

Test de la classe Artiste :

Préparation du test :

- création d'un artiste de base et définition de ses attributs

Méthode de test	Actions	Attendus
testConstructeurEtGetters	<i>Vérifie la création d'un objet Artiste avec un constructeur et teste les méthodes getter associées (nom, prénom, nationalité, rôle).</i>	<i>Le nom doit être "Doe". Le prénom doit être "John". La nationalité doit être "Française". Le rôle doit être initialisé par défaut à Artiste.Role.ACTEUR.</i>
testSetters	<i>Modifie le nom, le prénom, et la nationalité d'un artiste à l'aide des setters.</i>	<i>Le nom doit être modifié en "Smith". Le prénom doit être modifié en "Jane". La nationalité doit être modifiée en "Anglaise".</i>
testChangeRoleValide	<i>Change le rôle de l'artiste vers des valeurs valides (REALISATEUR, BOTH, et ACTEUR) successivement.</i>	<i>Le changement de rôle doit réussir pour chaque valeur valide (la méthode change_Role doit retourner true). Le rôle de l'artiste doit être mis à jour avec succès.</i>

Méthode de test	Actions	Attendus
testConstructeurEtGetters	Vérifie la création d'un objet Artiste avec un constructeur et teste les méthodes getter associées (nom, prénom, nationalité, rôle).	Le nom doit être "Doe". Le prénom doit être "John". La nationalité doit être "Française". Le rôle doit être initialisé par défaut à Artiste.Role.ACTEUR.
testChangeRoleInvalide	Tente de changer le rôle de l'artiste avec une valeur invalide (null).	Le changement de rôle doit échouer (la méthode change_Role doit retourner false). Le rôle de l'artiste doit rester inchangé (valeur initiale Artiste.Role.ACTEUR).
testToString	Vérifie la méthode toString de l'objet Artiste.	La chaîne retournée par toString doit correspondre au format attendu : "Doe, John, Française, ACTEUR".

Test de la classe Evaluation :

Préparation du test :

Fonction setUp, créée avant chaque fonction de test un nouvel utilisateur et initialise ses attributs, un nouvel artiste et ses attributs, un nouveau film et ses attributs, une évaluation avec commentaire et une évaluation sans commentaire.

Méthode de test	Actions	Attendus
<u>testConstructeurAvecCommentaire</u>	<i>récupération du nom de l'auteur de l'évaluation récupération de la note de l'évaluation récupération de la chaîne de caractère correspondante au commentaire</i>	<i>le nom de l'auteur est égal au nom instancié au départ la note de l'évaluation est égale à la note instanciée au départ le commentaire est égal au commentaire instancié au départ</i>
<u>testConstructeurSansCommentaire</u>	<i>récupération du nom de l'auteur de l'évaluation récupération de la note de l'évaluation récupération de la chaîne de caractère correspondante au commentaire</i>	<i>le nom de l'auteur est égal au nom instancié au départ la note de l'évaluation est égale à la note instanciée au départ le commentaire est null</i>
<u>testGetNote</u>	<i>récupère le commentaire de l'évaluation avec commentaire récupère le commentaire de l'évaluation sans commentaire</i>	<i>le commentaire est égal au commentaire instancié au départ le commentaire retourné est nul car pas de commentaire</i>

Méthode de test	Actions	Attendus
<u>testGetCom</u>	<i>recupère le commentaire de l'évaluation avec commentaire recupère le commentaire de l'évaluation sans commentaire</i>	<i>le commentaire est égal au commentaire instancié au départ le commentaire retourné est nul car pas de commentaire</i>

Test de la classe Film :

Préparation du test :

Fonction setUp, crée 3 artistes, un réalisateur, et un nouveau film afin de faire des tests dessus.

