

TP N° 1

Les outils de test unitaires

Exercice 1. Tests unitaires et évaluation de la qualité des tests

Le but est de tester l'opération calculant le déterminant d'une matrice et d'évaluer la qualité des tests unitaires.

- Créer un nouveau projet et intégrer le code source de l'exercice 1 et les bibliothèques correspondantes.
- Écrire le test unitaire validant l'opération matricielle *determinant* de la classe *MatrixMathematics*, qui calcule le déterminant d'une matrice.
- Calculer la couverture du test.
- Compléter les tests unitaires pour atteindre une couverture de 100% pour la classe *MatrixMathematics*.
- Évaluer la qualité des tests avec Pitest sur la classe *MatrixMathematics*.

Exercice 2. Tests avec base de données en mémoire et tests doubles

Le but est de tester différentes méthodes permettant de réserver une place de parking.

Le code contient de trois packages :

1. Le package *entity*: contient les différentes entités.
2. Le package *dao* contient les différentes classes permettant de manipuler directement la base de données.
3. Le package *service* contient les classes métier.

Le travail demandé consiste à :

1. Créer un nouveau projet et intégrer le code source et les librairies.
2. Écrire les tests unitaires de la classe *ReservationDao*.
3. Écrire les tests unitaires des méthodes *createReservation* et *cancelReservation* de la classe *ReservationManager*.
4. Écrire le test d'intégration de la méthode *cancelReservation* de la classe *ReservationManager*.

L'utilisation d'une base de données en mémoire est recommandée pour les tests nécessitant la manipulation de la base de données. Les paramètres d'accès à une base de données en mémoire sont:

- *username* = "sa"
- *password* = ""
- *jdbc_driver* = "org.h2.Driver"
- *db_url* = "jdbc:h2:mem:test"

Exercice 3. Test du client d'une API Rest

Le but est d'écrire les tests unitaire et d'intégration du client de l'API Rest du GitHub. Le service Rest <https://api.github.com/users/{username}> retourne les informations publiques de l'utilisateur github identifié par son username.

1. Créer un nouveau projet et intégrer le code source et les librairies.
2. Écrire le test unitaire de la classe *UserServiceClient*.
3. Écrire le test d'intégration de la classe *UserServiceClient*.