RAPPORT DE TP : TP FINAL POO_2 2025

Contexte : Dans ce TP s'agissait de faire une application de Gestion des evenements avec des contraintes de technologie données par l'Enseignant

Implementation

Packages :

***jsonmanipulation** :
Ce package a servi pour la serialisation et deserialisation des fichiers json.
Dans ce package, nous avons les classes : OrganisateurSerialisation pour la serialisation des organisateurs, ConcertSerialisation pour les concerts, ConferenceSerialisation pour les conferences
On se passera du principe d'heritage pour la serialisation au vu de la complexite de configuration

***Evenements** : Ce package contient les classes Conference, Concert
***Controlleurs** : Ce packages contient tous les controlleurs des fichiers fxml qui servent d'interface
***Personne** : Pour les classes intervenant, participant et Organisateur
***test** : Pour les tests unitaires, ici on a implementé le test de serialisation de l'organisateur

Interface Utilisateur : JavaFx a ete utilisé via des fichiers fxml, chaque interface ayant son controlleurs pour respecter le principe SOLID

Tests unitares :

Objectif du test

Vérifier que la classe OrganisateurSerialisation est capable :

1. D'ajouter un objet Organisateur à un fichier JSON.
2. De relire correctement ce fichier pour reconstituer la liste des organisateurs.

Méthodes testées

- addOrganisateur(Organisateur org)
- getAllOrganisateur()

Préparation (Setup du test)

1. **Création d'un fichier temporaire** (File.createTempFile(...)) pour ne pas affecter les données réelles.

2. **Redirection de la classe OrganisateurSerialisation vers ce fichier** via une méthode fictive overrideFilePath(...), qui permettrait de tester sans écrire dans le fichier de production.
3. **Création d'une instance** de la classe OrganisateurSerialisation.
4. **Création d'un objet Organisateur** avec des valeurs connues (id, prénom, nom, mot de passe).
5. **Ajout de cet organisateur au fichier** via addOrganisateur(...).

Étapes du test

- Appeler la méthode addOrganisateur pour écrire un objet dans le fichier.
- Utiliser getAllOrganisateur pour relire la liste complète depuis le fichier.
- Vérifier les assertions suivantes :
    - La liste contient exactement **un** élément.
    - Les données de l'organisateur sont bien celles qui ont été ajoutées (ex. prénom = "Jean").

Résultat attendu

- Le fichier JSON contient une entrée valide.
- La lecture via getAllOrganisateur restitue fidèlement l'objet.
- Les attributs de l'objet (id, nom, prenom, motDePasse) sont correctement conservés.

Résultat obtenu

- Test **réussi sauf celui de suppression d'un organisateur n'yant pas encore été implémenté**.
- L'objet lu depuis le fichier correspond exactement à celui ajouté.

Conclusion

Ce rapport est juste un guide illustratif de l'implementation du TP et des details plus detaillées seront données dans une version améliorée