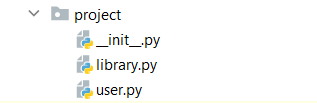
## Library\*

Create class called **Library**, where the information regarding the users and books rented/available will be stored. Create another one called **User,** where the information for each of the library users will be stored: user id, username, and **file** with **records of the books rented by the current user**.



### Class Library

In the **library.py** create class **Library**. Upon initialization it will not receive anything, but it should havethe following instance attributes:

* **user\_records** - empty list which will store the users (users objects) of the library
* **books\_available** - empty dictionary which will keep information regarding the authors (key: str) and the books available for each of the authors (list of strings)
* **rented\_books** - empty dictionary which will keep information regarding the usernames (key: str) and nested dictionary as value in which will keep information regarding the book names (key: str) and days left before returning the book to the library (int) - (**{usernames: {book names: days to return}})**.

You should also create **3 instance methods**:

* **add\_user(user: User)**:
  + Add the user if we do not have him/her in the library's user records already
  + Otherwise, return the message **"User with id = {user\_id} already registered in the library!"**
* **remove\_user(user: User):**
  + Remove the user from the library records if we have him/her in the library's user records
  + Otherwise, return the message **"We could not find such user to remove!"**
* **change\_username(user\_id: int, new\_username: str):**
  + Change the username with the new provided and return the message **"Username successfully changed to: {new\_username} for userid: {user\_id}"** if there is a record with the same user id in the library and the username is different than the provided one. Change his username in the **rented\_books** dictionary as well (if present).
  + If the new username is the same for this id return the following message **"Please check again the provided username - it should be different than the username used so far!"**.
  + If there is no record for the provided id return **"There is no user with id = {user\_id}!"**

### Class User

In the **user.py** create class **User**. Upon initialization it should receive **user\_id** (int) and **username** (string). The class should also havean instance attribute **books** which will be an empty list at the beginning. You should also create **3 instance methods**:

* **get\_book(author: str, book\_name: str, days\_to\_return: int, library: Library)**:
  + if the **book is available** in the library add it to the **books list** for this user, **update the library records (rented\_books and available\_books dicts)** and return the following message: **"{book\_name} successfully rented for the next {days\_to\_return} days!"**
  + if it is **already rented** return the following message **"The book "{book\_name}" is already rented and will be available in {days\_to\_return provided by the user rented the book} days!"**
* **return\_book(author:str, book\_name:str, library: Library):**
  + if the **book is in the user's books list return it in the library** (update **books\_available and rented\_books** class attributes) and **remove it from the books list** for this user
  + otherwise **return** the following message **"{username} doesn't have this book in his/her records!"**
* **info()** - **return** a string containing the books currently rented by the user in ascending order separated by comma and space.
* You should also override the **\_\_str\_\_** method to get a string in the following format **"{user\_id}, {username}, {list of rented books}"**

|  |
| --- |
| **Test Code** |
| from project.library import Library from project.user import User  user = User(12, 'Peter')  library = Library()  library.add\_user(user)  print(library.add\_user(user))  library.remove\_user(user)  print(library.remove\_user(user))  library.add\_user(user)  print(library.change\_username(2, 'Igor'))  print(library.change\_username(12, 'Peter'))  print(library.change\_username(12, 'George'))  [print(f'{user\_record.user\_id}, {user\_record.username}, {user\_record.books}') for user\_record in library.user\_records]  library.books\_available.update({'J.K.Rowling': ['The Chamber of Secrets',  'The Prisoner of Azkaban',  'The Goblet of Fire',  'The Order of the Phoenix',  'The Half-Blood Prince',  'The Deathly Hallows']})  user.get\_book('J.K.Rowling', 'The Deathly Hallows', 17, library)  print(library.books\_available)  print(library.rented\_books)  print(user.books)  print(user.get\_book('J.K.Rowling', 'The Deathly Hallows', 10, library))  print(user.return\_book('J.K.Rowling', 'The Cursed Child', library))  user.return\_book('J.K.Rowling', 'The Deathly Hallows', library)  print(library.books\_available)  print(library.rented\_books)  print(user.books) |
| **Output** |
| User with id = 12 already registered in the library!  We could not find such user to remove!  There is no user with id = 2!  Please check again the provided username - it should be different than the username used so far!  Username successfully changed to: George for userid: 12  12, George, []  {'J.K.Rowling': ['The Chamber of Secrets', 'The Prisoner of Azkaban', 'The Goblet of Fire', 'The Order of the Phoenix', 'The Half-Blood Prince']}  {'George': {'The Deathly Hallows': 17}}  ['The Deathly Hallows']  The book "The Deathly Hallows" is already rented and will be available in 17 days!  George doesn't have this book in his/her records!  {'J.K.Rowling': ['The Chamber of Secrets', 'The Prisoner of Azkaban', 'The Goblet of Fire', 'The Order of the Phoenix', 'The Half-Blood Prince', 'The Deathly Hallows']}  {'George': {}}  [] |
| **Test Code** |
| from library import Library  from user import User  user = User(12, 'Peter')  library = Library()  library.add\_user(user)  library.books\_available.update({'J.K.Rowling': ['The Chamber of Secrets',  'The Prisoner of Azkaban',  'The Goblet of Fire',  'The Order of the Phoenix',  'The Half-Blood Prince',  'The Deathly Hallows']})  user.get\_book('J.K.Rowling', 'The Deathly Hallows', 10, library)  print(user) |
| **Output** |
| 12, Peter, ['The Deathly Hallows'] |