Compte rendu TP Trafic de Lyon - C++

Tests fonctionnels

I. Choix généraux

Cette classe nous permettra d'interagir avec le programme en simulant l'entrée de commandes par l'utilisateur. On pourra donc vérifier, en fonction des commandes entrées, si le programme répond correctement

II. La classe 'TestsGestionTrafic'

II.1. Méthode 'LancementTests'

Cette méthode permet de lancer tous les tests (méthodes privées) de la classe. Elle est utilisée dans le main pour choisir entre le lancement réel de l'application ou le lancement des tests.

II.2.Méthode 'testAdd'

Dans cette méthode, nous testons si l'ajout d'un événement d'un capteur est bien assimilé par l'application.

Pour cela nous affichons toutes les durées qui ont été stockées après chaque ajout.

Nous avons testé l'ajout de données aux moments critiques d'une année : entre deux minutes, entre deux heures, entre deux jours, entre deux semaines et entre deux mois.

Ces tests n'ont pas été faits pour la vérification de l'ajout dans l'architecture qui gère le maximum de bouchons simultanés. En effet cette architecture est complètement à part dans notre application et nous ferons donc ces tests dans la méthode de tests de maxBouchon.

Nous considérons satisfaisants les résultats des tests de cette méthode.

II.3. Méthode 'testStatsCapteur'

Dans cette méthode, nous testons si les résultats du calcul des statistiques d'un capteur sont bien conformes à ceux que l'on peut retrouver en calculant à la main.

Nous avons testé les statistiques aux moments critiques d'une année : entre deux minutes, entre deux heures, entre deux jours, entre deux semaines et entre deux mois.

Nous avons également testé si l'extinction d'un capteur était bien prise en compte par l'application. C'est-à-dire que si un capteur arrête d'émettre pendant plus de 5minutes puis qu'il recommence, seulement les premières 5minutes de non activité seront comptabilisées par l'application.

Nous considérons satisfaisants les résultats des tests de cette méthode.

II.4. Méthode 'testStatsJourSemaine'

Dans cette méthode, nous testons si les résultats du calcul des statistiques d'un sur un jour de la semaine sont bien conformes à ceux que l'on peut retrouver en calculant à la main.

Nous avons testé statistiques aux moments critiques d'une année : entre deux minutes, entre deux heures, entre deux jours, entre deux semaines et entre deux mois.

Nous avons également testé si l'extinction d'un capteur était bien prise en compte par l'application. C'est-à-dire que si un capteur arrête d'émettre pendant plus de 5minutes puis qu'il recommence, seulement les premières 5minutes de non activité seront comptabilisées par l'application.

Nous considérons satisfaisants les résultats des tests de cette méthode.

II.5. Méthode 'testStatsCapteurStatsJourSemaineHeure'

Dans cette méthode, nous testons si les résultats du calcul des statistiques d'un sur un jour de la semaine et une heure précise sont bien conformes à ceux que l'on peut retrouver en calculant à la main.

Nous avons testé statistiques aux moments critiques d'une année : entre deux minutes, entre deux heures, entre deux jours, entre deux semaines et entre deux mois.

Nous avons également testé si l'extinction d'un capteur était bien prise en compte par l'application. C'est-à-dire que si un capteur arrête d'émettre pendant plus de 5minutes puis qu'il recommence, seulement les premières 5minutes de non activité seront comptabilisées par l'application.

Nous considérons satisfaisants les résultats des tests de cette méthode.

II.6. Méthode 'testStatsMaxBouchons'

Dans cette méthode, nous testons si les résultats de la recherche du maximum de bouchons simultanés est bien conforme à la logique.

Nous avons testé le cas simple, dans lequel le maximum apparaît (pic de trafic) puis n'est plus remis en question. Ce cas fonctionne.

Nous avons ensuite testé le cas où un autre maximum (plus important) vient remplacer le premier. La date est bien retrouvée mais le calcul est souvent faux.

Finalement nous avons testé le cas où le pic de bouchon ne se situe pas à un ajout de capteur (donc extinction d'un capteur vert ou jaune qui entraine un pic de bouchon à la date de l'extinction). La date est bien retrouvée mais le calcul semble incomplet.

Nous avons également vérifié que l'ajout des données dans l'architecture propre à maxBouchons se faisait correctement (en affichant les attributs modifiés à l'ajout d'un évènement). Ces tests sont plutôt concluants, excepté l'ajout d'une date d'extinction d'un capteur, cela explique peut-être pourquoi le test ci-dessus ne fonctionne pas entièrement.

Nous ne considérons pas satisfaisants les résultats des tests de cette méthode.
Si la recherche du maximum de bouchon simultané est si peu efficace c'est parce qu'elle ne partage pas la même architecture que les autres méthodes. Nous avons donc passé plus de temps à améliorer l'architecture des 4 autres méthodes que celle-ci.

Conclusion :

Les tests sont concluants sauf pour la méthode qui ne partage pas la même architecture. C'est donc plus une erreur de conception que de réalisation. Ceci nous servira d'exemple (à ne pas suivre) pour la suite.