Государственное бюджетное образовательное учреждение среднего профессионального образования Новосибирской области

Новосибирский химико- 630102 г. Новосибирск, ул. Садовая, 26,

технологический колледж. Тел./факс: (383) 266-00-44, тел.: (383) 266-00-54,

nhtk@mail.ru, http://nhtk-edu.ru

им. Д.И. Менделеева

**ОТЧЕТ ПО ПРОЕКТУ**

по дисциплине «Разработка программных модулей»

Выполнили:

Студенты группы 09.07.12п

Амирян Лиза и Корейба Егор

Проверил:

Соломенный Ю.Ю.

Новосибирск – 2024

Оглавление

[Введение 3](#_Toc185585803)

[Основные функции 3](#_Toc185585804)

[Дополнительные функции 4](#_Toc185585805)

[Архитектура приложения 4](#_Toc185585806)

[Макет приложения 5](#_Toc185585807)

[Структура проекта 6](#_Toc185585808)

[Основные модули и их взаимодействие 6](#_Toc185585809)

[Взаимодействие между модулями 9](#_Toc185585810)

[Тестирование 10](#_Toc185585811)

# Введение

Приложение "Моя библиотека" предназначено для управления личной библиотекой книг. Пользователи могут добавлять новые книги, просматривать информацию о книгах, фильтровать и искать книги, а также создавать и редактировать заметки к книгам. Приложение разработано с использованием Python и библиотеки PyQt5 для создания графического интерфейса.

**Описание функциональности**

Основные функции

1. Добавление новой книги:
   * Пользователь может добавить новую книгу, указав название, автора, жанр, путь к книге и обложке.
   * Информация о книге сохраняется в базе данных.
2. Просмотр библиотеки книг:
   * Книги отображаются в виде карточек, содержащих название, автора, жанр и обложку.
   * Карточки книг распределяются по горизонтали по 4 штуки в одной строке.
   * Поддерживается прокрутка книг.
3. Фильтрация и поиск книг:
   * Пользователь может фильтровать книги по автору и жанру.
   * Пользователь может искать книги по названию и автору.
4. Создание и редактирование заметок:
   * Пользователь может перейти к заметкам для выбранной книги, дважды нажав на карточку книги.
   * Пользователь может создавать и редактировать заметки к книгам.
   * Заметки сохраняются в базе данных.
5. Навигация:
   * Пользователь может вернуться к библиотеке книг из читательского дневника.

# Дополнительные функции

1. Стилизация интерфейса:
   * Приложение стилизовано с использованием CSS для улучшения внешнего вида.
   * Основной цвет интерфейса - пурпурный, второстепенный - бледно-зеленый.
2. Открытие книги:
   * Пользователь может открыть книгу, нажав на кнопку "Открыть книгу" на карточке книги.

# Архитектура приложения

**Компоненты**

1. Главное окно (MainWindow):
   * Загружает стили из файла **styles.css**.
   * Содержит два основных виджета: **LibraryView** и **NotesView**.
   * Переключается между виджетами в зависимости от действий пользователя.
2. Библиотека книг (LibraryView):
   * Отображает список книг в виде карточек.
   * Поддерживает фильтрацию книг по автору и жанру.
   * Поддерживает поиск книг по названию и автору.
   * Позволяет добавлять новые книги через диалоговое окно.
   * Поддерживает двойное нажатие на карточку книги для перехода к заметкам.
3. Читательский дневник (NotesView):
   * Отображает заметки для выбранной книги.
   * Позволяет редактировать и сохранять заметки.
   * Поддерживает возврат к библиотеке книг.
4. Диалоговое окно добавления книги (AddBookDialog):
   * Позволяет пользователю ввести информацию о новой книге (название, автор, жанр, путь к книге и обложке).
   * Сохраняет новую книгу в базе данных.
5. База данных (Database):
   * Хранит информацию о книгах и заметках.
   * Поддерживает операции добавления, обновления и получения данных.

