PLT - Pig Latin Translator

Goal

Pig Latin is a language game in which English words are altered according to some rules. PLT (Pig Latin Translator) is an API to translate an English phrase into the Pig-Latin language.

So far, you do not need to know more. Further details will be given in the User Stories section.

Instructions

READ the instructions carefully.

You are asked to **DEVELOP** PLT by following **TDD**.

You **DO NOT** need to develop a GUI.

The PLT requirements are divided into a set of **USER STORIES**, which serve as a to-do list.

You should be able to incrementally develop PLT without an upfront comprehension of all the PLT requirements. **DO NOT** read ahead, and handle the requirements (i.e., specified in the user stories) one at a time in the provided order. Develop PLT by starting from the first story's requirement. When a story is **IMPLEMENTED**, move on to the **NEXT** one. A story is implemented when you are confident that your program correctly implements all the functionality stipulated by the story's requirement. This implies that all your test cases for that story and all the test cases for the previous stories pass. You may need to review your program as you progress towards more advanced requirements.

If you need to handle error situations (including situations unspecified by the user stories), throw a **`PigLatinException`**.

User Stories

Remember to read and implement the user story once at a time (in the provided order). Therefore, do not read the next user story, if the current one is not implemented yet.

Each time you see the JUnit green bar (i.e., end of the **GREEN** phase and end of the **REFACTOR** one), **COMMIT**.

1 - Input phrase

A translator (from English to Pig Latin) takes a phrase as the input. The phrase is represented as a string where words are separated by white spaces.

Requirement: The translator that takes a phrase as the input. Let the *client* get the input phrase from the translator.

Example: The translater takes as the input the phrase *"hello world".* The *client* gets the input phrase: *"hello world"*.

2 - Translating an empty phrase

The translator translates an input phrase into Pig Latin. The input phrase can be an empty string. When this happens, the result of the translation is *"nil"*.

Requirement: Let the translator translate an empty string.

Example: The translation of an empty string is *"nil"*.

3 - Translating a word starting with a vowel

The input phrase can be a single word starting with a vowel. In that case, the translator applies to following translation rules:

a. If the word ends with y, append *"nay"* to the and of the word.
b. If the word ends with a vowel, append *"yay"* to the and of the word.
c. If the word ends with a consonant, append *"ay"* to the and of the word.

Requirement: Let the translator translate a word starting with a vowel.

Example: The translation of "*any*" is *"anynay"* (a). The translation of "*apple*" is *"appleyay"* (b). The translation of "*ask*" is *"askay"* (c).

4 - Translating a word starting with a single consonant

The input phrase can be a single word starting with a single consonant (nota that the *y* letter is considered a consonant). In that case, the translator applies to following translation rule:

- Remove the consonant from the beginning of the word and add it to the end of the word. Finally, append *"ay"* to the end of the resulting word.

Requirement: Let the translator translate a word starting with a single consonant.

Example: The translation of "*hello*" is *"ellohay"*.

5 - Translating a word starting with more consonants

The input phrase can be a single word starting with more consonants. In that case, the translator applies to following translation rule:

- Remove the consonants from the beginning of the word and add them to the end of the word. Finally, append *"ay"* to the end of the resulting word.

Requirement: Let the translator translate a word starting with more consonants.

Example: The translation of *"known"* is *"ownknay"*.

6 - Translating a phrase containing more words

The input phrase can contain more words (separated by white spaces). In that case, the translator applies the translation rules (reported in User Stories 3-5) to the single words. Moreover, for the composite words (those separated by a *"-"*), the translation rules apply to the single words.

Requirement: Let the translator translate a phrase containing more words, as well as composite words.

Example: The translation of *"hello world"* is *"ellohay orldway"*. The translation of "*well-being"* is *"ellway-eingbay"*.

7 - Translating a phrase containing punctuations

The input phrase can contain punctuations. In that case, the translator applies the translation rules to the single and composite words while preserving the punctuations. Only the following punctuation characters are allowed: point, comma, semi-colon, colon, question mark, exclamation mark, apostrophe, and round parenthesis.

Requirement: Let the translator translate a phrase containing punctuations.
Example: The translation of *"hello world!"* is *"ellohay orldway!"*.

8 - Translating a phrase with upper- and title-case words (this story is optional)

The input phrase can contain upper- and title-case words. In those cases, the translator applies the translation rules to the single and composite words while preserving the upper- and title-cases. Cases different from upper- and title-cases are not allowed (e.g., "biRd").

Requirement: Let the translator translate a phrase containing upper- and title-case words.

Example: The translation of *"APPLE"* (upper case) is *"APPLEYAY"*. The translation of *"Hello"* (title case) is *"Ellohay"*.