

Plan de Tests JUnit

1. LocationTest

Objectifs des tests

Méthode de test	Actions	Attendu
testRegister	Créer un nouvel utilisateur avec informations personnelles	Retourne un ID valide (≥ 0)
testLogin	Se connecter avec identifiants valides	Retourne true
testGetCurrentUser	Récupérer l'utilisateur connecté	L'utilisateur retourné correspond aux données attendues
testRentMovie	Louer un film disponible	Film ajouté à la liste des locations de l'utilisateur
testIsRentable	Vérifier disponibilité d'un film	Retourne true si film disponible
testAddReview	Ajouter un avis sur un film	Avis ajouté aux critiques du film
testAllMovies	Récupérer catalogue complet	Liste contient le film test
testMoviesByGenre	Rechercher films par genre	Liste contient les films du genre spécifié

2. ArtistManagerTest

Objectifs des tests

Méthode de test	Actions	Attendu
testAddArtist	Ajouter un artiste	Artiste ajouté à la liste
testAddActor	Ajouter un acteur	Acteur ajouté aux listes acteurs et artistes
testAddDirector	Ajouter un réalisateur	Réalisateur ajouté aux listes réalisateurs et artistes
testDeleteArtist	Supprimer un artiste	Artiste retiré de toutes les listes
testGetArtistsWithPredicate	Filtrer artistes par prédicat	Liste filtrée correctement
testSerializeAndParse	Sérialiser/désérialiser données	Données restaurées identiques aux originales

3. ArtistTest

Objectifs des tests

Méthode de test	Actions	Attendu
-----------------	---------	---------

Méthode de test	Actions	Attendu
testConstructor	Créer un artiste avec paramètres	Attributs correctement initialisés
testLastNameGetterSetter	Modifier/accéder au nom	Nom correctement modifié/retourné
testFirstNameGetterSetter	Modifier/accéder au prénom	Prénom correctement modifié/retourné
testNationalityGetter	Accéder à la nationalité	Nationalité correctement retournée
testAddMovie	Ajouter des films à un artiste	Films correctement ajoutés à la liste
testToString	Vérifier format chaîne	Format "Nom Prénom" respecté
testEquals	Comparer des artistes	Égalité basée sur nom/prénom/nationalité
testSerialize	Sérialiser un artiste	JSON correct généré
testParse	Désérialiser un artiste	Objet correctement restauré

4. LocationAdminTest

Objectifs des tests

Méthode de test	Actions	Attendu
testCreateArtist	Créer un nouvel artiste	Artiste créé avec attributs corrects
testDeleteArtist	Supprimer un artiste existant	Artiste supprimé avec succès
testCreateMovie	Créer un nouveau film	Film créé avec attributs corrects
testAddActors	Ajouter des acteurs à un film	Acteurs correctement associés au film
testAddGenres	Définir genres d'un film	Genres correctement associés au film
testAddPoster	Ajouter une affiche à un film	Chemin de l'affiche correctement enregistré
testDeleteMovie	Supprimer un film existant	Film supprimé avec succès
testGetAllMovies	Lister tous les films	Liste complète retournée
testGetMoviesByDirector	Filtrer films par réalisateur	Films correctement filtrés
testRentalAvailability	Gérer disponibilité location	Statut de location correctement mis à jour

5. ReservationTest

Objectifs des tests

Méthode de test	Actions	Attendu
testReservationConstructorAndGetters	Créer une réservation et vérifier les getters	Les attributs correspondent aux valeurs fournies

Méthode de test	Actions	Attendu
testSetters	Modifier tous les attributs d'une réservation	Les nouvelles valeurs sont correctement enregistrées
testEquals	Comparer des réservations	Égalité basée sur utilisateur/film/date/durée

6. ReviewManagerTest

Objectifs des tests

Méthode de test	Actions	Attendu
testAddReviewWithParameters	Ajouter un avis avec paramètres individuels	Avis correctement ajouté et récupérable
testAddReviewObject	Ajouter un objet Review complet	Avis correctement ajouté et récupérable
testAddDuplicateReview	Tenter d'ajouter un avis en double	Exception IllegalArgumentException levée
testGetReviewsWithPredicate	Filtrer les avis avec un prédicat	Liste filtrée correctement retournée

7. MovieManagerTest

Objectifs des tests

Méthode de test	Actions	Attendu
testReset	Réinitialiser la liste des films	Liste vide après reset
testAddMovie	Ajouter un nouveau film	Film correctement ajouté à la liste
testGetMovies	Récupérer tous les films	Liste complète retournée
testGetMoviesByPredicate	Filtrer les films avec un prédicat	Films correctement filtrés
testDeleteMovie	Supprimer un film	Film correctement retiré de la liste