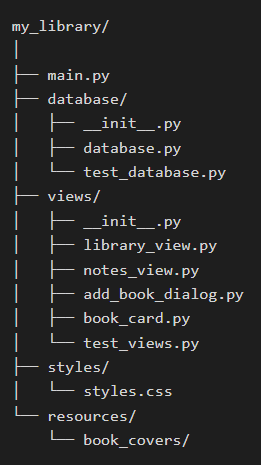
# Макет приложения

Визуализация главной страницы****



Всплывающее окно, при нажатии кнопки «Добавить книгу»

# Структура проекта

****

# Основные модули и их взаимодействие

1. Главное окно (MainWindow)
   * Файл: **main.py**
   * Описание: Основной файл приложения, который инициализирует главное окно и загружает стили.
   * Взаимодействие:
     + Загружает стили из файла **styles.css**.
     + Создает и управляет виджетами **LibraryView** и **NotesView**.
     + Устанавливает связи между виджетами для переключения между библиотекой книг и читательским дневником.
2. Библиотека книг (LibraryView)
   * Файл: **views/library\_view.py**
   * Описание: Виджет, отображающий список книг в виде карточек. Поддерживает фильтрацию и поиск книг, а также добавление новых книг.
   * Взаимодействие:
     + Загружает данные о книгах из базы данных (**database.database**).
     + Взаимодействует с **BookCard** для отображения информации о книгах.
     + Открывает диалоговое окно **AddBookDialog** для добавления новых книг.
     + Переключается на **NotesView** при двойном нажатии на карточку книги.
3. Читательский дневник (NotesView)
   * Файл: **views/notes\_view.py**
   * Описание: Виджет, отображающий заметки для выбранной книги. Позволяет редактировать и сохранять заметки.
   * Взаимодействие:
     + Загружает заметки для выбранной книги из базы данных (**database.database**).
     + Сохраняет изменения в заметках в базе данных.
     + Переключается на **LibraryView** при нажатии кнопки "Назад к библиотеке".
4. Диалоговое окно добавления книги (AddBookDialog)
   * Файл: **views/add\_book\_dialog.py**
   * Описание: Диалоговое окно, позволяющее пользователю ввести информацию о новой книге (название, автор, жанр, путь к книге и обложке).
   * Взаимодействие:
     + Сохраняет новую книгу в базе данных (**database.database**).
5. Карточка книги (BookCard)
   * Файл: **views/book\_card.py**
   * Описание: Виджет, отображающий информацию о книге в виде карточки.
   * Взаимодействие:
     + Отображает информацию о книге (название, автор, жанр, обложка).
     + Открывает книгу при нажатии на кнопку "Открыть книгу".
     + Переключается на **NotesView** при двойном нажатии на карточку книги.
6. База данных (Database)
   * Файл: **database/database.py**
   * Описание: Модуль, отвечающий за хранение и управление данными о книгах и заметках.
   * Взаимодействие:
     + Хранит информацию о книгах и заметках.
     + Поддерживает операции добавления, обновления и получения данных.
     + Взаимодействует с **LibraryView**, **NotesView**, **AddBookDialog** и **BookCard** для выполнения операций с данными.
7. Стили (styles.css)
   * Файл: **styles/styles.css**
   * Описание: Файл стилей, используемый для стилизации графического интерфейса приложения.
   * Взаимодействие:
     + Загружается и применяется в **main.py** для стилизации всех виджетов приложения.

