**Курс:** Шаблоны проектирования приложений

**Тема:** Модуль 11 Диаграммы вариантов использования. Диаграмма классов

*Баллы:*

**Цель работы:**

Разработка системы управления библиотекой с учетом принципов объектно-ориентированного программирования и проектирования, с использованием диаграммы классов и последующей реализации на языке C#.

**Описание задания:**

Необходимо спроектировать и реализовать систему управления библиотекой, которая включает следующие основные функции:

1. Управление пользователями библиотеки (читатели, библиотекари).
2. Управление книгами (добавление, редактирование, удаление, поиск книг).
3. Управление операциями выдачи и возврата книг.
4. Отчеты по использованию библиотеки (популярность книг, активность читателей и т.д.).

**Этапы выполнения работы:**

**1. Проектирование диаграммы классов:**

Создайте диаграмму классов, отражающую основные сущности и их взаимосвязи. Основные сущности:

* **Book** (Книга)
* **User** (Пользователь)
  + **Reader** (Читатель)
  + **Librarian** (Библиотекарь)
* **Loan** (Выдача книги)
* **Library** (Библиотека)
* **Author** (Автор)
* **Report** (Отчеты)

Каждый класс должен содержать необходимые свойства и методы для выполнения функций управления библиотекой.

**Пример связей:**

* Книга может быть написана несколькими авторами.
* Один пользователь может взять несколько книг.
* Выдача книги регистрируется как отдельная операция (класс Loan).

**2. Реализация классов на языке C#:**

На основе разработанной диаграммы классов, реализуйте систему с использованием C#. Каждый класс должен включать поля (свойства), методы и конструкторы. Пример реализации некоторых классов:

* **Book**:
  + Поля: Title, ISBN, Author, PublicationYear, AvailabilityStatus.
  + Методы: ChangeAvailabilityStatus(), GetBookInfo().
* **User**:
  + Поля: Id, Name, Email, UserType.
  + Методы: Register(), Login(), BorrowBook() (для читателя).
* **Loan**:
  + Поля: Book, Reader, LoanDate, ReturnDate.
  + Методы: IssueLoan(), ReturnBook().

**3. Внедрение принципов SOLID:**

При реализации классов необходимо следовать принципам SOLID:

* Принцип единственной ответственности (каждый класс должен отвечать за одну задачу).
* Принцип открытости/закрытости (классы должны быть открыты для расширения, но закрыты для модификации).
* Принцип подстановки Барбары Лисков.
* Принцип разделения интерфейсов.
* Принцип инверсии зависимостей.

**4. Написание кода:**

Реализуйте проект с использованием классов и методов в соответствии с диаграммой классов. Убедитесь, что все связи и зависимости между объектами работают корректно. Используйте наследование для классов Reader и Librarian.

**Дополнительные требования:**

1. **Исключения**: Реализуйте обработку исключений, например, при попытке выдать книгу, которая недоступна.
2. **Интерфейсы и абстрактные классы**: Используйте интерфейсы или абстрактные классы для описания поведения пользователей библиотеки.
3. **Сохранение данных**: Добавьте возможность сохранения информации о книгах, пользователях и операциях в файл или базу данных.

**Задание:**  
Создайте простое консольное приложение на языке C#, которое моделирует систему управления библиотекой. Приложение должно включать компоненты, представляющие основные сущности системы: Книга, Читатель, Библиотекарь, Каталог, Учетная система.

1. **Описание компонентов:**
   * **Книга** - компонент, представляющий сущность книги в библиотеке. Включает такие свойства, как Название, Автор, Жанр, ISBN.
   * **Читатель** - компонент, представляющий читателя библиотеки. Включает свойства Имя, Фамилия, Номер билета.
   * **Библиотекарь** - компонент, отвечающий за выдачу и возврат книг. Обеспечивает операции по работе с книгами и учетной системой.
   * **Каталог** - компонент, представляющий каталог библиотеки. Включает методы поиска и фильтрации книг по жанру, автору, названию.
   * **Учетная система** - компонент, который ведет учет всех выданных и возвращенных книг, фиксирует информацию о том, кто и когда взял книгу.
2. **Создайте диаграмму компонентов**, чтобы визуализировать архитектуру системы и зависимости между этими компонентами. В диаграмме должны быть отображены все перечисленные выше компоненты и связи между ними.
3. **Реализуйте основные классы** для каждого компонента на языке C#:
   * Определите классы для всех компонентов.
   * Добавьте основные свойства и методы для каждой сущности (например, метод ВыдатьКнигу() в компоненте Библиотекарь, метод ПоискКниги() в компоненте Каталог).
4. **Дополнительные требования:**
   * Опишите взаимодействие между компонентами.
   * Определите интерфейсы для основных операций в каждом компоненте, например, интерфейс для учетной системы и интерфейс для каталога.

**Результат выполнения задания:**

* Исходный код всех компонентов.
* Диаграмма компонентов системы с указанием связей между ними.