**Project 2 – Pig Latin**

**(20 points)**

**Name: Ozaner Hansha**

**Due Date: Thursday, October 8, 2015**

**Description:**

We have been refreshing our understanding of the Java language, its common data types, and the correct vocabulary for described various concepts and syntactic features of the language. We have reviewed the Java API documentation, the Eclipse IDE, etc.

Here is your chance to develop a set of fairly simple classes that progressively develop a Pig Latin translator. This will reacquaint you with the methods in the String class. Strings are incredibly useful in most applications so it is important that you gain mastery of them.

Carry out the following steps:

1. Create an Eclipse project for your PigLatin project.
2. Create a class called PigLatin that contains a main method containing a simple read-eval-print loop that prompts for and reads a word, translates it into Pig Latin using the rules in the appendix below, and then prints out the translation. The loop (and program) should terminate on an empty line input.
3. To ensure some modularity, use a method with the following declaration:  
    public static String translate(String word)

This will be the method that your main program uses to get the translation. You will be making heavy use of the String class methods as you implement translate.

1. Create another method with the declaration:  
    public static String speak()

* This method produces a random phrase for what a pig says. One obvious choice is “oink” or “oink oink”, but you can be creative. You should have at least a dozen different phrases. Your main program will append what the pig says as a “comment” on the end of your translation of the given word or phrase.   
    
  If you are interested in what people in other cultures think a pig says, here are a couple of interesting web sites: <http://www.eleceng.adelaide.edu.au/Personal/dabbott/animal.html>  
  [http://www.buzzfeed.com/robinedds/what-noises-do-animals-make-in-other-languages-here-is-an-im#4cnut4g](http://www.buzzfeed.com/robinedds/what-noises-do-animals-make-in-other-languages-here-is-an-im%234cnut4g)

1. Here is a sample session (**Porky** is the name of our pig in this example). Note that when the empty line is entered Porky announces “oodbyegay” (translation of “goodbye”) and terminates.

**This program translates words into Pig Latin.  
  
Next input > computer  
Porky says > omputercay // oink oink  
  
Next input > chatham  
Porky says > athamchay // … more random pig sayings …  
  
Next input > question  
Porky says > estionquay // … more random pig sayings …  
  
Next input > apple  
Porky says > appleyay // … more random pig sayings …  
  
Next input >**

**Porky says > oodbyegay**

1. Use good programming style to make your program readable, use constants where appropriate, use private helper methods if useful, provide Javadoc comments for every public item, etc.
2. Submission: Copy and paste your console output and your Java source code listings to the end of this document. Upload that document to the Hand-In folder that you created and shared with me. Be careful when copying and pasting to preserve the font information from Eclipse. It makes reading your code **much** easier.
3. Save this document as “Project 2 – Pig Latin (*Lastname Firstname*).doc”
4. Fill out your name at the top of this page.
5. Append your program and console output where indicated at the end of this document.
6. Run your program on the following inputs:  
   pig, banana, trash, stream, squash, happy, duck, glove, quack, eat, omelet, are, honesty, hourly, at, a, p, h
7. Upload (drag) your project document into the Google Drive folder that you shared with me.

**Project Extensions**

For those who complete the project quickly, here are some extensions to implement:

1. ✔Generalize your program to take a phrase or sentence rather than a single word on each line. You can assume that a sentence is simply a sequence of space-delimited words.
2. ✔Handle capitalization correctly. If the original word was capitalized, then capitalize the corresponding Pig Latin word.
3. Handle punctuation in your input. (Preserve it in the translation.)
4. (Hard) Handle the Pig Latin pronunciation of numbers.  
   Examples:
   * + "43" → "ortythreefay"
     + "1" → "oneway"

**Appendix**

To translate a word into Pig Latin,

* For words that begin with an initial string of consonants, i.e., one or more occurrences of b, c, d, f, g, h, j, k, m, n, p, qu, r, s, t, v, w, x, y or z: Call this sequence *prefix* and the remainder of the string *suffix*. Note that prefix may be the empty string if the word begins with a vowel. The Pig Latin translation is formed by *suffix + prefix + "ay"*.