Méthode de test	Actions	Attendus
<u>testConstructeurEtGetters</u>	<i> récupère : le nom du film le nom du réalisateur l'année de sortie du film l'âge minimum les acteurs le genre les évaluations l'état </i>	<i> Nom du film tel que set dans setUp Nom du réalisateur tel que dans setUp Année de sortie telle que dans setUp Âge minimum tel que dans setUp Acteurs vide Genre vide Evaluation vide Etat null </i>
<u>testSetters</u>	<i>set des valeurs aux différents attributs du film via les setters</i>	<i>attributs récupéré ayant la même valeurs que les valeurs insérées dans les setters</i>
<u>testActeurs</u>	<i>ajoute deux acteurs à la collections des acteurs participant au film</i>	<i>les acteurs doivent être présents dans la collection renvoyé en sortie la collection en sortie est de taille 2</i>
<u>testGenres</u>	<i>ajoute deux acteurs à la collection des genre correspondants au film</i>	<i>les genres sont présent dans la collection renvoyée en sortie la collection en sortie est de taille 2</i>
<u>testEtat</u>	<i>set l'état à true</i>	<i>l'état est à true</i>
<u>testEvaluations</u>	<i>ajoute deux évaluations au film</i>	<i>le nombre d'évaluation est 2</i>

Test de la classe GestionnaireArtiste :

Préparation du test :

Fonction setUp, crée un nouvel GestionnaireArtiste.

Méthode de test	Actions	Attendus
<u>testCreerArtiste_nominal</u>	<i>Crée un artiste avec des paramètres valides.</i>	<i>L'artiste est créé avec succès (non nul). Les attributs de l'artiste (nom, prénom, nationalité) sont correctement initialisés.</i>
<u>testCreerArtiste_nonNominal</u>	<i>Tente de créer des artistes avec des paramètres invalides :</i> <ul style="list-style-type: none">• <i>Valeurs nulles pour le nom, le prénom ou la nationalité.</i>• <i>Duplication d'un artiste déjà existant.</i>	<i>La création échoue (l'artiste retourné est null).</i>
<u>testEnsembleActeurs_nominal</u>	<i>Ajoute un artiste avec le rôle d'acteur et récupère l'ensemble des acteurs.</i>	<i>L'ensemble contient l'acteur ajouté.</i>
<u>testEnsembleActeurs_nonNominal</u>	<i>Vérifie l'ensemble des acteurs sans avoir ajouté d'acteurs.</i>	<i>L'ensemble est vide.</i>

Méthode de test	Actions	Attendus
<u>testEnsembleRealisateurs_nominal</u>	<i>Ajoute un artiste avec le rôle de réalisateur et récupère l'ensemble des réalisateurs.</i>	<i>L'ensemble contient le réalisateur ajouté.</i>
<u>testEnsembleRealisateurs_nonNominal</u>	<i>Vérifie l'ensemble des réalisateurs sans avoir ajouté de réalisateurs.</i>	<i>L'ensemble est vide.</i>
<u>testEnsembleArtiste_nominal</u>	<i>Ajoute un artiste et récupère l'ensemble des artistes.</i>	<i>L'ensemble contient un seul artiste. La taille de l'ensemble est égale à 1.</i>
<u>testEnsembleArtiste_nonNominal</u>	<i>Vérifie l'ensemble des artistes sans en avoir ajouté.</i>	<i>L'ensemble est vide.</i>
<u>testGetActeur_nominal</u>	<i>Ajoute un acteur à la liste des artistes et le récupère par son nom et prénom.</i>	<i>L'acteur retourné correspond à celui ajouté.</i>
<u>testGetActeur_nonNominal</u>	<i>Tente de récupérer un acteur avec des paramètres invalides ou inexistant.</i>	<i>Le résultat est null.</i>
<u>testGetRealisateur_nominal</u>	<i>Ajoute un réalisateur à la liste des artistes et le récupère par son nom et prénom.</i>	<i>Le réalisateur retourné correspond à celui ajouté.</i>
<u>testGetRealisateur_nonNominal</u>	<i>Tente de récupérer un réalisateur avec des paramètres invalides ou inexistant.</i>	<i>Le résultat est null.</i>

