

TP Test (noté)

Algorithmique – Programmation FIP (ING39)

durée 1h30

V. Aponte, S. Rosmorduc, P. Courtieu

Résumé

Deux contrats de méthodes vous sont donnés avec des implémentations vides (lèvent une exception ou retournent une valeur quelconque). N'Y TOUCHEZ PAS (sinon vous aurez des conflits à gérer pendant le TP). Vous devez écrire des tests pour ces contrats. Une implémentation correcte des deux méthodes vous sera fournie environ 40mn avant la fin du TP, vous permettant de réajuster vos tests si le comportement attendu n'était pas exactement celui auquel vous pensiez (en principe les contrats sont clairs).

Calcul de la note : Vos tests seront effectués sur différentes versions *buggées* des deux fonctions. Le pourcentage de bugs détectés sera votre note. Vous n'avez évidemment pas accès aux versions buggées : votre travail de testeur¹ est d'écrire, seulement à partir du contrat, un jeu de test qui détectera un maximum d'erreurs.

Attention si votre projet ne compile pas ou si vos tests échouent sur le code correct fourni **vous avez 0/20**. Vous aurez 40mn pour vous assurer que ce n'est pas le cas donc ça ne devrait pas arriver. *Mais vérifiez quand même régulièrement*, ça ne prend qu'un clic !). Commitez régulièrement (quand ça compile) et n'oubliez pas de PUSSEZ aussi régulièrement.

Mise en place

Le TP est dans votre projet gitlab. Clonez le si ce n'est pas déjà fait ou bien mettez à jour un clone existant (avec `git pull`).

Ensuite ouvrez le projet dans votre IDE. Faites une modification insignifiante, ensuite (add+)commit et un push (soit via Eclipse soit via la ligne de commande). VÉRIFIEZ QUE LA MODIFICATION EST BIEN POUSSÉE SUR GITLAB. C'est bon ? OK vous êtes prêt à commencer. N'OUBLIEZ PAS DE FAIRE DES COMMITS (et push) RÉGULIERS.

Ce qu'il faut faire

Vous devez écrire des tests pour les 2 méthodes de la classe `tp3.Tp3`. La classe de test est déjà créée. Les implémentations des deux méthodes **sont vides**. Donc vos tests échoueront pour la plupart tant que vous n'aurez pas reçu les implémentations correctes.

Vous devez donc commencer l'écriture des tests avant de pouvoir les lancer, sur la base du contrat écrit. En revanche vérifiez en permanence que la *compilation* passe (pas de code souligné en rouge dans l'IDE).

Les contrats sont plus lisibles en javadoc (projet ⇒ Generate javadoc).

40mn avant la fin du TP les implémentations **correctes** seront poussées dans le fichier `tp3.Tp3.java` (donc n'y touchez pas vous-même sauf si vous savez gérer les conflits git...), afin que vous puissiez corriger les tests qui ne passeraient pas.

1. On parle de test de la *boîte noire*, où on teste « à l'aveugle » sans voir le code, par opposition au test de la *boîte blanche* où on crée les tests à partir du code.

Conseils importants *Il n'est pas demandé d'implanter ces méthodes*, ne le faites pas. D'abord parce-que vous risquez de les implanter incorrectement et de baser vos tests sur des comportements erronés. Ensuite car c'est une perte de temps. Basez vous sur les contrats et attendez la version correcte pour confirmer/corriger. Vous pouvez aussi poser des questions si la documentation vous semble ambiguë.

Encore une fois : ne touchez pas au fichier `src/main/java/tp3/Tp3.java`. Ça évitera des conflits lors du push de l'implantation correcte.

Le TP sera corrigé en récupérant votre projet git donc n'oubliez pas de (add+)commiter et pusher votre projet à la fin du TP. MAIS SURTOUT commitez et poussez *au fur et à mesure du TP* (toutes les 10mn, quand ça compile). Comme ça vous ne perdrez pas trop de travail en cas de problème, et par ailleurs vous ne serez pas coincé par un problème git de dernière minute.