Examples:

* + - "pig" → "igpay"
    - "banana" → "ananabay"
    - "trash" → "ashtray"
    - "happy" → "appyhay"
    - "duck" → "uckday"
    - "glove" → "oveglay"
    - "quack" → "ackquay"
* For words that begin with vowel sounds, just add *“yay”* to the end. In English, words of French origin sometimes begin with a silent ‘h’ (hour, honest, heir, honor) and hence begin with vowel sounds and fall under this rule. Include a number of these special case words and their related forms (honest/honesty/honestly).

Examples:

* + - "eat" → "eatyay"
    - "omelet" → "omeletyay"
    - "are" → "areyay"
    - "honest" → "onestyay"
* Single alphabet letters are pronounced like they sound, not how they are written. For example, the letter “f” is pronounced “ef” and therefore has an initial vowel sound. The pronunciation should be spelled out in the Pig Latin translation:

Examples:

* + - "f" → "efyay"
    - "t" → "eetay"
    - "o" → "ohyay"
    - "q" → "ookyay"

Here is a spelling of English letter names:  
A/a ay

B/b bee

C/c cee

D/d dee

E/e ee

F/f ef

G/g gee

H/h aitch

I/I eye

J/j jay

K/k kay

L/l el

M/m em

N/n en

O/o oh

P/p pee

Q/q kyoo

R/r ar

S/s ess

T/t tee

U/u you

V/v vee

W/w double-you

X/x ex

Y/y wy

Z/z zee

**Code for** PigLatin:

**package** **unit1**;  
  
**import** **java.util.ArrayList**;  
**import** **java.util.Arrays**;  
**import** **java.util.Scanner**;  
  
*/\*\**  
 *\* This program:<br>*  
 *\* -takes a word from the user,<br>*  
 *\* -translates it to Pig Latin, <br>*  
 *\* -then prints that translation to the console.<br><br>*  
 *\* The program also outputs occasional pig onomatopoeias to the console as well<br><br>*  
 *\**   
 *\* AP Computer Science<br>*  
 *\* Dr. Jones<br>*  
 *\* Class 7<br>*  
 *\* @author Ozaner Hansha*  
 *\*/*  
**public** **class** **EnglishToPigLatin**  
{  
 */\*\**  
 *\* A set of the lowercase English Alphabet in order.*  
 *\*/*  
 **public** **static** **final** Character[] ALPHABET =   
 {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};  
   
 */\*\**  
 *\* A set of strings representing the pronunciation of the English {@link #ALPHABET}.*  
 *\*/*  
 **public** **static** **final** String[] ALPHABET\_PRONUNCIATION =   
 {"ay","bee","cee","dee","ee","ef","gee","aitch","eye","jay","kay","el","em","en",  
 "oh","pee","kyoo","ar","ess","tee","you","vee","double-you","ex","wy","zee"};  
 */\*\**  
 *\* This is an array of the lowercase vowels of the English alphabet in char form.*  
 *\*/*  
 **public** **static** **final** Character[] VOWELS = {'a','e','i','o','u'};  
   
 */\*\**  
 *\* A list of words that begin with a silent H.*  
 *\*/*  
 **public** **static** **final** String[] SILENT\_H\_WORDS = {"honor", "honest", "honestly", "honesty", "heir", "hour", "hourly"};  
   
 */\*\**  
 *\* A set of strings of Pig onomatopoeias in several languages.*  
 *\*/*  
 **public** **static** **final** String[] PIG\_NOISES =   
 {"Oink!", "Buu!", "Hunk!", "Grunz..", "Nöff..", "Sweek!", "Quiek!", "¡Oinc!", "Ghrutu!", "Knor!", "Kwi!", "Röf!"};  
   
 */\*\**  
 *\* The {@link Scanner} used to gather input from the console to be evaluated.*  
 *\*/*  
 **private** **static** **final** Scanner IN = **new** Scanner(System.in);  
   
 */\*\**  
 *\* The phrase to end the program. It is "done" in Pig Latin.*  
 *\*/*  
 **private** **static** **final** String SENTINEL = "";  
   
 */\*\**  
 *\* An array of Strings that are currently being evaluated.*  
 *\*/*  
 **private** **static** ArrayList<String> currentWords = **new** ArrayList<String>();  
   
