionality.

4. Design pattern

This project doesn’t apply design patterns.

5. Design Change

For Sprint 2, I added the functionality to count lines. For this one, since this functionality doesn’t use previous code or similar code to previous ones, I didn’t change my previous code. What I did is adding method has\_new\_line(text), count\_new\_line(text) and line\_count(text).

For Sprint 3, I added the functionality to replace words with target words. I didn’t change my previous code. What I did is adding method text\_break(text), word\_to\_string(arr) and word\_replace(text). Though text\_break(text) is similar to word\_split(text), it’s hard to merge these two methods. For word\_split(text), we only put words in the array that we return. However, for text\_break(text), we put both words and separators sequentially in the array that we return. We would write a method to remove the separators from text\_break(text)’ return values if we plan to keep text\_break(text) and discard word\_split(text).