Documentation du projet Scala

Gestion de Bibliothèque

Étudiants :
Hiba Jabilou
Chaimae Rizki
Encadrant : **Mourad KAROUI**

Année universitaire 2023/2024

Pour réaliser le projet Bibliothèque, nous avons défini deux classes Livre et Bibliotheque et un objet interaction. Nous sauvegardons les données dans un fichier texte nommé Bibliothèque où chaque ligne désigne un livre défini comme suit : Titre, Auteur, année de publication, une variable booléene indiquant si le livre est emprunté ou pas.

1 Classe Livre

Elle comporte les attributs suivants :

— titre : Le titre du livre de type String.
— auteur : Le nom de l'auteur du livre sous format string.
— anneeDePublication : L'année de publication du livre sous formar entier.
— estEmprunté : Une variable booléenne pour savoir si le livre est emprunté ou non.

La classe Livre contient les méthodes suivantes.

— emprunter() : Cette fonction a pour but de permettre à l'utilisateur d'emprunter le livre. Nous gérons le cas où l'utilisateur tente d'emprunter un livre déjà emprunté, en recourant au bloc try, qui affiche le message d'erreur que le livre a été déjà emprunté grâce à la fonction throw exception.

  Si le livre a été emprunté avec succès, le résultat indique que l'emprunt a été réalisé sans erreurs. Sinon, le bloc failure est exécuté affichant le message d'erreur.
— Rendre(titre String) : Cette fonction permet de rendre un livre déjà emprunté. On gère le cas où le livre n'est pas déjà emprunté grâce à au bloc Try, qui affiche un message d'erreur. Après, nous gérons aussi les exceptions à l'aide du pattern matching. La variable result contient la sortie de la fonction. Si celle-ci est exécutée sans erreur, le bloc success est exécuté. Sinon, le bloc Failure est exécuté et un message d'erreur est affiché.
— getestEmprunté : Retourne l'état d'emprunt du livre (true ou false). C'est un getter qui nous permet d'accéder à la variable estemprunté en dehors de la classe.

2 Classe Bibliotheque

La classe Bibliotheque est constitué des méthodes suivantes.

— ajouterLivre(livre : Livre) : Ajoute un livre à la bibliothèque en prenant comme argument un objet de la classe Livre.
— emprunterLivre(titre : String) : Permet d'emprunter un livre de la bibliothèque par le titre. Nous gérons les exceptions grâce au pattern matching en recourant au bloc Some, qui permet d'emprunter le livre au cas où celui si existe. Sinon le bloc None est exécuté, et un message d'erreur indiquant que le livre n'est pas dans la bibliothèque est affiché.
— rendreLivre(titre : String) : Permet de rendre un livre emprunté en prenant comme argument le titre.
— rechercherParTitre(titre : String) : Recherche un livre en fonction du titre et retourne ses informations avec gestion des exceptions.
— rechercherParAuteur(auteur : String) : Recherche des livres en se basant sur le nom de l'auteur et retourne une liste de livres correspondants. Sinon, si l'auteur n'a aucun livre dans la bibliothèque, la fonction retourne None.
— getLivres : Retourne une liste contenant tous les livres disponibles dans la bibliothèque.

3 Object Interaction

Il permet à l'utilisateur d'interagir avec la bibliothèque en recourant à un menu textuel. Il est dotée des options suivantes :

— Ajouter un livre.
— Emprunter un livre.
— Rendre un livre.
— Rechercher un livre par titre.
— Rechercher un livre par auteur.
— Sauvegarder la bibliothèque dans un fichier.
— Quitter.

Le programme affiche un menu interactif où l'utilisateur peut choisir une action en entrant un numéro correspondant à l'option souhaitée. Si l'utilisateur donne comme entrée tout sauf un chiffre du menu, il invite l'utilisateur à réessayer.

L'utilisateur a l'option de sauvegarder l'état la bibliothèque à chaque moment

4 Test Unitaires

Les tests unitaires pour la classe ‘Livre‘ comprennent plusieurs scénarios pour l’emprunt et rendre.

Dans le premier test, nous vérifions que la méthode ‘emprunter‘ fonctionne correctement en empruntant un livre qui n’est pas déjà emprunté et en vérifiant que l’état ‘estEmprunté‘ est correctement mis à jour.

Dans le deuxième test, nous tentons d’emprunter un livre qui est déjà emprunté, et nous nous attendons à ce que cela échoue.

Dans le troisième test, nous vérifions que la méthode ‘rendre‘ fonctionne correctement en rendant un livre emprunté et en vérifiant que l’état ‘estEmprunté‘ est correctement mis à jour.

Enfin, dans le quatrième test, nous tentons de rendre un livre qui n’est pas emprunté, et nous nous attendons à ce que cela échoue.

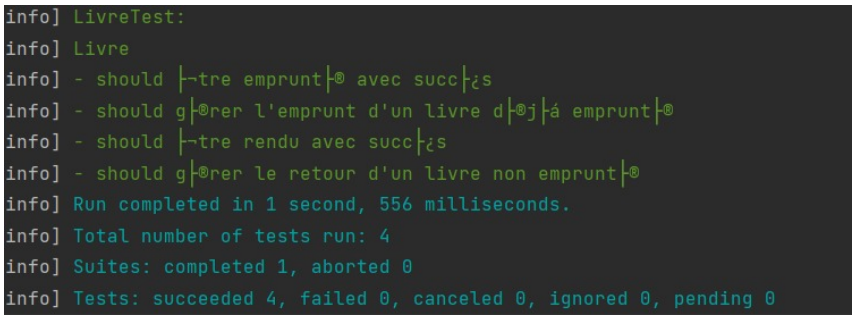

Figure 1 – Test Unitaires