



TP 1 : Génération de cas de test aléatoire

Cours LOG6305

Prof. Giuliano Antoniol

Chargé de Lab : Housseem Ben Braiek

Date de lab : 16/01/19

Date d'échéance : 29/01/19

Critères d'évaluation

1. Le code complété doit être compressé en zip et renommé comme *prénom_nom_de_famille_matricule.zip* ;
2. Un rapport contenant les réponses aux questions sous format PDF et renommé comme *prénom_nom_de_famille_matricule.pdf*.

Ce travail devrait être soumis à l'adresse mail : housseem.ben-braiek@polymtl.ca au plus tard le 29/01/19 à 23 :59h ; Sinon des pénalités seront appliquées (**5 points** de déduction pour le retard de toutes les **24 heures**)

Code (11 pts)

Objectif 1. Génération de données : Les données du triangle doivent être générées automatiquement ;

Objectif 2. Couverture de branches : La couverture de branches est calculée de façon incrémentielle et devrait atteindre jusqu'à 95% (si votre couverture atteint seulement 70%, 3 points seront déduits).

Objectif 3. Commentaires : La classe *RandomCoverageTest.java* doit être clairement commentée ; Sinon une pénalité sera appliquée.

Rapport (9 pts)

Question 1. Pourquoi est-il plus compliqué d'atteindre une couverture élevée lorsque l'intervalle des entiers augmente ? Par exemple, en utilisant des nombres aléatoires compris entre 0 et 100 en comparaison de 0 à 1000.

Question 2. Dans vos expériences, êtes-vous capable d'atteindre une couverture de 100% ? Si ce n'est pas le cas, expliquer pourquoi ?

Question 3. À votre avis, quelles sont les limites de la génération de nombres aléatoires comme données d'essai et comment pouvons-nous surmonter ces limitations pour atteindre une couverture élevée dans les tests logiciels ?