Méthode de test	Actions	Attendus
<u>testGetArtiste_nominal</u>	<i>Ajoute un artiste et le récupère par son nom et prénom.</i>	<i>L'artiste retourné correspond à celui ajouté.</i>
<u>testGetArtiste_nonNominal</u>	<i>Tente de récupérer un artiste avec des paramètres invalides ou inexistant.</i>	<i>Le résultat est null.</i>
<u>testSupprimerArtiste_nominal</u>	<i>Ajoute un artiste, le supprime, puis vérifie qu'il n'est plus dans la liste.</i>	<i>La suppression est réussie (retourne true). L'artiste ne figure plus dans la liste des artistes.</i>
<u>testSupprimerArtiste_nonNominal</u>	<i>Tente de supprimer un artiste inexistant ou un artiste nul.</i>	<i>La suppression échoue (retourne false).</i>

Test de la classe GestionnaireFilm :

Préparation du test :

Fonction setUp, crée un nouveau GestionnaireFilm.

Méthode de test	Actions	Attendus
<u>testCreerFilm</u>	<i>Crée un nouveau film en appelant la méthode creerFilm avec des paramètres valides. Tente de créer un film avec un titre déjà existant pour tester la duplication. Tente de créer un film avec un titre null pour vérifier la gestion des paramètres invalides.</i>	<i>Le film créé avec des paramètres valides ne doit pas être null et doit avoir le titre spécifié. La tentative de création d'un film en double doit retourner null. La création d'un film avec des paramètres invalides (comme un titre null) doit également retourner null.</i>
<u>testAjouterGenres</u>	<i>Ajoute un genre à un film existant. Tente d'ajouter un genre à un film inexistant.</i>	<i>Le genre doit être ajouté avec succès au film, et le film doit contenir ce genre dans sa liste de genres. L'ajout d'un genre à un film inexistant doit retourner false.</i>
<u>testAjouterAffiche</u>	<i>Ajoute une affiche à un film existant avec un chemin valide. Tente d'ajouter une affiche à un film inexistant.</i>	<i>Le chemin de l'affiche doit être correctement associé au film existant. Une exception IOException doit être levée lors de l'ajout d'une affiche à un film inexistant.</i>

Méthode de test	Actions	Attendus
<u>testSupprimerFilm</u>	<i>Supprime un film existant du gestionnaire. Tente de supprimer un film inexistant.</i>	<i>La suppression d'un film existant doit retourner true. La suppression d'un film inexistant doit retourner false.</i>
<u>testEnsembleFilms</u>	<i>Vérifie si l'ensemble des films dans le gestionnaire contient un film nouvellement créé.</i>	<i>L'ensemble des films doit inclure le film nouvellement ajouté.</i>
<u>testEnsembleFilmsRealisateur</u>	<i>Vérifie si l'ensemble des films associés à un réalisateur contient un film nouvellement créé.</i>	<i>L'ensemble des films du réalisateur doit inclure le film nouvellement ajouté.</i>
<u>testEvaluationMoyenne</u>	<i>Vérifie l'évaluation moyenne d'un film (évaluation fictive ou non initialisée ici).</i>	<i>Si aucun score d'évaluation n'est défini, l'évaluation moyenne doit être -1.0.</i>
<u>testOuvrirEtFermerLocation</u>	<i>Ouvre la location d'un film existant. Ferme la location du même film.</i>	<i>L'ouverture de la location doit retourner true, et l'état du film (etat) doit être true. La fermeture de la location doit retourner true, et l'état du film (etat) doit être false.</i>

Test de la classe GestionnaireUtilisateur:

Préparation du test :

Fonction setUp, crée un nouveau GestionnaireUtilisateur et un nouvel utilisateur, lui ajoute ses informations.

