

Travaux sur Machine Encadrés No5

Test MBT

Frédéric Peschanski

4 mars 2019

Projet : système de contrôle d’un ascenseur

On reprend dans cet exercice les spécifications du système de contrôle de circulation étudié en TD. Sur le site de l’UE on trouvera une archive contenant une implémentation partielle des contrats associés à la spécification du système. On dispose aussi dans le sous-répertoire `jars/` de trois implémentations de l’ascenseur sans leur code source.

Dans cette archive on trouve les paquetages :

- dans le répertoire `src/` :
 - `lift.services` : les interfaces de service
 - `lift.decorators` : les décorateurs de service
 - `lift.contracts` : les implémentations de contrat
 - `lift.impl` : l’implémentation du gestionnaire de commandes avec son code source
 - `lift.test` : une classe abstraite dont il faudra hériter pour rédiger les scripts de test
- dans le répertoire `jars/` :
 - `junit.jar` : le *framework* junit (version 4.1.0)
 - `lift1.jar` : la première implémentation de l’ascenseur
 - `lift2.jar` : la seconde implémentation de l’ascenseur
 - `lift3.jar` : la troisième implémentation de l’ascenseur
- dans le fichier `ant build.xml` les cibles :
 - `compile` : pour la compilation des sources (tests inclus)
 - `test1` : pour exécuter les tests sur la première implémentation
 - `test2` : pour exécuter les tests sur la seconde implémentation
 - `test3` : pour exécuter les tests sur la troisième implémentation
 - `dist` : pour la soumission

Question :

- On déduira des cas de test élaborés en TD des scripts de tests exécutables que l’on placera dans la classe `AbstractLiftTest`
- Pour chacune des implémentations on créera une sous-classe concrète de `AbstractLiftTest` pour pouvoir lancer les tests sur la bonne implémentation (cf. cibles `test1`, `test2`, `test3`)
- On lancera les tests et on conservera les rapports de test générés par junit (fichiers `.txt`).