

HLIN 505 Correction Examen Session 1 2017 (partie Tests)

Q7. 7 méthodes de test. 2 appels pour la classe MonTest, et 9 appels pour la classe TestParam (une fois par élément retourné par la méthode data, qui retourne ici une collection de 9 éléments).

Q8.

On exécute la méthode setUp (car annotation @Before) puis l'une des deux méthodes de test puis la méthode setUp puis la méthode de test non exécutée précédemment. Les deux méthodes de test sont exécutées dans un ordre a priori quelconque.

Q9.

Il faut :

- * ajouter à chaque élément retourné par la méthode data le résultat attendu pour l'appel à la méthode à tester contientEspaceOuApostropheouVirgule avec la donnée de test courante. On le rajoute par exemple en dernière position de l'élément.

- * dans le constructeur, on rajoute un paramètre booléen qui sert à récupérer le résultat attendu, et on le stocke dans un nouvel attribut (déclaré dans la classe TestParam) que l'on nomme par exemple contientEspaceApOuVirgule.

- * augmenter la méthode test pour y ajouter : un appel à la méthode à tester contientEspaceOuApostropheouVirgule, et l'oracle, c'est-à-dire ici une assertion vérifiant que le résultat est celui contenu dans contientEspaceApOuVirgule

Q10.

L'important dans cette question était bien de discuter du diagnostique et pas de la facilité d'écriture des tests ni du choix des données de test.

La classe TestParam permet facilement d'identifier sur quelle donnée de test le test échoue, grâce au test paramétré. Ensuite les deux méthodes testées étant dans la même méthode de test, on les distingue un peu moins facilement ici, d'autant qu'il n'y a pas de texte dans l'assertion.

La classe MonTest, elle, a une méthode de test par méthode testée, mais les données de test ne sont pas bien différenciées. Si une assertion est violée, il sera en l'état difficile de savoir avec quelle donnée de test elle l'a été (quel "mot" de la boucle).