**Ministerul Educaţiei și Cercetării al Republicii Moldova Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Laboratory work 1:**

Creational Design Patterns

# Elaborated:

st. gr. FAF-222 Șarov Andrei

# Verified:

asist. univ. Furdui Alexandru

# Chişinău - 2024

**Objectives:**

* Study and understand the Creational Design Patterns.
* Choose a domain, define its main classes/models/entities and choose the appropriate instantiation mechanisms.
* Use some creational design patterns for object instantiation in a sample project.

**Main Tasks:**

* Choose an OO programming language and a suitable IDE or Editor (No frameworks/libs/engines allowed).
* Select a domain area for the sample project.
* Define the main involved classes and think about what instantiation mechanisms are needed.
* Based on the previous point, implement atleast 3 creational design patterns in your project.

**Used Design Patterns:**

* **Factory Method:** Used in UserFactory to create different types of users (MemberUser, AdminUser), enabling flexibility to add more user types.
* **Singleton:** Ensures only one instance of LibraryManager manages library data, preventing conflicts from multiple instances.
* **Builder:** Used to construct complex Book objects with optional attributes, making book creation flexible and structured

**Implementation:**

**Define Classes**:

* Created a base User class and subclasses (MemberUser, AdminUser) for different user types.
* Implemented UserFactory to handle user creation logic based on type.

**Implement Singleton**:

* Designed LibraryManager to manage a single instance, overseeing lists for books and users.

**Build Book Objects**:

* Developed a Book class for essential attributes and a BookBuilder class for step-by-step attribute setting.
* Used the builder to create Book instances with specific details.

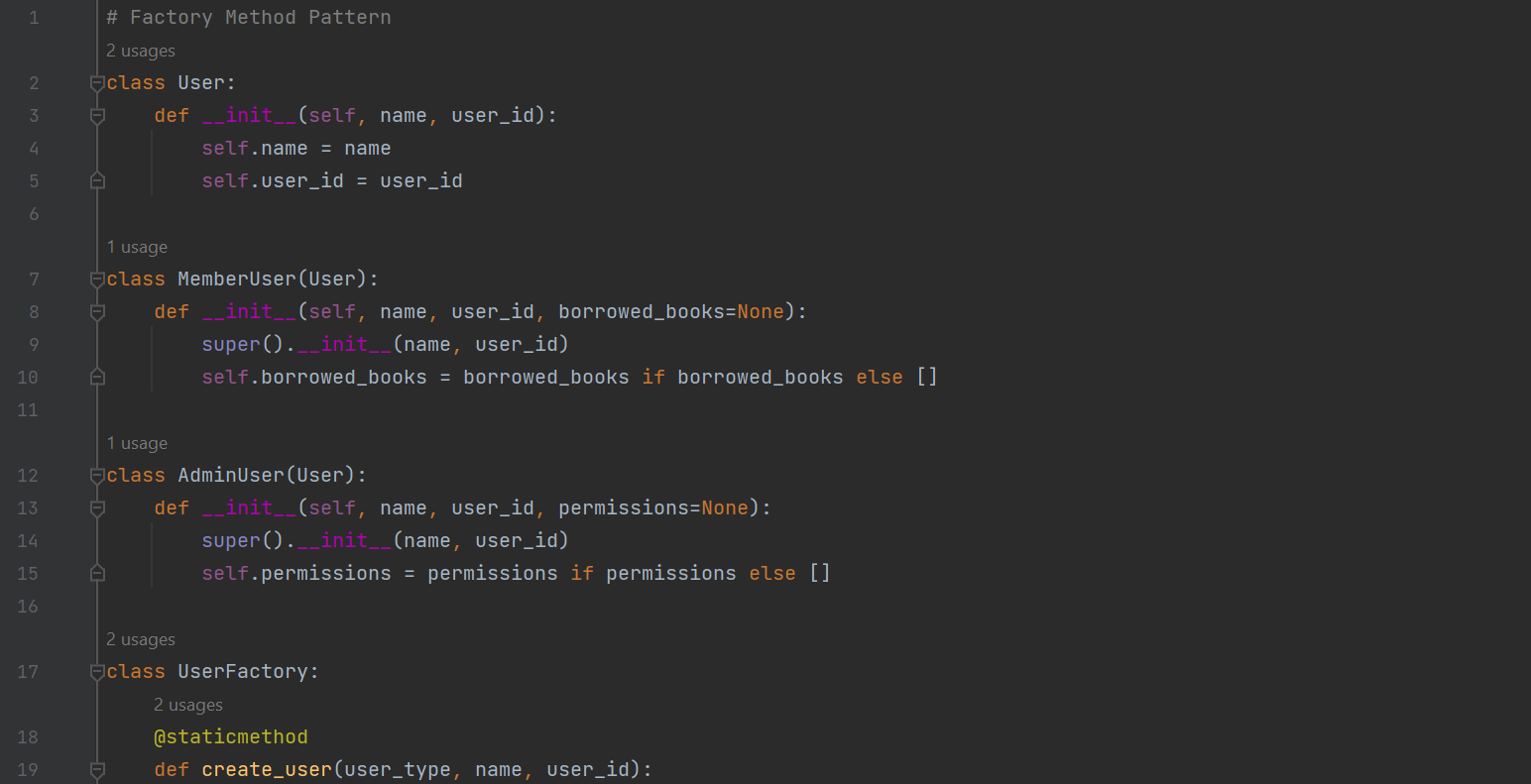
**Usage**:

