

Лабораторна робота №2.

Тема. Проектування та розроблення простих класів в Python.

Мета. Ознайомлення з об'єктно-орієнтованим проектуванням та отримання навиків по реалізації класів в Python.

Хід роботи.

1. Розглянути приклад проектування системи електронного обліку в бібліотеці.

Короткий опис предметної області:

Сучасні бібліотеки використовують електронні каталоги для обліку книжок та пошуку в бібліотечних фондах. Потрібно розробити такий електронний каталог. Каталог містить перелік книжок. Користувачі повинні мати можливість здійснювати пошук за галузями знань, назвами та авторами.

Попередній аналіз:

Книжки однозначно ідентифікуються за International Standard Book Number (ISBN). Також кожна книжка має децимальний номер згідно Dewey Decimal System (DDS), який дозволяє розділити книжки за галузями знань. Попередній аналіз дозволяє встановити, що об'єктами в такій системі будуть *Book* з атрибутами *author*, *title*, *subject*, *ISBN*, та *DDS number*. Також зрозуміло, що каталог повинен забезпечувати пошук. Результат попереднього аналізу можна представити за допомогою наступної діаграми класів, в якій для представлення об'єктів системи призначені класи **Book**, **Author**, **Catalog**:

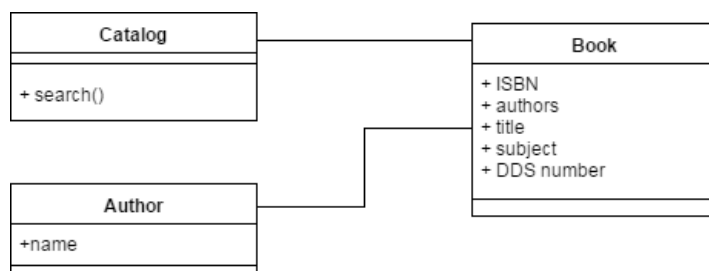


Рис.1. Попередня діаграма класів

Проектування системи:

Подальший аналіз при проектуванні системи показує, що у сучасній бібліотеці зберігаються не тільки книжки а також газети, журнали, DVD та CD диски. Журнали ідентифікуються за назвам, томами і номерами. DVD ідентифікуються за акторами, режисерами та жанрами а CD диски за авторами. Результатом більш докладного аналізу буде наступна діаграма класів:

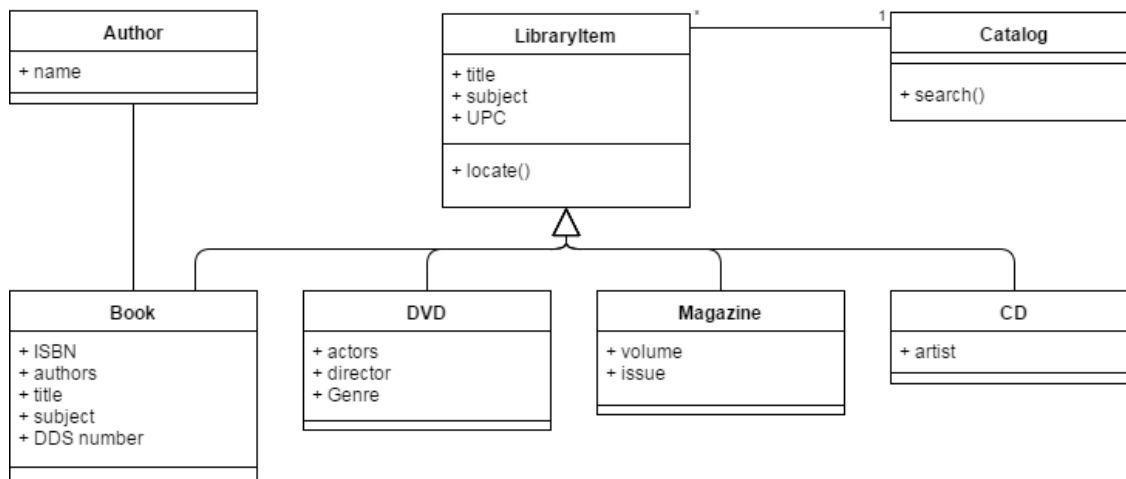


Рис.2. Діаграма класів (варіант 1)

В цій діаграмі використовується зв'язок успадкування між класами **Book**, **DVD**, **Magazine**, **CD** та **LibraryItem** для більш зручної організації бібліотечних фондів. Успадкування доцільно використовувати оскільки всі одиниці з бібліотечних фондів володіють багатьма спільними атрибутами але і мають суттєві відмінності які дозволяють встановити їх місцезнаходження. Для книжок це DDS номер а для інших це UPC номер.