Méthode de test	Actions	Attendus
<u>testInscriptionNominal</u>	<i>Inscrit un nouvel utilisateur avec un pseudo, un mot de passe et des informations personnelles valides.</i>	<i>La méthode retourne 0, indiquant une inscription réussie.</i>
<u>testInscriptionPseudoDejaUtilise</u>	<i>Inscrit un utilisateur avec un pseudo déjà utilisé.</i>	<i>Le résultat de l'inscription doit être 1, indiquant que le pseudo est déjà utilisé.</i>
<u>testConnexionNominal</u>	<i>Inscrit un utilisateur, puis tente de se connecter avec le pseudo et le mot de passe corrects.</i>	<i>La connexion doit réussir et retourner true.</i>
<u>testConnexionPseudoInvalide</u>	<i>Tente de se connecter avec un pseudo incorrect.</i>	<i>La connexion doit échouer et retourner false.</i>
<u>testLouerFilmNominal</u>	<i>Inscrit un utilisateur, se connecte, puis loue un film disponible.</i>	<i>Le film doit être ajouté à la liste des films loués de l'utilisateur.</i>
<u>testLouerFilmDejaLoue</u>	<i>Inscrit un utilisateur, se connecte, loue un film, puis tente de louer le même film à nouveau.</i>	<i>Une exception LocationException doit être levée lorsqu'on tente de louer un film déjà loué.</i>

Méthode de test	Actions	Attendus
<u>testAjouterEvaluationNominal</u>	<i>Inscrit un utilisateur, se connecte, loue un film, puis ajoute une évaluation à ce film.</i>	<i>L'évaluation doit être ajoutée avec succès à la liste des évaluations du film.</i>
<u>testAjouterEvaluationFilmNonLoue</u>	<i>Inscrit un utilisateur, se connecte, et tente d'ajouter une évaluation à un film qu'il n'a pas loué.</i>	<i>Une exception LocationException doit être levée.</i>
<u>testModifierEvaluationNominal</u>	<i>Inscrit un utilisateur, se connecte, loue un film, ajoute une évaluation, puis modifie cette évaluation.</i>	<i>La nouvelle évaluation doit remplacer l'ancienne dans la liste des évaluations du film.</i>
<u>testModifierEvaluationNonExistante</u>	<i>Inscrit un utilisateur, se connecte, loue un film, et tente de modifier une évaluation inexistante.</i>	<i>Une exception LocationException doit être levée.</i>
<u>testFilmsEnLocation</u>	<i>Inscrit un utilisateur, se connecte, loue un film, puis récupère la liste des films en location.</i>	<i>La liste des films loués ne doit pas être nulle. Le film loué doit apparaître dans la liste.</i>

Test de la classe InfoPersonnelles:

Préparation du test :

Fonction setUp, crée une nouvelle info personnelle basique puis complète grâce à la basique.

Méthode de test	Actions	Attendus
<u>testAge25Basique</u>	<i>Modifie l'âge de l'instance infoBasique à 25.</i>	<i>La méthode getAge retourne 25, confirmant que la mise à jour de l'âge a bien été effectuée.</i>
<u>testAgeNegatifBasique</u>	<i>Tente de définir un âge négatif (-20) pour l'instance infoBasique.</i>	<i>L'âge de l'instance ne doit pas être négatif. La méthode getAge retourne une valeur différente de -20.</i>
<u>testAgeNegatifCompleet</u>	<i>Tente de définir un âge négatif (-20) pour l'instance infoComplete.</i>	<i>L'âge ne doit pas changer. La méthode getAge retourne la même valeur qu'avant l'appel à setAge.</i>
<u>testAdresseNonNull</u>	<i>Vérifie que les adresses des instances infoBasique et infoComplete ne sont pas nulles après leur création.</i>	<i>La méthode getAdresse ne retourne pas null pour les deux instances.</i>
<u>TestSetterAdresseNull</u>	<i>Tente de définir une adresse null pour l'instance infoComplete.</i>	<i>L'adresse de l'instance ne doit pas être définie sur null. La méthode getAdresse retourne une valeur non nulle.</i>

Méthode de test	Actions	Attendus
<u>testConstructeur</u>	<i>Vérifie le comportement du constructeur avec des paramètres non valides (adresse null, âge négatif).</i>	<i>Le nom est correctement initialisé à "Vador". Le prénom est correctement initialisé à "Dark". L'adresse est automatiquement remplacée par une valeur non nulle par défaut. L'âge est ajusté pour être une valeur valide (≥ 0).</i>

Test de la classe Utilisateurs:

Préparation du test :

Fonction setUp, crée une nouvelle info personnelle basique puis complète grâce à la basique, permet de creer un utilisateur, cree un realisateur, acteur et film.