* Demonstrated functionality by creating users with the factory, managing them through the singleton, and constructing books with the builder.

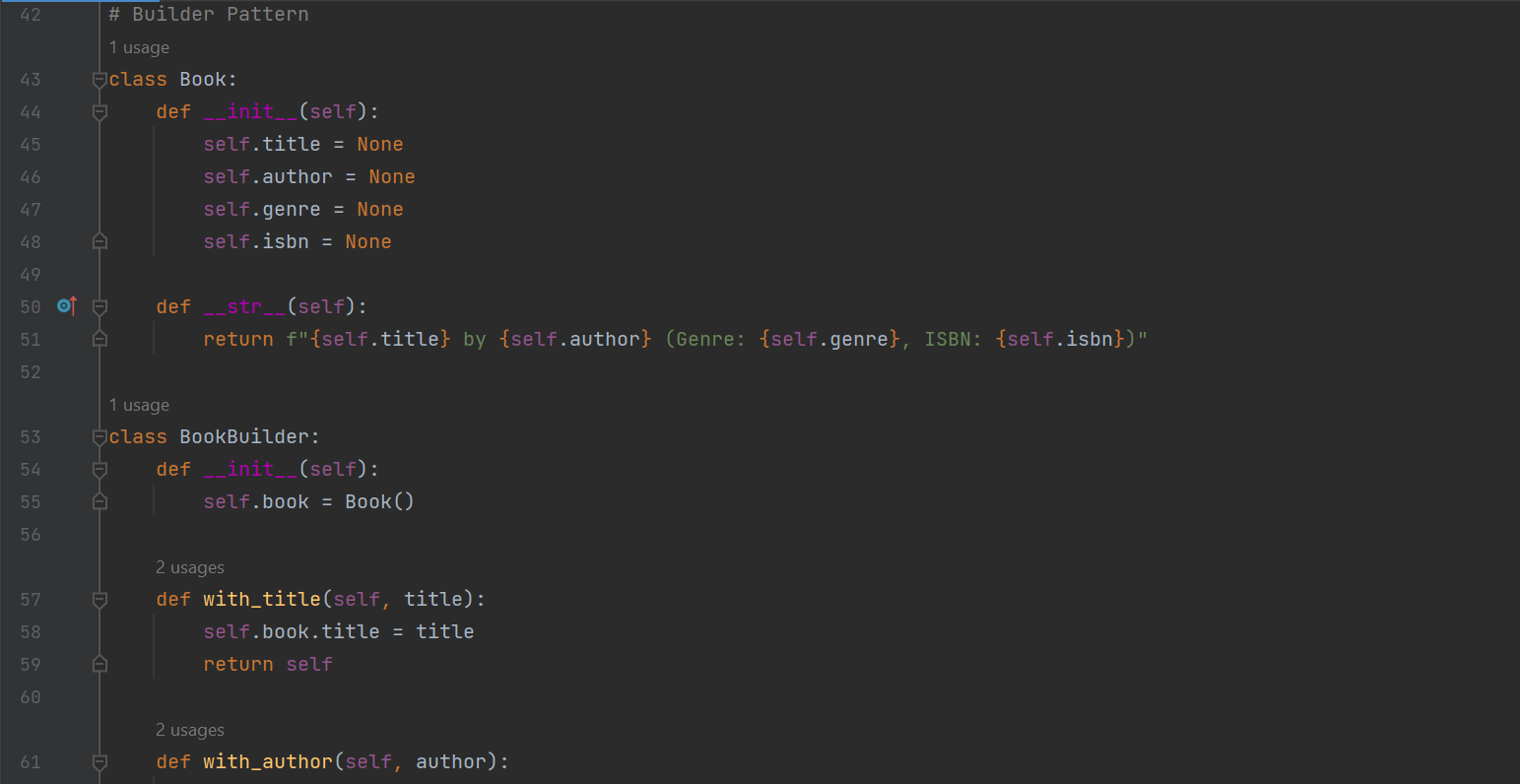
**Conclusions:**

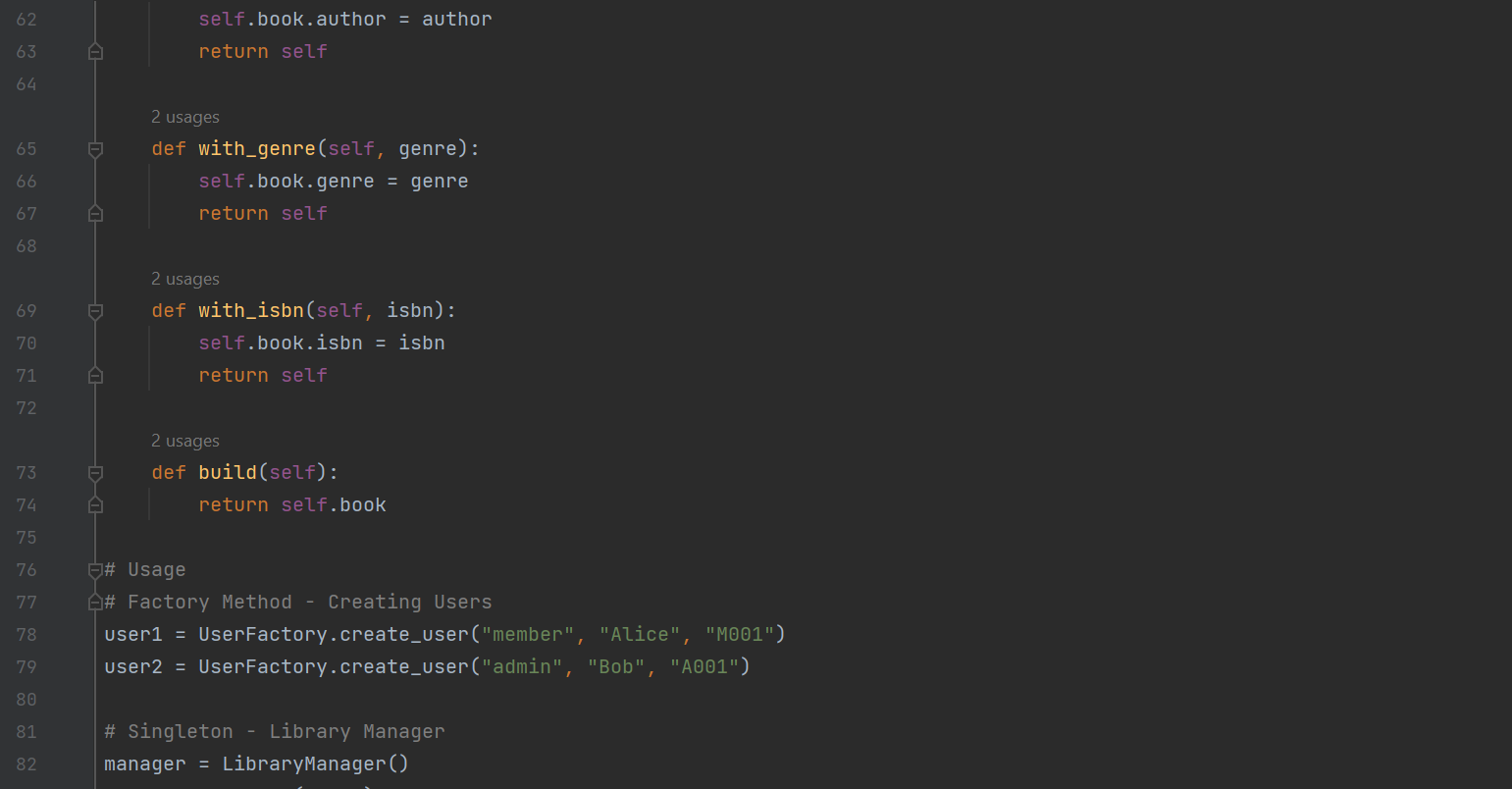
This laboratory successfully illustrated the application of creational design patterns to improve code organization and extensibility in a Library Management System. By employing the Factory Method, the system can easily accommodate new user types without modifying existing code. The Singleton pattern ensures consistent management of library data through a single point of access, while the Builder pattern streamlines the creation of complex book objects, enhancing readability and maintainability. Overall, these patterns contributed to a well-structured and scalable design, meeting the objectives of the laboratory.

**Code:**

****







**Results:**