 */\*\**  
 *\* An array of boolean values that correspond the the capitalization of words in the {@link #currentWords} list.*  
 *\*/*  
 **private** **static** ArrayList<Boolean> capitalization = **new** ArrayList<Boolean>();  
   
 */\*\**  
 *\* Checks if a given String is a single character in the English {@link #ALPHABET}.<br>*  
 *\* Ex. j --> jay*  
 *\**   
 *\* @param word - The String to check.*  
 *\* @return true if the given String is a single character in the {@link #ALPHABET}.*  
 *\*/*  
 **public** **static** boolean isSingleLetter(String word)  
 {  
 **return** word.length() == 1 && Character.isLetter(word.charAt(0)); *//If word is 1 char long and contains a letter of the ALPHABET.*  
 }  
   
 */\*\**  
 *\* This method returns whether or not this word is pronounced with a vowel at the start*  
 *\**   
 *\* @param word - The word to check for a vowel sound.*  
 *\* @return True if this is a vowel sound word, false if not.*  
 *\*/*  
 **public** **static** boolean startsWithVowelSound(String word)  
 {  
 **return** Arrays.asList(VOWELS).contains(word.charAt(0)) || Arrays.asList(SILENT\_H\_WORDS).contains(word);  
 }  
   
 */\*\**  
 *\* This method returns the index of the first vowel in a given String.*  
 *\**   
 *\* @param word - The String to evaluate.*  
 *\* @return The index of the first vowel of the given String, -1 if no vowels.*  
 *\*/*  
 **public** **static** int indexOfFirstVowel(String word)  
 {  
 **for**(int i = 0; i < word.length(); i++)  
 {  
 **for**(int v = 0; v < VOWELS.length; v++)  
 {  
 **if**(word.charAt(i) == VOWELS[v])  
 **return** i;  
 }  
 }  
 **return** -1;  
 }  
   
 */\*\**  
 *\* Translated individual letters to Pig Latin.*  
 *\**   
 *\* @param word - A character in the {@link #ALPHABET} to convert to Pig Latin.*  
 *\* @return a translation of the given char to Pig Latin*  
 *\*/*  
 **public** **static** String translateLetter(String word, boolean capitalization)  
 {  
 **return** translate(ALPHABET\_PRONUNCIATION[Arrays.asList(ALPHABET).indexOf(word.charAt(0))], capitalization);  
 }  
   
 */\*\**  
 *\* This method translates a word to Pig Latin with the Vowel Sound rules.<br>*  
 *\* Ex. elephant --> elephantyay.*  
 *\**   
 *\* @param word - A word to be translated to Pig Latin with the Vowel Sound rules.*  
 *\* @param capitilized - Whether or not this word should be capitalized.*  
 *\* @return the given word translated to Pig Latin.*  
 *\*/*  
 **public** **static** String translateVowelSound(String word)  
 {  
 **return** word + "yay"; *//ABCD --> ABCDyay*  
 }  
   
 */\*\**  
 *\* This method translates a given word to Pig Latin.<br>*  
 *\* Ex. Hello --> ellohay*  
 *\**   
 *\* @param word - A word to be translated to Pig Latin.*  
 *\* @param capitilized - Wheather or not this word should be capitalized.*  
 *\* @return The Pig Latin translation of the given word.*  
 *\*/*  
 **public** **static** String translateNormal(String word)  
 {  
 *//for "qu" words.*  
 **if**(indexOfFirstVowel(word) <= word.indexOf("qu"))  
 {  
 **return** word.substring(word.indexOf("qu") + 2) + word.substring(0, word.indexOf("qu") + 2) + "ay";  
 }  
   
 *//For words with no vowels.*  
 **if**(indexOfFirstVowel(word) <= -1)  
 **return** word + "ay";  
   
 *//Non "qu" words.*  
 String prefix = word.substring(0, indexOfFirstVowel(word));  
 String suffix = word.substring(indexOfFirstVowel(word));  
 **return** suffix + prefix + "ay"; *//Dirty --> irtyday*  
 }  
   
