

Zadanie: "System Zarządzania Biblioteką - Obiektowe podejście w Pythonie"

Cel zadania:

Celem ćwiczenia jest praktyczne wykorzystanie paradygmatów programowania obiektowego w Pythonie, ze szczególnym uwzględnieniem:

- projektowania klas i dziedziczenia,
 - hermetyzacji danych przy użyciu dekoratora @property,
 - pracy z kolekcjami obiektów,
 - implementacji podstawowych mechanizmów OOP: enkapsulacji, dziedziczenia i polimorfizmu.
-

Opis problemu:

Twoim zadaniem jest zaprojektowanie i zaimplementowanie systemu zarządzania biblioteką, który pozwala na rejestrowanie książek oraz zarządzanie listą wypożyczeń przez użytkowników. System ma działać w modelu obiektowym.

Wymagania funkcjonalne:

1. Klasa Book

- Atrybuty:
 - title (str) – tytuł książki,
 - author (str) – autor książki,
 - year (int) – rok wydania,
 - available (bool) – informacja, czy książka jest dostępna.
- Zastosuj @property do kontrolowanego odczytu i modyfikacji atrybutu available.
- Zaimplementuj metodę __str__(), która zwróci czytelny opis książki.

2. Klasa User

- Atrybuty:
 - name (str) – imię i nazwisko użytkownika,
 - borrowed_books (kolekcja) – lista wypożyczonych książek.
- Metody:
 - borrow_book(book: Book) – umożliwia wypożyczenie książki (jeśli jest dostępna),
 - return_book(book: Book) – umożliwia zwrot książki,
 - list_borrowed_books() – wypisuje listę wypożyczonych książek.

3. Klasa Library

- Atrybuty:
 - books (kolekcja) – lista wszystkich książek w bibliotece.
- Metody:
 - add_book(book: Book) – dodaje nową książkę do zasobów biblioteki,
 - search_by_author(author: str) – wyszukuje i zwraca książki danego autora,
 - search_by_title(title: str) – wyszukuje książki po tytule (pełnym lub częściowym),
 - list_available_books() – wyświetla wszystkie dostępne książki.

Wymagania techniczne:

Zaimplementuj mechanizmy hermetyzacji przy pomocy dekoratora @property.

Używaj kolekcji do przechowywania obiektów (list, set lub dict, uzasadnij wybór).

Pokaż działanie polimorfizmu (np. nadpisanie metody __str__()), ew. rozważ rozszerzenie klasy Book o inne typy publikacji np. Magazine, Ebook).

Zapewnij odpowiednią walidację danych przy wypożyczaniu i zwrocie książek.

Efekt końcowy:

Stwórz skrypt Python, który:

- Zademonstruje utworzenie przykładowych książek i użytkowników,
- Dodanie książek do biblioteki,
- Obsługę wypożyczeń i zwrotów,
- Prezentację wyników działania metod wyszukiwania i listowania zasobów.

Kod powinien być przejrzysty, dobrze udokumentowany, z zachowaniem zasad OOP.

Zadanie dodatkowe (dla chętnych):

Rozszerz klasę Library o mechanizm zliczania wypożyczeń i raportowania, które książki są najczęściej wypożyczane.