8. MovieTest

Objectifs des tests

Méthode de test	Actions	Attendu
testConstructor	Créer un film avec paramètres	Attributs correctement initialisés
testTitleGetterSetter	Modifier/accéder au titre	Titre correctement modifié/retourné

Méthode de test	Actions	Attendu
testYearGetterSetter	Modifier/accéder à l'année	Année correctement modifiée/retournée
testDirectorGetterSetter	Modifier/accéder au réalisateur	Réalisateur correctement modifié/retourné
testAddActor	Ajouter des acteurs	Acteurs correctement ajoutés à la liste
testMinimumAgeGetterSetter	Modifier/accéder à l'âge minimum	Âge minimum correctement modifié/retourné
testAvailabilityGetterSetter	Modifier/accéder à la disponibilité	Disponibilité correctement modifiée/retournée
testPosterGetterSetter	Modifier/accéder à l'affiche	Chemin d'affiche correctement modifié/retourné
testAddGenre	Ajouter des genres	Genres correctement ajoutés à la liste
testAddReview	Ajouter des avis	Avis correctement ajoutés à la liste
testEquals	Comparer des films	Égalité basée sur titre/année/réalisateur
testToString	Vérifier la représentation texte	Format "Titre (Année)" respecté

9. ReservationManagerTest

Objectifs des tests

Méthode de test	Actions	Attendu
testReset	Réinitialiser les réservations	Liste vide après reset
testGetReservations	Récupérer toutes les réservations	Liste complète retournée
testAddReservation	Ajouter une réservation	Réservation correctement ajoutée
testRemoveReservation	Supprimer une réservation	Réservation correctement retirée
testGetReservationsByUser	Filtrer par utilisateur	Liste des réservations de l'utilisateur
testGetReservationsByMovie	Filtrer par film	Liste des réservations du film

10. UserManagerTest

Objectifs des tests

Méthode de test	Actions	Attendu
testRegisterValidUser	Enregistrer un utilisateur valide	Enregistrement réussi (code 0)

Méthode de test	Actions	Attendu
testRegisterDuplicateUsername	Tenter d'enregistrer un nom d'utilisateur en double	Échec avec code 1
testRegisterEmptyCredentials	Tenter d'enregistrer avec identifiants vides/null	Échec avec code 2
testRegisterNullPersonallInfo	Tenter d'enregistrer avec infos personnelles null	Échec avec code 3
testLoginValidCredentials	Se connecter avec identifiants valides	Connexion réussie
testLoginInvalidCredentials	Se connecter avec identifiants invalides	Connexion échouée
testLogout	Se déconnecter après connexion	Déconnexion réussie
testLogoutWithNoCurrentUser	Se déconnecter sans être connecté	NotLoggedInException levée
testResetUsers	Réinitialiser la base utilisateurs	Base utilisateurs vidée

11. UserTest

Objectifs des tests

Méthode de test	Actions	Attendu
testConstructorAndGetters	Vérifier construction et getters	Attributs correctement initialisés
testSetters	Modifier tous les attributs	Nouvelles valeurs correctement enregistrées
testReservations	Ajouter/récupérer réservations	Réservations correctement gérées
testCancelReservation	Annuler une réservation	Réservation supprimée
testAddTooManyReservations	Dépasser limite réservations	RentingException levée
testAddAndGetReviews	Ajouter/récupérer avis	Avis correctement ajoutés
testMultipleReviews	Gérer plusieurs avis	Tous les avis correctement enregistrés
testUniqueReservations	Vérifier unicité réservations	Pas de doublons dans les réservations

12. ReviewTest

Objectifs des tests

Méthode de test	Actions	Attendu
testDefaultConstructor	Créer avis vide	Tous les attributs null/0

Méthode de test	Actions	Attendu
testParameterizedConstructor	Créer avis avec paramètres	Attributs correctement initialisés
testSetters	Modifier attributs d'un avis	Nouvelles valeurs correctement enregistrées
testEquals	Comparer des avis	Égalité basée sur utilisateur/film/note/commentaire

13. TestPersonallInformation

Objectifs des tests

Méthode de test	Actions	Attendu
testAge25Basic	Définir âge valide	Âge correctement enregistré
testNegativeAgeBasic	Définir âge négatif (info basique)	Âge négatif refusé
testNegativeAgeComplete	Définir âge négatif (info complète)	Âge original conservé
testAddressNotNull	Vérifier non-nullité adresse	Adresse jamais null
testSetterAddressNull	Tenter définir adresse null	Adresse originale conservée
testConstructor	Créer info avec paramètres invalides	Valeurs invalides corrigées