ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

LOG320 – LABORATOIRE #1  
ALGORITHME DE HUFFMAN

PAR  
MARC-ANDRE DESTREMPES – AJ86290  
SAMUEL BEAUCHEMIN - AJXXXXX

MONTRÉAL, 10 mai 2013

Table des matières

[Description de votre programme 3](#_Toc355935815)

[Analyse de complexité des algorithmes 4](#_Toc355935816)

[Description des problèmes rencontrés 5](#_Toc355935817)

[Description des améliorations que vous avez implémentées 6](#_Toc355935818)

# Description de votre programme

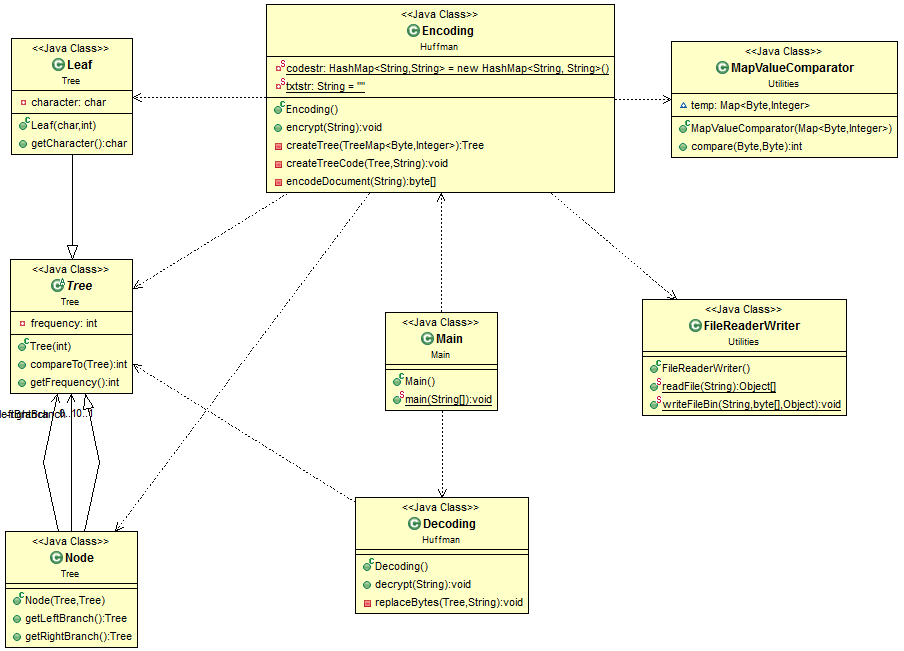


Figure - Diagramme de classe

Le programme se divise en 2 sections, l’encodage et le décodage. L’encodage est ce qui permet de compression un fichier en utilisant l’algorithme de Huffman tandis que le décodage permet de faire l’opération inverse. L’arbre de fréquence est stocké dans un objet Tree qui contient des Node ou des Leaf. Les Leaf représentent chacune des lettres avec leur fréquence, tandis que les Node sont l’intersection de soit deux Node ou deux Leaf et contient l’addition de la fréquence des deux classes.

L’encodage commence par lire le fichier voulu. Ensuite, il construit l’arbre et il construit le code représentant chacune des lettres. Par la suite, il remplace chacun des caractères ASCII par le code correspondant construit précédemment. Pour terminer, il écrit le fichier en stockant l’arbre dans le header et le nouveau contenu du fichier dans le payload.

Le décodage …

# Analyse de complexité des algorithmes

Bla bla

# Description des problèmes rencontrés

Un premier problème rencontré a été l’écriture du fichier lors de l’encodage. Le header du fichier s’écrit correctement, mais le payload ne s’écrit pas en binaire, il écrit le binaire comme une string. Alors, au lieu de compresser le fichier, le fichier augmente de taille car tout est écrit comme une string.

# Description des améliorations que vous avez implémentées

Bla bla