Діаграма класів, яка показана на рис.2 має суттєвий недолік. В цій діаграмі розділені автори книжок та актори і автори CD та DVD дисків. Для авторів книжок передбачено окремий клас а актори і автори CD та DVD дисків це атрибути класів хоча ці всі об'єкти використовуються для представлення людей. Для врахування цього пропонується змінити діаграму і додати окремий клас для всіх осіб, які зробили певний внесок в ту чи іншу одиницю бібліотечного фонду.

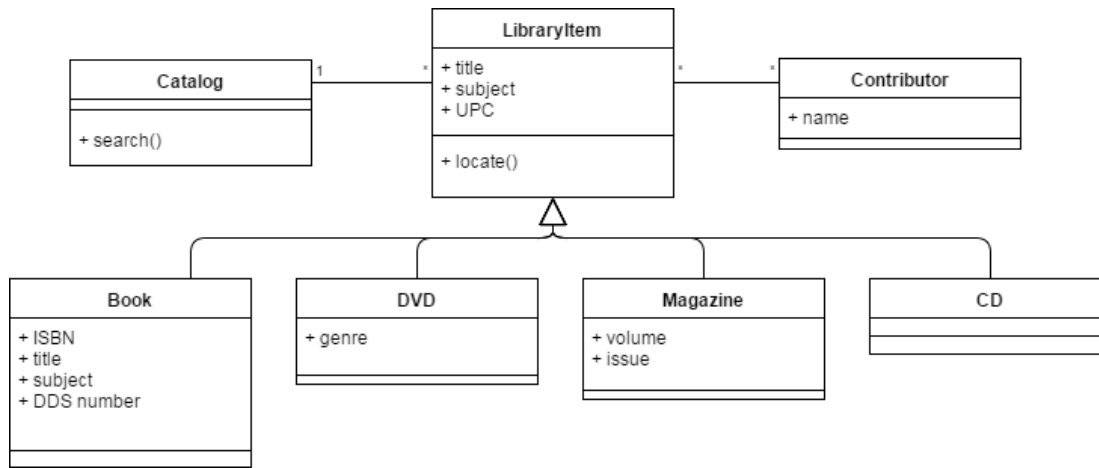


Рис.3. Діаграма класів (варіант 2)

Зміни в діаграмі класів (рис.3) дозволяють врахувати те що як DVD так і книга можуть мати декілька осіб, які їх створювали але разом з тим втрачено інформацію про те, який саме внесок вони зробили. Можна звичайно спробувати це врахувати встановивши інші зв'язки асоціації між класами та додавши метод для виявлення ролі автора. Результат представлено на наступному рисунку:

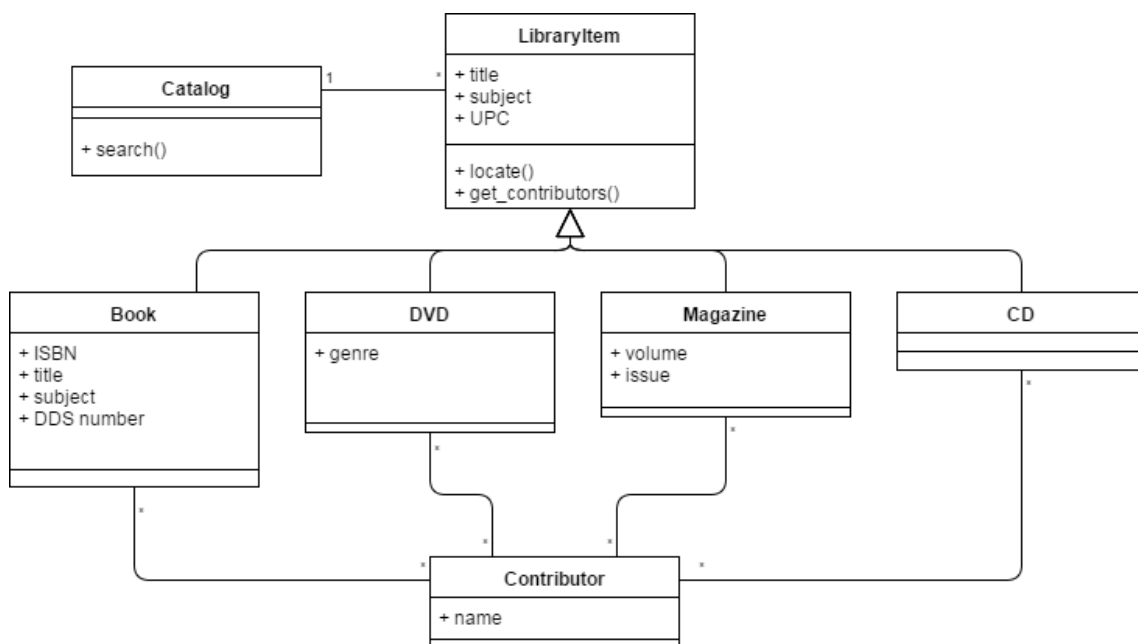


Рис.4. Діаграма класів (варіант 3)

Незважаючи на те, що вдалося позбутися згаданого вище недоліку, система, яка представлена цією діаграмою класів стала доволі складною і для розуміння і для керування і для розширення. Правильним способом модифікації діаграми для врахування ролі кожної особи буде здійснення рефакторингу системи. До системи додається окремий клас для композиції всіх осіб та для ідентифікації їх ролі.

