TP2 de C++

Tests unitaires et fonctionnels

B3211 : Quentin « johnny » Guye, Louis Mer

Introduction

Lors de ce TP nous avons utilisé un programme de non régression en shell, ainsi qu'un programme C++ de tests unitaires.

I – Tests unitaires

Certains tests du programme ne sont pas confirmés automatiquement, l'utilisateur doit vérifier lui même si les entrées donnent un résultat correct en sortie.

Procédure test1 :
- Teste le constructeur de TrafficData ainsi que tous les getters de la classe.

Procédure test2 :
- Teste le constructeur avec l'identifiant en paramètre de Sensor ainsi que le getter GetID .

Procédure test 3 :
- Teste la méthode AddEvent et le getter GetEventPerDay de Sensor.

Procédure test 4 :
- Teste la méthode GetStats de Sensor.

Procédure test 5 :
- Teste la méthode GetStatsJamADAH de Sensor.

Procédure test 6 :
- Teste la méthode GetStatsForADayWeek de Sensor.

Procédure test 7 :
- Teste la méthode OptimizedPath de Sensor.

Procédure test 8 :
- Teste le constructeur, la méthode AddEvent et le getter GetSensor de Traffic.

Procédure test 9 :
- Teste la méthode GetStats de Traffic.

Procédure test 10 :
- Teste la méthode GetStatsJamADAH de Traffic.

Procédure test 11 :
- Teste la méthode GetStatsForADayWeek de Traffic.

Procédure test 12 :
- Teste la méthode OptimizedPath de Traffic.

II – Tests fonctionnels

Le programme shell Nrtester réalise 5 tests, un pour chaque commande qu'il est possible d'appeler depuis le main : ADD, STATS_C,STATS_D7,JAM_DH,OPT.
Des entrées ainsi que les sorties correspondantes ont été générées, et ce programme permet de vérifier que l'application renvoie les mêmes résultats avec les fichiers .in en entrée et .out en sortie.