

Rendu – Rust

Main :

Pour que le main reste actif tant que l'utilisateur le souhaite, j'utilise une boucle while utilisant un booléen qui reste vraie tant que l'utilisateur ne choisi pas l'option de quitter le programme

j'utilise également un match pour qu'il puisse choisir l'action à effectuer

Fonction add_book :

Pour ajouter des livres, l'utilisateur les saisies. Il y a des sécurités permettant qu'il ne puisse pas rentrer des dates absurdes et une condition vérifiant que le livre qu'il cherche à ajouter n'existe pas déjà.

Fonction borrow_book :

L'utilisateur saisi le nom du livre à chercher, puis le programme cherche le titre, s'il est trouvé, on vérifie s'il est disponible : s'il l'est on le rend indisponible est on l'indique à l'utilisateur, sinon on prévient juste l'utilisateur que le livre est déjà pris

Si le livre n'est pas trouvé, un booléen permet de l'indiquer à l'utilisateur

Fonction give_back :

La logique est la même que pour la fonction précédente, on inverse seulement la vérification de la disponibilité

Fonction print_all_book :

On commence par regarder la longueur du vecteur, si elle est à 0, on indique que la liste est vide. Sinon on parcourt le vecteur et on affiche à chaque fois le titre, l'auteur, et la date de publication.

Fonction print_disponible_book :

La logique est similaire à la fonction précédente, cependant on crée un second vecteur dans lequel on met uniquement les livres disponibles, cela permet d'utiliser la longueur du vecteur pour vérifier qu'il n'y a aucun livre disponible.