Méthode de test	Actions	Attendus
<u>testConstructeurBasique</u>	<i>Vérifie le comportement du constructeur basique de la classe Utilisateur et ses valeurs par défaut.</i>	<i>Le pseudo est correctement initialisé ("lukeSkywalker"). Le mot de passe est correctement initialisé ("password123"). La collection des films est non nulle et vide.</i>
<u>testGetSetPseudo</u>	<i>Modifie le pseudo d'un utilisateur avec un setter.</i>	<i>Le pseudo est correctement modifié ("newLukeSkywalker").</i>
<u>testGetSetMdp</u>	<i>Modifie le mot de passe d'un utilisateur avec un setter</i>	<i>Le mot de passe est correctement modifié ("newPassword123").</i>
<u>testGetSetInfos</u>	<i>Modifie les informations personnelles d'un utilisateur avec un setter.</i>	<i>Le nom est correctement modifié ("Vador"). Le prénom est correctement modifié ("Dark"). L'adresse est correctement modifiée ("Planète Coruscant").</i>

Méthode de test	Actions	Attendus
<u>testAjouterRetirerFilm</u>	<i>Ajoute un film à la collection d'un utilisateur, puis le retire.</i>	<i>La collection des films contient le film après ajout. La collection des films est vide après retrait.</i>
<u>testSetFilms</u>	<i>Définit directement une collection de films pour un utilisateur.</i>	<i>La collection des films de l'utilisateur contient les films spécifiés.</i>
<u>testUtilisateurAvecFilms</u>	<i>Ajoute un film à l'utilisateur et vérifie la cohérence de la collection.</i>	<i>La collection des films n'est pas vide. La collection contient exactement un film.</i>
<u>testUtilisateurSansFilm</u>	<i>Vérifie qu'un utilisateur nouvellement créé n'a pas de films associés.</i>	<i>La collection des films est vide.</i>
<u>testUtilisateurSansInfos</u>	<i>Vérifie le comportement d'un utilisateur sans informations personnelles définies</i>	<i>Les informations personnelles de l'utilisateur sont nulles.</i>
<u>testSetMdpVide</u>	<i>Définit un mot de passe vide pour un utilisateur.</i>	<i>Le mot de passe de l'utilisateur est une chaîne vide.</i>
<u>testSetPseudoVide</u>	<i>Définit un pseudo vide pour un utilisateur.</i>	<i>Le pseudo de l'utilisateur est une chaîne vide.</i>
<u>testSetAdresseVide</u>	<i>Définit une adresse vide dans les informations personnelles d'un utilisateur.</i>	<i>L'adresse dans les informations personnelles de l'utilisateur est une chaîne vide.</i>
<u>testArtisteRealisateur</u>	<i>Change le rôle d'un artiste en réalisateur.</i>	<i>Le rôle de l'artiste est REALISATEUR. Le réalisateur du film a le rôle REALISATEUR.</i>

Méthode de test	Actions	Attendus
<u>testArtisteActeur</u>	<i>Change le rôle d'un artiste en acteur.</i>	<i>Le rôle de l'artiste est ACTEUR. Le film n'a pas encore d'acteurs.</i>
<u>testArtisteBothRoles</u>	<i>Change le rôle d'un artiste en les deux rôles (BOTH) et l'ajoute à un film.</i>	<i>Le rôle de l'artiste est BOTH. L'artiste est ajouté à la collection des acteurs du film.</i>
<u>testAssociationArtisteFilm</u>	<i>Associe un artiste à un film.</i>	<i>L'artiste est correctement associé à un film (afilm est vrai).</i>