Examen Sustitutorio

Curso: CC211/CC201 **Ciclo: 2019.1**

Debe enviar UN SOLO ARCHIVO de nombre `Exam.java`, no más; SE REVISARÁ SOLAMENTE EL ARCHIVO `Exam.java`.

1. (1 pto.) Cree la interfaz `Listable` que defina los métodos `void addItem(String string)` y `boolean isItemThere(String string)`.

2. (4 ptos.) Defina la clase `Dictionary` que implemente la interfaz `Listable`. Dicha clase debe tener un atributo `ArrayList`: `words` que almacenrá las palabras de un diccionario. Además, dicha clase debe implementar el método `addItem` tal que al pasarle una cadena la adicione a la coleción `words` siempre y cuando esta empiece con la letra 'i' o 'u'; caso contrario, muestre en la pantalla la palabra ingresada y un mensaje indicando que dicha palabra no ha sido adicionada. Finalmente, dicha clase debe también implementar el método `isItemThere` tal que al pasarle una cadena retorne verdadero si está en `words` y falso caso contrario.

3. (4 ptos.) Elabore la clase `Directory` que también implemente la interfaz `Listable`. Dicha clase debe tener un atributo `ArrayList`: `cities` que almacenará los nombres de ciudades norteamericas. Luego, dicha clase debe implementar el método `addItem` tal que al pasarle una cadena la adicione a la coleción `cities` siempre y cuando esta tenga una longitud mayor que cero y menor que 49; caso contrario, muestre en la pantalla la cadena ingresada y un mensaje indicando que dicha cadena no es el nombre de una ciudad. Por último, dicha clase debe también implementar el método `isItemThere` tal que al pasarle una cadena retorne verdadero si está en `cities` y falso caso contrario.

4. (4 ptos.) Dibuje el diagrama de clases de UML de las clases `Dictionary` y `Directory` y la interfaz `Listable`.

5. (3 ptos.) Cree la clase `FileManager` que implemente los métodos:

   - `public static Scanner openTextFile(String string)`
   - `public static void readTextFile(Scanner input, Listable list)`
   - `public static void closeTextFile(Scanner input)`

   donde: al pasarle una cadena (dirección de un archivo de texto) al método `openTextFile`, este nos retorne un objeto `Scanner` asociado a la dirección ingresada; luego, al pasarle un objeto `Scanner` asociado a un archivo de texto y una variable `Listable`,

lea cada línea de dicho archivo de texto y se la pase al método `addItem` de dicha variable `Listable`; finalmente, al pasarle un objeto `Scanner` asociado a un archivo de texto, lo cierra.

6. (4 ptos.) En el método `main` de la clase `Exam`: cree una variable `Dictionary` y otra `Directory`. Luego, defina un arreglo `Listable` inicializado con dichas variables. Por último:

   - Abra el archivo words.txt, léalo con el método `readTextFile` y ciérrelo.
   - Abra el archivo list-of-cities-in-usa-1.txt, léalo con el método `readTextFile` y ciérrelo.
   - Abra el archivo dictionaryTest.txt; léalo línea por línea y en cada lectura de línea muestre un mensaje en la pantalla indicando si dicha línea (cadena) es una palabra deletreada correctamente o no, según la variable `Dictionary` creada en la pregunta 2; finalmente, cierre dicho archivo.
   - Abra el archivo directoryTest.txt; léalo línea por línea y en cada lectura de línea muestre un mensaje en la pantalla indicando si dicha línea(cadena) es el nombre de una ciudad deletreada correctamente o no, según la variable `Directory` creada en la pregunta 3; finalmente, cierre dicho archivo.

17 de julio de 2019

`Print only when necessary`.