LOG3430 – Méthodes de test et de validation du logiciel

Laboratoire 3

Tests OO – MaDUM

Département de génie informatique et de génie logiciel

École Polytechnique de Montréal



Soumis par

Roman Zhornytskiy (1899786)

Hakim Payman (1938609)

Gabriel Tagliabracci (1935775)

Groupe : 02

Soumis à Noureddine Kerzazi

Hiver 2020

4. Travail à effectuer

4.1. Identifier la situation la plus favorable pour un test MaDUM ensuite construire le MaDUM en identifiant respectivement les constructors, reporters, transformers, et autres pour les attributs de la classe huffman.py.



4.2. À l’aide de unittest, écrire une classe de test unitaire pour tester les tranches identifiées dans l’étape précédente. Pour chaque tranche, la séquence des méthodes doit suivre le principe de MaDUM.

**Roman : Je ne sais pas si on a bien fait le #1, car dans la question 2, il faut tester toutes les fonctions pour chaque « tranche ». Alors, il faut faire 5 x 7 = 35 – 9 = 26 tests…**

4.3. Est-ce que cet exemple montre une ou plusieurs limitations de MaDUM ? Si oui, pouvez-vous les citer ?

Oui

4.4. À l’aide de l’outil Coverage.py, évaluez la couverture de la classe Huffman.py (dans le fichier huffman.py) et identifiez les parties de code non couvertes, s’il y en a. Pour les parties non couvertes, essayez de faire des tests boîte blanche pour atteindre la couverture maximale.

4.5. Qu’est-ce qui se passe si la fonction responsable du padding ne fonctionne pas comme attendu ? Proposez une solution pour corriger le bogue.