

**Задание:** Управление библиотекой с помощью умных указателей

**Цель:** Закрепить знания об умных указателях и их применении для управления памятью в C++.

**Описание:** Реализуйте программу, которая использует умные указатели для управления библиотекой и демонстрирует различные сценарии их использования.

**Задание:**

**Часть 1: Умные указатели `std::unique_ptr` (1 балл)**

- Создайте класс `Book`, содержащий поля `title`, `author` и `year`.
- Используйте `std::unique_ptr` для управления объектами `Book`.
- Напишите функцию, которая принимает `std::unique_ptr<Book>` и выводит информацию о книге.
- Напишите функцию, которая создает объект `Book` с помощью `std::unique_ptr` и передает его в функцию.

**Часть 2: Умные указатели `std::shared_ptr` (1 балл)**

- Создайте класс `Library`, содержащий список книг (используйте контейнер `std::vector<std::shared_ptr<Book>>`).
- Напишите методы для добавления книги в библиотеку, удаления книги из библиотеки и вывода списка книг.
- Напишите функцию, которая создает несколько объектов `Book` с помощью `std::shared_ptr`, добавляет их в библиотеку и выводит список книг.

**Часть 3: Умные указатели `std::weak_ptr` (1 балл)**

- Добавьте в класс `Library` метод, возвращающий `std::weak_ptr<Book>` для поиска книги по названию и/или автору.
- Напишите функцию, которая находит книгу по названию и/или автору, используя `std::weak_ptr`, и выводит информацию о книге, если она существует.

**Часть 4: Управление пользователями библиотеки (2 балл)**

- Создайте класс `User`, содержащий поля `name` и `id`.
- Используйте `std::shared_ptr` для управления объектами `User`.
- Добавьте в класс `Library` методы для добавления и удаления пользователей, а также для выдачи книг пользователям.
- Напишите функцию, которая демонстрирует добавление пользователей, выдачу книг пользователям и вывод списка пользователей и их книг.

**Часть 5: `std::make_shared` и `std::allocate_shared` (1 балл)**

- Используйте `std::make_shared` и `std::allocate_shared` для создания объектов `Book` и `User`.
- Напишите функцию, которая демонстрирует использование этих функций для создания объектов и их добавления в библиотеку.

**Часть 6: Реализация графического интерфейса с использованием Qt (2 балла)**

- Создайте проект с использованием Qt и настройте основной виджет, чтобы отображать список книг и пользователей.
- Используйте `QListView`, `QTableView` или другие подходящие виджеты для отображения списка книг и пользователей.

- Реализуйте диалог для добавления/удаления книг и пользователей, используя слоты и сигналы для обработки событий.
- Добавьте возможность поиска книги по названию и/или автору с помощью диалогового окна.
- Создайте меню с элементами для добавления, удаления и поиска книг и пользователей.

#### **Часть 7: Пользовательские диалоги и взаимодействие с пользователем (2 балла)**

- Добавьте в проект пользовательские диалоги для добавления/удаления книг и пользователей, а также для поиска книг.
- Реализуйте слоты и сигналы для обработки взаимодействий пользователя с графическим интерфейсом.
- Обеспечьте корректное обновление списка книг и пользователей при добавлении, удалении и поиске.
- Создайте окно, отображающее информацию о выбранной книге и пользователе.
- Реализуйте механизмы валидации ввода данных в диалогах для предотвращения некорректного ввода.

#### **Итоговое задание:**

1. Создайте класс `Book`, содержащий поля `title`, `author` и `year`.
2. Используйте `std::unique_ptr` для управления объектами `Book`.
3. Создайте класс `Library`, содержащий список книг (используйте контейнер `std::vector<std::shared_ptr<Book>>`).
4. Добавьте методы для добавления, удаления и поиска книг.
5. Создайте класс `User`, содержащий поля `name` и `id`, и используйте `std::shared_ptr` для управления объектами `User`.
6. Добавьте методы для добавления, удаления пользователей и выдачи книг пользователям.
7. Реализуйте графический интерфейс с использованием `Qt`, включающий отображение списка книг и пользователей, а также диалоги для добавления, удаления и поиска.
8. Реализуйте слоты и сигналы для обработки взаимодействий пользователя и корректного обновления интерфейса.