Mission 5 : Rapport Final

Groupe 2.2

Vendredi 28 Novembre 2014

Introduction

Il nous a été demandé de réaliser un programme de compression et de décompression de fichiers. Ce programme devait s'inspirer de l'algorithme de Huffman. Le codage de Huffman faisant appel à une file de priorité, il était nécessaire d'implementer cette file de la manière la plus efficace possible.

Choix d'implémentation

Le programme comporte 5 classes, InputBitStream, OutputBitStream, Compress, Decompress, HTree. Les deux premières classes citées, nous ont été fournies. Vous trpuverez ci-dessous une explication pour chacune d'elles.

InputBitStream et OutputBitStream (Question 7)

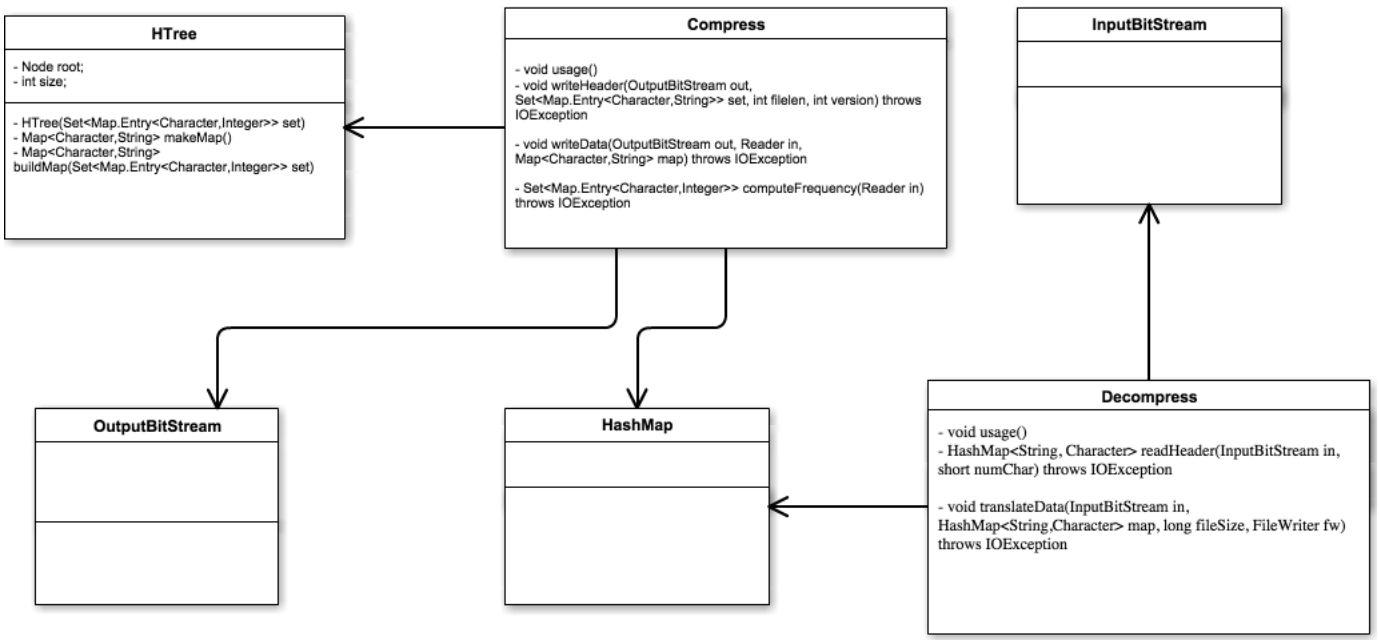
Compress

Decompress

HTree

La classe HTree utilise un SetMap contenant tous les caractères utilisés dans le fichier ainsi que leur fréquence pour créer un arbre de Huffman à l'aide d'une priorityQueue. L'abre de Huffman ainsi créé est ensuite parcouru de manière récussive pour créer un Hashmap contenant les caractères comme clé et comme valeur un string représentant ce caractère compressé. Nous avons pris la décision d'implémenter complétement la classe HTree et donc de ne pas étendre la classe générique TreeSet. L'avantage de cette approche, c'est que cette classe correspond exactement à ce qu'on attend d'elle. Par contre, l'inconvénient c'est que l'on perd en portabilité.

Diagramme de classe (Question 8)



Test (Question 9 et 10)

Conclusion

La décomposition des classes dans notre programme, nous a facilité la répartition des taches, l'utilisation de multiple classe permet aussi de pouvoir être plus modulable. Tant au niveau de la repartition du travail, qu'au niveau de l'ajout de nouvelles fonctions pour adapter notre programme.