

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 (lève une exception). 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 !*).

Mise en place

Un projet gitlab a été créé pour vous (tp_test_<username>). Vous avez accès à ce projet via gitlab. Clonez le soit via l'interface d'eclipse, soit avec la commande suivante dans un terminal :

```
cd ~/eclipse-worspace # optionnel
git clone https://gitinfo.cnam.fr/gitlab/fipl_2020/tp_test_XXX.git
```

Ensuite ouvrez le projet dans Eclipse. 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.

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 étant vides, les tests échoueront pour la plupart tant que vous n'aurez pas reçu l'implémentation correcte. 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.

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

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.

Le TP sera corrigé en récupérant votre projet git donc n'oubliez pas de (add+)commit 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).

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

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.