Project 2: Pig Latin Translator

Collaboration Policy: This assignment must be completed individually. You may discuss high-level ideas with other students, but you should not share code.

Grading

- 40 points basic translation (20 points each way)
- 30 points for the JUnit Tests
- 20 points for presentation and error handling
- 10 points code readability and organization

Description

The goal of this homework assignment is to create a pig latin translator.

To translate a word to pig latin, you do the following:

- If the word starts with a vowel or the letter y, you add the letters ‘ay’ to the end of the word.
- If the word starts with any other letter, then move all of the letters up to the first vowel to the end of the word and add ‘ay’ at the end.

To make reverse translation possible, we will add an ‘ before the new or moved letters.

Examples:

- eggs become eggs’ay
- yay becomes yay’ay
- word becomes ord’way
- the becomes e’thay

We will translate a sentences from a file either to or from pig latin. If we are translating from Pig Latin to English and there is no ‘ in the word, then we must tell them that that we cannot translate their sentence. You will make your own TranslationException class for this error.

Example: Mary had a little lamb?! <=> Ary’may ad’hay a’ay ittle’lay amb’lay?!

Regular Expressions

Regular expression syntax, or regex as it is often abbreviated, is a pattern matching syntax that is used across several programming languages. Many problems validating or parsing strings can be simplified through the use of regex pattern matching. While this project can be done without the use of regex patterns, you may find that taking some time to learn it will greatly simplify your coding.

You can find an online regex tester with some basic reference information at http://regexr.com/
There are also some tutorial videos based on using this site on YouTube at
https://www.youtube.com/watch?v=GVZOJ1rEnUg&list=PLfdtiltiRHWGRPyPMGuLPWuiWgEI9Kp1w

PigLatinTranslator

All of your translation code should be contained in the edu.htc.piglatin.PigLatinTranslator class.

This class must contain the following four methods:

- public static String translateToPigLatin(String sentence)
- public static String translateFromPigLatin(String sentence)
- protected static String wordToPigLatin(String word)
- protected static String wordFromPigLatin(String word)

You may also create additional helper methods in this class as needed.

Note that if asked to translate an empty or blank String, the above methods should just return back the input value unaltered. Make sure to code and test for this rule.

JUnit Testing

You must include JUnit tests for your PigLatinTranslator. All test classes should be kept in a test sources directory that is separate from your other classes. Remember that if you are having trouble running the tests in IntelliJ to check your project setup, specifically your output paths. You can also try to close the project, delete the IntelliJ files and make a new project from existing sources.

Your grade for the testing piece will be partially based on how thoroughly you test this code. Make sure to test with capital and lowercase letters, different types of punctuation and unexpected values. You may find the code coverage tools in IntelliJ helpful for this.

Main Program

Once you have built your translator class, you will need to demonstrate its use by translating files. The files for translation will contain one sentence per line, and should be written out in a similar fashion.

1. Read in the sampleText.txt file from the data package and write out a sampleText_PigLatin.txt file that contains the text translated into PigLatin.
2. Read in the samplePigLatin.txt file from the data package and write out a samplePigLatin_English.txt file that contains the text translated into English.

Note that you may use the classes built in the prior homework for reading and writing the data to/from a file. You are encouraged but not required to JUnit test these classes.

The Main class should contain only a short main method that uses the File I/O and PigLatinTranslator classes to do the actual work. The code organization and readability points will be influenced by how well you organize your classes and focus your methods on doing one small thing.