CW - OOP – Library

Let's model the central library of our city! :)

Library Class

The Library class represents a library and should have properties specific to your model. Here are a few suggestions to consider:

- Add properties such as `name`, `location`, and `opening_hours` to provide more details about the library.
- Implement getter and setter methods for each property to ensure proper encapsulation.
- Consider adding validation to the setter methods, for example, to check if the provided values are of the correct type or format. If the validation fails, raise an error to notify the user.

Shelves

The shelves in the library have some specific characteristics. Here are a few ideas to improve the shelf implementation:

- You can create a Shelf class that inherits from the list class. This way, you can customize its behavior and add additional methods as needed.
- Add a maximum capacity property to each shelf, limiting the total number of pages that can be placed on it. Implement a mechanism to enforce this limit.
- Assign a unique non-repeating identifier to each shelf. You can combine the category ID and shelf number, as you mentioned, to create a unique identifier for each shelf.
- Some libraries categorize their books based on their category(ficion, action, philosophy, etc.) if every category has its own unique identifier for example 9923 then the combination of that id with the shelf number will be the id of the shelf we are creating => `1 – 9923`

  [shelf number] – [category id]

```
B1, b2= Book(page_numer=100) , Book(page_numer=200)
s = Shelf()
s.add_book(b1, b2)
>>>len(s)
300
```

Librarian

The librarian plays a crucial role in managing the library and assisting patrons. Let's enhance the functionality of the Librarian class:

- Implement a method to search for books based on various criteria such as name, ID, author, or category. Return a list of found books in a table view format, including details like book name, shelf, author, and book ID.
- Develop a method to retrieve a book using its ID.
- Allow the librarian to add new categories to the library's collection. Handle errors gracefully if a duplicate category is attempted to be added.
- Enable the librarian to search for a book and place it on a desired shelf using the shelf label and book ID, name, or author info.

Category

The Category class represents different book categories. Consider the following improvements:

- Allow only the librarian to add new categories to ensure proper control.
- Implement the necessary magic methods, such as `__len__`, to return the number of books or pages in a specific genre.
- Ensure that duplicate categories cannot be added, and handle any errors appropriately.

```
c = librarian1.add_category('action')
b1 = Book(page_number=100, category=c)
b2 = Book(page_number=100, category=c)
len(c) # 200
```

Book

The Book class represents individual books in the library. Here are a few suggestions:

- Make certain book information mandatory and unchangeable, such as page count, unique ID, price, category, and author.
- Utilize the Author class to define the author-related values for each book, promoting code reusability through composition.

General Recommendations

In addition to the specific improvements mentioned above, keep in mind the following principles:

- Try adding unique id to your models when necessary so you can retrieve them by their ID
- Utilize the concepts of mixin, abstraction, encapsulation, polymorphism, and inheritance as appropriate within your implementation.
- Apply composition where necessary, such as when defining the values related to the author of each book using the Author class.
- Creativity counts! Feel free to introduce additional features and libraries you have encountered in other projects to make your exercise more creative and challenging.

Challange

Consider implementing a FileHandler class to automate reading and writing pickle files for storing your objects. This can serve as a database for your library. (here is a guide to python's pickle: https://snyk.io/blog/guide-to-python-pickle/)