

Библиотека

Создать систему управления библиотекой. Библиотека состоит из отделов (например, отдел художественной литературы, отдел научной литературы и т.д.). В каждом отделе есть свой набор книг. Книги могут перемещаться между отделами. Читатели могут брать книги в отделе и возвращать их обратно.

Решение:

```
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author
        self.status = "available"

    def take(self):
        self.status = "taken"

    def return_back(self):
        self.status = "available"

class Library:
    class Department:
        def __init__(self, name):
            self.name = name
            self.books = []

        def add_book(self, book: Book):
            self.books.append(book)

        def remove_book(self, book: Book):
            self.books.remove(book)

        def __str__(self):
            result = f"Department: {self.name}\n"
            result += "Books:\n"
            for b in self.books:
                result += f"\t{b.title} by {b.author}, status: {b.status}\n"
            return result

    def __init__(self, name):
        self.name = name
        self.departments = {}

    def add_department(self, department):
        self.departments[department] = (self.__class__.Department(department))

    def move_book(self, book: Book, department_from, department_to):
        department_from.remove_book(book)
        department_to.add_book(book)

    def __str__(self):
        result = f"Library: {self.name}\n"
        result += "Departments:\n"
        for department in self.departments:
            result += f"\t{department}\n"
            for b in self.departments[department].books:
                result += f"\t\t{b.title} by {b.author}, status: {b.status}\n"
        return result
```

Здесь класс `Book` представляет книгу, класс `Department` представляет отдел библиотеки, а класс `Library` представляет библиотеку в целом. Класс `Book` содержит методы `take()` и `return_back()`, которые позволяют изменять статус книги.

Класс `Department` содержит методы `add_book()` и `remove_book()`, которые позволяют добавлять и удалять книги из отдела. Класс `Library` содержит метод `move_book()`, который позволяет перемещать книги между отделами.

Пример работы кода:

```
book1 = Book("1984", "George Orwell")
book2 = Book("Brave New World", "Aldous Huxley")

department1 = "Science Fiction"
department2 = "Dystopian Literature"

library = Library("New York Public Library")
library.add_department(department1)
library.add_department(department2)

library.departments[department1].add_book(book1)
library.departments[department1].add_book(book2)

print(library)

library.move_book(
    book1,
    library.departments[department1],
    library.departments[department2]
)

print(library)
```

Здесь мы создаем две книги, два отдела и одну библиотеку. Затем добавляем книги в первый отдел и отделы в библиотеку. Выводим содержимое библиотеки на экран. Затем перемещаем первую книгу из первого отдела во второй отдел и снова выводим содержимое библиотеки на экран.

В этом решении мы используем агрегацию, так как книги являются частью отдела, но при этом могут существовать и взаимодействовать независимо от него. Также мы используем композицию, так как отделы являются частью библиотеки и не могут существовать без неё.