Master 1 informatique
UFR mathématique et informatique

Université Paris Cité

Programmation avancée - Projet machine

- Language: Python ou C++
- Platform: Moodle
- You should only provide the cpp file or the py file. This file must be named project_NAME.cpp or project_NAME.py, where NAME is your last name in caps (e.g. project_LOBRY.py)
- In C++, only including iostreams and string is authorized. In python, no import is authorized. Using another library will be considered as cheating.
- Using any generative tool (e.g. ChatGPT, co-pilot, …) will be considered as cheating. Using internet is authorized except for these tools.

L'objectif de ce projet est d'implémenter un logiciel permettant de gérer une bibliothèque classique (permettant d'emprunter des livres).

Attention, votre fichier principal (project_NAME.xx) doit pouvoir être interprété/compilé et montrer un cas d'utilisation de la bibliothèque.

Université Paris Cité

Partie 1 : Une bibliothèque basique

1.1 Livre

Dans cette première partie, nous allons nous intéresser à la modélisation des différents composants d'une bibliothèque. Vous devez d'abord proposer une structure de donnée pertinente permettant de modéliser un livre. Au minimum, vous devez sauvegarder pour un livre donné, les informations suivantes:

- son titre
- son auteur
- sa disponibilité

1.2 Adhérent

Vous devez ensuite modéliser un adhérent. Cette modélisation devra nécessairement se faire par la création d'une classe Member qui contiendra les informations suivantes:

- nom du membre
- numéro d'identifiant du membre
- les livres empruntés

De plus, vous devez gérer son affichage sur la sortie standard (soit sous forme de flux, soit lorsque l'objet est passé à print).

1.3 Bibliothèque

Vous devez écrire une classe Library qui permet de gérer les livres présents dans la bibliothèque et les adhérents inscrits.

Cette classe doit contenir les méthodes suivantes:

- add_book : prenant un livre en argument, et l'ajoute à la bibliothèque
- remove_book : prenant un livre en argument, et l'enlève de la bibliothèque. Vous devez gérer avec une exception le cas ou le livre n'était déjà pas présent
- list_books : ne prenant pas d'argument, liste tous les livres de la bibliothèque
- add_member : prenant un adhérent en argument et l'ajoute à la bibliothèque
- remove_member : prenant un adhérent en argument, et l'enlève de la bibliothèque. Vous devez gérer avec une exception le cas ou l'adhérent n'était déjà pas inscrit
- list_members : ne prenant pas d'arguments, liste tous les membres inscrits

1.4 Emprunts

Vous devez permettre de gérer les emprunts et retours de livres. L' implémentation est libre.

Université Paris Cité

Partie 2 : une bibliothèque utilisable

2.1 Sauvegarde

Vous devez modifier la classe Library afin de permettre de sauvegarder son état dans un fichier “libstate" qui se trouve à côté de votre programme.

2.2 Chargement

Vous devez ajouter la fonctionnalité de chargement du fichier que vous venez de créer. De plus, lorsque ce fichier existe, il doit être chargé automatiquement au démarrage du programme.

2.3 Interface

Vous devez créer une interface permettant de faire toutes les actions décrites ci-dessus depuis la ligne de commande.