

Examen Pratique - Tests Unitaires et d'Intégration en Java

Objectif :

L'objectif de cet examen est d'évaluer vos compétences dans la rédaction de tests unitaires et d'intégration en Java. Vous travaillerez sur une application de gestion de projets.

Consignes générales :

1. Utilisez le framework de tests unitaires JUnit pour les tests unitaires.
2. Pour les tests d'intégration, vous pouvez utiliser des classes de tests spécifiques ou JUnit selon votre préférence.
3. Créez un fichier de tests distinct pour chaque composant que vous testez.
4. Assurez-vous d'inclure des commentaires expliquant le but de chaque test.
5. Respectez les conventions de nommage des tests.
6. Tous les tests doivent être réussis pour obtenir la note maximale.

Partie 1: Tests Unitaires pour une classe de gestion de tâches

Contexte :

Vous travaillez sur une application de gestion de projets. Vous devez tester la classe `GestionTaches` qui gère les tâches dans un projet. La classe a les méthodes suivantes :

- `ajouterTache(String titre, String description)`: ajoute une nouvelle tâche au projet.
- `completerTache(String titre)`: marque une tâche comme complétée.
- `verifierTache(String titre)`: vérifie si une tâche est complétée.

Tâches :

1. Écrivez des tests unitaires pour la classe `GestionTaches`. Assurez-vous de tester les cas où vous essayez de compléter une tâche qui n'existe pas.
2. Testez également le cas où vous essayez de vérifier une tâche qui n'est pas dans la liste.
3. Assurez-vous de couvrir tous les cas possibles pour la méthode `verifierTache`.

Partie 2: Tests d'Intégration pour une classe de Projet

Contexte :

La classe `Projet` représente un projet dans l'application. Elle a les attributs suivants :

- `nom` (String): le nom du projet.
- `taches` (List<Tache>): une liste des tâches associées au projet.

Tâches :

1. Écrivez des tests d'intégration pour la classe `Projet`. Testez la manière dont la classe `Projet` interagit avec la classe `GestionTaches`. Assurez-vous de vérifier que les tâches ajoutées à un projet sont également mises à jour dans la classe `GestionTaches`.

Partie 3: Tests pour une fonction utilitaire

Contexte :

Vous avez une fonction utilitaire appelée `calculerDureeTotale(List<Tache> taches)`, qui calcule la durée totale nécessaire pour toutes les tâches d'un projet. Chaque tâche est un objet avec une propriété `duree` représentant la durée estimée pour cette tâche.

Tâches :

1. Écrivez des tests pour la fonction `calculerDureeTotale`. Assurez-vous de tester les cas où la liste de tâches est vide et les cas où les tâches ont des durées négatives.

Partie 4: Ajouter un controller pour afficher la liste des tâches et chaque tâche par nom et faite les tests nécessaire pour ce controller