Після проведення проектування системи діаграма класів буде мати такий вигляд:

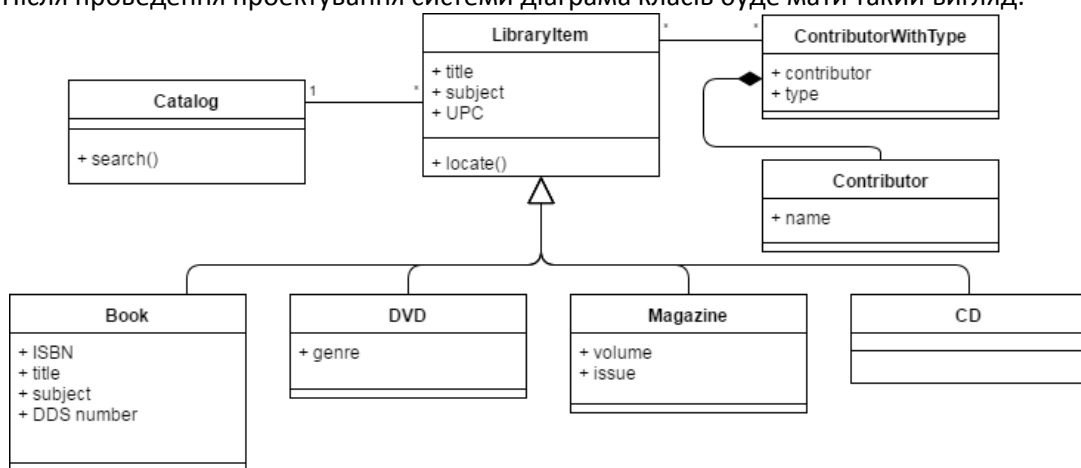


Рис.5. Діаграма класів (остаточний варіант перед реалізацією).

2. Розглянути приклад реалізації класів згідно діаграми класів для розроблення програми – Записник (Notebook).

Записник(Notebook) це програма для створення та роботи з щоденними записами. Запис це короткий текст, який зберігається в Notebook. Кожен запис крім тексту містить дату його створення та може містити тег для організації пошуку в записнику. Користувач повинен мати можливість не тільки здійснювати пошук записів а також і модифікувати їх.

Діаграма класів (рис.6) пояснює наступні елементи:

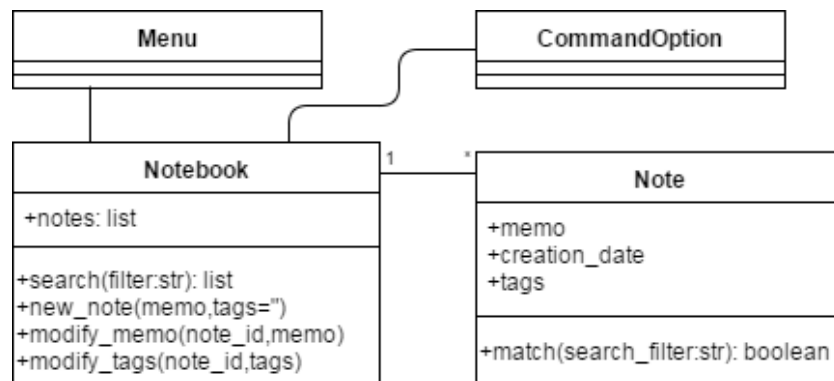


Рис.6. Діаграма класів для програми Notebook.

3. Розглянути реалізацію програми Notebook (див. Літературу). Дослідити працездатність цієї програми та пояснити за допомогою функцій isinstance, dir, наступні поняття: об'єкт, атрибути класу, методи класу, self, метод __init__, метод __str__.

4. Завдання лабораторної роботи для виконання в аудиторії: Здійснити попередній аналіз та встановити об'єкти їх атрибути та поведінки. Розробити діаграму класів для класів Ship, Field, Game, Player з проекту "Морський бій". Діаграму класів представити у файлі довільного графічного формату з іменем see_battle_Прізвище студента.* (наприклад, see_battle_Romanyuk.jpg)

5. Завдання лабораторної роботи: Реалізувати класи згідно заданої діаграми класів та їх опису.

Література:

1. Python 3 Object-oriented Programming Second Edition , Dusty Phillips, 2015
Chapter 1. Object-oriented Design
Chapter 2. Objects in Python