

TP BIBLIOTHÈQUES

I - Nomenclature

Fichiers :

classes.py : PascalCase

autres.py : camelCase

.txt : camelCase

Code :

classes : PascalCase

attributs classe : snake_case

méthodes : PascalCase

fonctions : snake_case

variables : snake_case

II - Code :

Classes

Personne -> User - *Admin*

Library -> Item -> Livre -> BD - Comic

-> *Jeux*

-> *Film*

-> *CDAudio* -> *Music* - *AudioBook*

Personne :

Attributs :

- id (str)
- nom (str)
- prenom (str)
- mdp (str)

Méthodes :

-
-

User :

Attributs :

- emprunts [liste]
- grade (int)

grade_liste = [nivgrade qui correspond au nb max d'emprunt simultané, nombre d'emprunt pour atteindre le grade]

grade_liste = [[1, 0], [2, 5], [3, 10]]

grade{'nom_grade': [nb_max_emprunt, nb_emprunts_pour_atteindre_grade]}

grades : {'moldu': [1, 0], 'sang-de-bourbe': [2, 10], 'sorcier': [3, 20], 'Auror': [5, 40]}

Méthodes :

- ChangerMdp(self)
- Emprunt(self, livre)
- Rendre(self, livre)
- AfficherEmprunt(self)

(Pour aller plus loin :)

Admin :

Attributs :

- ?

Méthodes :

- AjouterStock
- RetirerStock
- Exporter (liste bibliotheque)
- RadierUser
- ProlongerEmprunt (d'un User)
- AttribuerEmprunt (à un User)

Bibliotheque :

Attributs :

- nom (str)
- rayon_liste [liste]
- auteur_liste [liste]
- livre_liste [liste]
- user_liste [liste]

Méthode :

- AfficherDispo(self) (return bool)
- Rechercher(self, ref) (return bool)
- InscriptionUser(self, nom, prenom, mdp) (+ enregistrement in bibliotheque.user_liste)
- ImporterLivre
- ExporterLivre
- ImporterUser
- ExporterUser

Livre

:Attributs :

- titre (str)
- auteur (str)
- langue (str)
- genre (str)

- categorie (str)
- ref (str)
- dispo (bool)
- retour (date en str)

Methodes :

- AfficherInfo(self)

BD :

Attributs :

- couleur (bool)
- dessinateur (str)

Méthodes :

- AfficherInfo(self) (rajouter couleur + dessinateur)

Fonctions :

- **check_int(data): (return bool)**
- check_dispo_livre(livre): (print("Non disponible") si déjà emprunté par catégorie cat/auteur/genre/langue sinon print info livre)
- **check_user(user): (return bool vérifie l'existence du user)**
- **check_user_mdp(user, mdp): (check le couple user + mdp return bool)**
- afficher_dispo_livre(livre_liste): (return les print de la méthode AfficherLivre de manière organisée)
- rechercher_livre(livre_liste, livre|cat|aut|genre) : (return print info livre qui correspondent)

Main :

- **boucle initiale**
- **Création menu général**
- Création sous-menu (boucle while avec champ libre pour appels méthodes en com)

Répartition:

- **Teddy** :- Main
 - fonctions
- **David** :- Personne
 - User
 - switch Bibliotheque après
- **Johan** :- Bibliotheque

- livres (BD)

Exemples de livres :

Harry Potter 1 ; J.K Rowling ; Français ; Fantastique ; Roman ; HJ135203 ; True ; None

Assassin Royal 1 ; R Hobb ; Anglais ; Fantastique ; Roman ; AR1940289 ; False ;

2021-02-27 Injustice 1 ; DC ; Français ; Fantastique ; BD ; ID104382 ; True ; None ; True ; J Raapack

Pour David

Id User

```
def IdUser(nom, prenom):  
  
    Id = str(nom[0]+"."+prenom)  
  
    return Id
```

Exemple de user :

p.maxime ; przybylo ; maxime ; Azerty77 ; [Objet livre Assassin Royal] ; 5

Question à poser à Maxime :

...

Pour aller ENCORE **ENCORE** plus loin:

- Génération mot de passe FULL aléatoire
- Ajouter couleur console sur les print et input (la carotte pour motiver les troupes !)
- Pour les grades, livres malus ! xD
- Grades en mode RPG, avec accès à certains rayon débloquable en fct du grade
- Intégrer built-in de sécurité de cryptage mdp
- Ajouter choisissez votre maison (Poudlard) quand grade sorcier atteint
- Classement des 4 Maisons

Pour aller **ENCORE ENCORE ENCORE** plus loin:

- Faire une fusée et aller dans l'espace !!