

Mémo. Utilisation des tests

Unité SMINFL3A : Programmation Java orientée objet

1 Généralités

La plupart des exercices sont fournis avec une librairie de tests, ayant pour but de permettre de tester son propre code.

Les tests disponibles pour un exercice donné sont décrits dans l'énoncé de ce dernier. De façon générale, une classe de test T est fournie pour chaque classe à écrire C , et une méthode de test `testUneMéthode` est proposée par cette classe pour chaque méthode `uneMéthode` à écrire.

Pour utiliser les tests pour une classe donnée C , il faut donc :

- télécharger la librairie de tests, qui est fournie sous forme d'une archive Java (`.jar`);
- écrire une classe exécutable, dont la méthode `main` crée une instance de la classe de tests T , et appelle sur elle les méthodes de test `testUneMéthode` pour chacune des méthodes `uneMéthode` déjà écrites dans C (le `main` de test est donc à mettre à jour au fur et à mesure que l'on écrit les méthodes demandées dans l'énoncé);
- compiler cette classe exécutable et les classes écrites en ajoutant la librairie de tests au `classpath`;
- exécuter cette classe exécutable en ajoutant à nouveau la librairie de tests au `classpath`.

Un exemple de classe exécutable de test est donné dans le code de l'exemple des personnes.

2 Points de vigilance

- Seules les méthodes `testUneMéthode` correspondant aux méthodes `uneMéthode` déjà écrites doivent être appelées dans le `main`.
- Les méthodes de test sont listées dans les énoncés dans un ordre précis, qui correspond à l'ordre dans lequel il est demandé d'écrire les méthodes; cet ordre doit être respecté, une méthode de test pouvant utiliser une autre méthode.
- Les noms des méthodes demandées dans les énoncés doivent être respectés exactement, casse incluse, car c'est ainsi que les tests les utilisent; dans le cas contraire, l'exécution peut générer une erreur comme

```
Exception in thread "main" java.lang.NoSuchMethodError:
'int_persons.Person.getCurrentAge()'
    at personstests.PersonTests.testGetCurrentAge(...)
    at persons.Test.main(Test.java:19)
```

par exemple parce qu'on a orthographié « `getCurrentage` »; on peut également obtenir une erreur de type `NoClassDefFoundError`, en particulier si c'est le nom de la classe qui est mal orthographié.

- Les tests utilisant des interfaces ou classes abstraites demandées dans l'énoncé définissent parfois des implémentations par défaut (des *mock objects*); on ne doit donc pas ajouter de méthodes (non définies) aux interfaces, ni de méthodes abstraites aux classes, autres que celles demandées dans l'énoncé (les *mock objects*, qui implémentent exactement ce qui est attendu dans l'énoncé, ne compileraient pas avec des méthodes abstraites additionnelles).
- Les tests sont conçus pour s'arrêter à la première erreur; ce n'est donc pas parce qu'il n'y a qu'une ligne d'erreur qu'il n'y a qu'un test qui échoue.
- Attention enfin, ce ne sont que des tests : **le fait que les tests soient réussis ne garantit pas de manière absolue que le code est correct !**