

Architecture du projet java :

Nous avons 2 grandes parties :

- La partie main ou nous avons nos classes
- La partie test qui permet de tester les fonctionnalités de ces classes.

Dans la partie main nous avons plusieurs package :

- Tout d'abord le package "Tree" qui contient deux classes, la classe "Branch" et la classe "Tree". La classe Branch a 3 champs : 2 String qui sont les deux mots de la branche et le score de similarité entre ces 2 mots. Ces branches sont créées grâce au fichier que l'on récupère envoyé par le C. La classe Tree contient une liste de branches. Elle a une méthode qui lui permet de trier ces branches par ordre croissant de score de similarité ainsi qu'une méthode qui permet de vérifier qu'une branche est bien dans cet arbre.
- Ensuite on a un package "files_text" qui va venir accueillir le fichier envoyé par le C et le fichier créé par notre code.
- Enfin nous avons le package "Calcul" qui contient la classe "TreeOptimizer" qui nous permet de faire nos calculs sur notre arbre. Cette classe contient une grosse méthode "optimizeTrees" qui permet d'optimiser l'arbre. Cette méthode étant conséquente, nous avons donc créé 3 autres méthodes que cette méthode va appeler. Premièrement la méthode "removeSmallestLiaison" qui permet de retirer la liaison dite "inutile", donc la plus petite d'une boucle. On a ensuite la fonction "readSimilaritiesFromFile" qui permet de créer un arbre à partir d'un fichier que l'on reçoit. Enfin nous avons la méthode "writeIntoFiles" qui écrit l'arbre que nous avons créé donc l'arbre qui est optimisé dans un fichier.
- Nous avons également une classe Main que nous n'utilisons pas pour l'instant car nous n'avons pas encore relié le code java aux autres parties

Dans la partie test nous avons :

- Le package "Tree" qui contient la classe TreeTest. Cette classe permet de tester que la méthode de trie de liste fonctionne bien.
- Le package "files_text", dans ce package on simule des fichiers qui seraient envoyés par le C et on renvoie des fichiers qui contiennent les arbres optimisés. Les fichiers avec le préfixe "similarite" sont les faux fichiers d'entrée que le C pourrait nous envoyer. Les fichiers avec le préfixe "arbre" sont des fichiers que l'on crée dans notre code. Dans la deuxième partie du nom de ces fichiers c'est la complexité de ces arbres à être optimisés : "Simple", "Medium" ou "Advanced" comme ça nous pouvons couvrir correctement notre code.
- Enfin le package Calcul qui contient une classe TreeOptimizerTest et qui permet donc de tester que l'on renvoie bien le bon arbre optimisé.

Tous les fichiers que l'on manipule avec le C sont des fichiers en .txt