 */\*\**  
 *\* This method returns a given word's Pig Latin translation.*  
 *\**   
 *\* @param word - A string to be translated to Pig Latin.*  
 *\* @param capitilized - Wheather or not this word should be capitalized.*  
 *\* @return The Pig Latin translation of the word given, returns null if String given is null.*  
 *\*/*  
 **public** **static** String translate(String word, boolean capitalized)  
 {  
 String tempWord = word;  
 **if**(word == **null**) *//If word is null.*  
 **return** **null**;  
 **else** **if**(isSingleLetter(word)) *//If word is just a single letter*  
 tempWord = translateLetter(word, capitalized);  
 **else** **if**(startsWithVowelSound(word)) *//If word starts with a vowel sound.*  
 tempWord = translateVowelSound(word);  
 **else** *//if word passes all other tests (A normal word).*  
 tempWord = translateNormal(word);  
   
 *//Checks for Capitalization*  
 **if**(capitalized)  
 tempWord = tempWord.substring(0, 1).toUpperCase() + tempWord.substring(1);  
 **return** tempWord;  
 }  
   
 */\*\**  
 *\* @return A random string from the {@link #PIG\_NOISES} array.*  
 *\*/*  
 **public** **static** String speak()  
 {  
 **return** PIG\_NOISES[(int)(Math.random()\*PIG\_NOISES.length)]; *//random int from 0 to Length of array (12)*  
 }  
   
 */\*\**  
 *\* This Program takes a word or sentence(Input),<br>*  
 *\* translates it to Pig Latin(Evaluate),<br>*  
 *\* then prints it to the console(Print).<br><br>*  
 *\**   
 *\* Program also prints out random items from {@link #PIG\_NOISES} every translation.*  
 *\**   
 *\* @param args - no command line arguments expected*  
 *\*/*  
 **public** **static** void main(String[] args)   
 {  
 System.out.println("This program translates words into Pig Latin.\n​"); *//Initialization Message.*  
   
 *//Print Loop*  
 **while**(**true**)  
 {  
 System.out.print("Next Input > ");  
 Scanner lineIn = **new** Scanner(IN.nextLine());  
   
 *//Gathers all words delimited by a space into currentWords*  
 **while**(lineIn.hasNext())  
 {  
 String tempWord = lineIn.next(); *//Stores this word for analysis.*  
 capitalization.add(Character.isUpperCase(tempWord.charAt(0))); *//Adds its capitalization status to list.*  
 currentWords.add(tempWord.toLowerCase()); *//Adds this word to list (in lower case)*  
 }  
 System.out.print("Porkey Says >");  
   
 *//Terminates Program via Sentinel*  
 **if**(currentWords.isEmpty())  
 {  
 lineIn.close();  
 IN.close();  
 System.out.println(" " + translate("goodbye", **true**) + "...");  
 System.exit(0);  
 }  
   
 *//Outputs all Strings in currentWords translated*  
 **for**(int w = 0; w < currentWords.size(); w++)  
 {  
 System.out.print(" " + translate(currentWords.get(w), capitalization.get(w)));  
 }  
 currentWords.clear();  
 capitalization.clear();  
 System.out.print(" // " + speak() + "\n\n");  
 }  
 }  
}

**Console Output**

This program translates words into Pig Latin.

Enter 'oneday' to quit.

Next Input > pig

Porkey Says > igpay // Nöff..

Next Input > banana

Porkey Says > ananabay // Hunk!

Next Input > trash

Porkey Says > ashtray // Quiek!

Next Input > stream

Porkey Says > eamstray // ¡Oinc!

Next Input > squash

Porkey Says > ashsquay // Kwi!

Next Input > happy

Porkey Says > appyhay // ¡Oinc!

Next Input > duck

Porkey Says > uckday // Oink!

Next Input > glove

Porkey Says > oveglay // Knor!

Next Input > quack

Porkey Says > ackquay // Oink!

Next Input > eat

Porkey Says > eatyay // Röf!

Next Input > omelet

Porkey Says > omeletyay // Grunz..

Next Input > are

Porkey Says > areyay // Sweek!

Next Input > honesty

Porkey Says > honestyyay // ¡Oinc!

Next Input > hourly

Porkey Says > hourlyyay // Ghrutu!

Next Input > at

Porkey Says > atyay // Röf!

Next Input > a

Porkey Says > ayyay // Knor!

Next Input > p

Porkey Says > eepay // ¡Oinc!

Next Input > h

Porkey Says > aitchyay // Röf!

Next Input >

Porkey Says > Oodbyegay...