

TP N° 1

Les outils de test unitaires

Exercice 1. Les tests unitaires avec Junit

Il s'agit de tester un ensemble d'opérations sur les matrices et d'évaluer la qualité des tests, en réalisant les tâches suivantes:

- Ecrire les tests unitaires validant les opérations matricielles *transpose*, *determinant*, *cofactor et inverse* (Ecrire pour chaque méthode un seul test)
- Calculer le pourcentage de couverture du code
- Compléter les tests unitaires pour tester toutes les parties de la classe *MatrixMathematics*
- Evaluer la qualité des tests avec Pitest sur la classe *MatrixMathematics*

Exercice 2. Les tests doubles avec Mockito

Il s'agit d'utiliser les tests doubles pour tester les méthodes métiers d'une simple application de réservation d'hôtel.

Le code source est composé de trois packages:

1. *com.example.model*: contient les entités du domaine
2. *com.example.dao*: contient les services d'accès à la base de données
3. *com.example.service*: contient les services métiers

Utilisez Mockito pour tester la réservation d'une chambre et l'annulation d'une réservation (les méthodes de la classe *BookService*)

Livrables.

Les fichiers source *.Java* des tests et le rapport HTML de Pitest

Exercice 3. Application de l'approche TDD

Utilisez l'approche TDD pour écrire un programme qui affiche des nombres de 1 à 100. Pour les nombres multiples de 3, il affiche le mot “Fizz” au lieu d’afficher le nombre et pour les multiples de 5, il affiche le mot “Buzz”. Pour les nombres qui sont multiples de 3 et 5, le programme affiche le mot “FizzBuzz”.