CANAUD Frédéric - CAMPREDON Thomas

Mini projet GénieLog

**Test d’un dictionnaire bilingue en java**

Module S6 – GénieLog

Licence 3 informatique

29 avril 2021



Enseignants : Emmanuel Conchon

# Exercice 1 :

Cet exercice consiste à initialiser la classe **Dictionnary** et **DictionnaryTest.**

On teste la méthode *getName()*, le test passe.

# Exercice 2 :

Ici on ajoute les méthodes pour ajouter et récupérer les traductions.

Pour l’instant on se sert d’une HashMap<String, String> pour stocker les traductions.

Lors du test on ajoute une traduction à l’aide de la méthode *addTranslation(String, String)* puis on vérifie que la valeur retourner par la méthode *getTranslation(String)* est bien la traduction qui a été ajouté précédemment.

# Exercice 3 :

On décide qu’il serait plus intéressant d’avoir plusieurs traductions pour un même mot.

On écrit alors le test pour la méthode *getMultipleTranslations(String)* qui retournera une ArrayList<String> contenant toutes les différentes traductions ajouté par l’utilisateur.

Cependant, l’implémentation actuelle ne permet que d’avoir une traduction pour un mot. On décide donc de changer le contenant des traductions en une HashMap<String, ArrayList<String>>.

Ce changement implique de modifier les méthodes déjà présentes dans la classe **Dictionnary**.

Par exemple, lors de l’ajout d’une traduction il faudra vérifier si le mot a déjà une traduction, si oui alors on en ajoute une autre à l’ArrayList, sinon on initialise l’ArrayList et on ajoute la nouvelle traduction.

De même pour le getter de traduction, où on renverra désormais la première valeur trouver dans l’ArrayList.

On en déduit de nouveau tests pour vérifier le bon fonctionnement de la classe.

# Exercice 4 :

Cet exercice demande de rajouter les traductions bidirectionnelles.

Pour ce faire nous avons fait le choix de rajouter une HashMap<String, ArrayList<String>> qui sera en charge de faire la traduction inversée. On aurait pu simplement garder l’implémentation précédente, mais la recherche inversée aurait nécessité de parcourir toute la HashMap pour retrouver la clé. Ici, on dédouble les données mais on s’assure d’avoir de meilleures performances.

On ajoute également une méthode *getReverseTranslations(String)* qui retournera une ArrayList<String> des mots donnant les traductions.

On modifie les méthodes d’ajout de traductions pour ajouter les inverses dans la nouvelle HashMap.

Les tests à passer son les même que pour les traductions classiques mais en partant des traductions.

# Exercice 5 :