# Взаимодействие между модулями

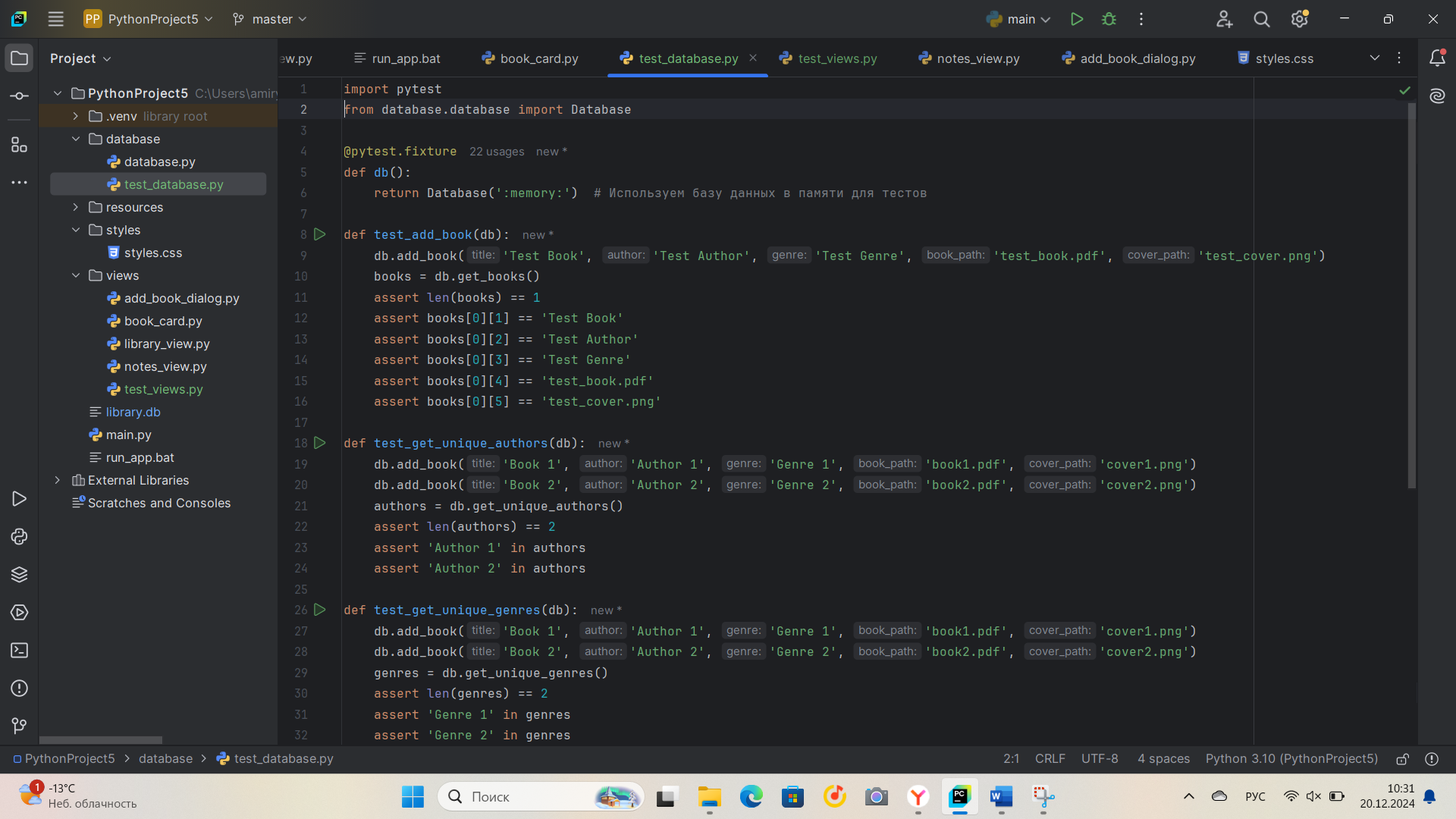
1. Инициализация и загрузка стилей:
   * **main.py** загружает стили из **styles/styles.css** и инициализирует главное окно.
2. Отображение библиотеки книг:
   * **LibraryView** загружает данные о книгах из **database.database** и отображает их в виде карточек (**BookCard**).
   * **LibraryView** взаимодействует с **AddBookDialog** для добавления новых книг.
   * **LibraryView** переключается на **NotesView** при двойном нажатии на карточку книги.
3. Редактирование и сохранение заметок:
   * **NotesView** загружает заметки для выбранной книги из **database.database** и отображает их.
   * **NotesView** сохраняет изменения в заметках в **database.database**.
   * **NotesView** переключается на **LibraryView** при нажатии кнопки "Назад к библиотеке".
4. Добавление новой книги:
   * **AddBookDialog** сохраняет новую книгу в **database.database**.
5. Отображение информации о книге:
   * **BookCard** отображает информацию о книге и взаимодействует с **NotesView** при двойном нажатии на карточку книги.

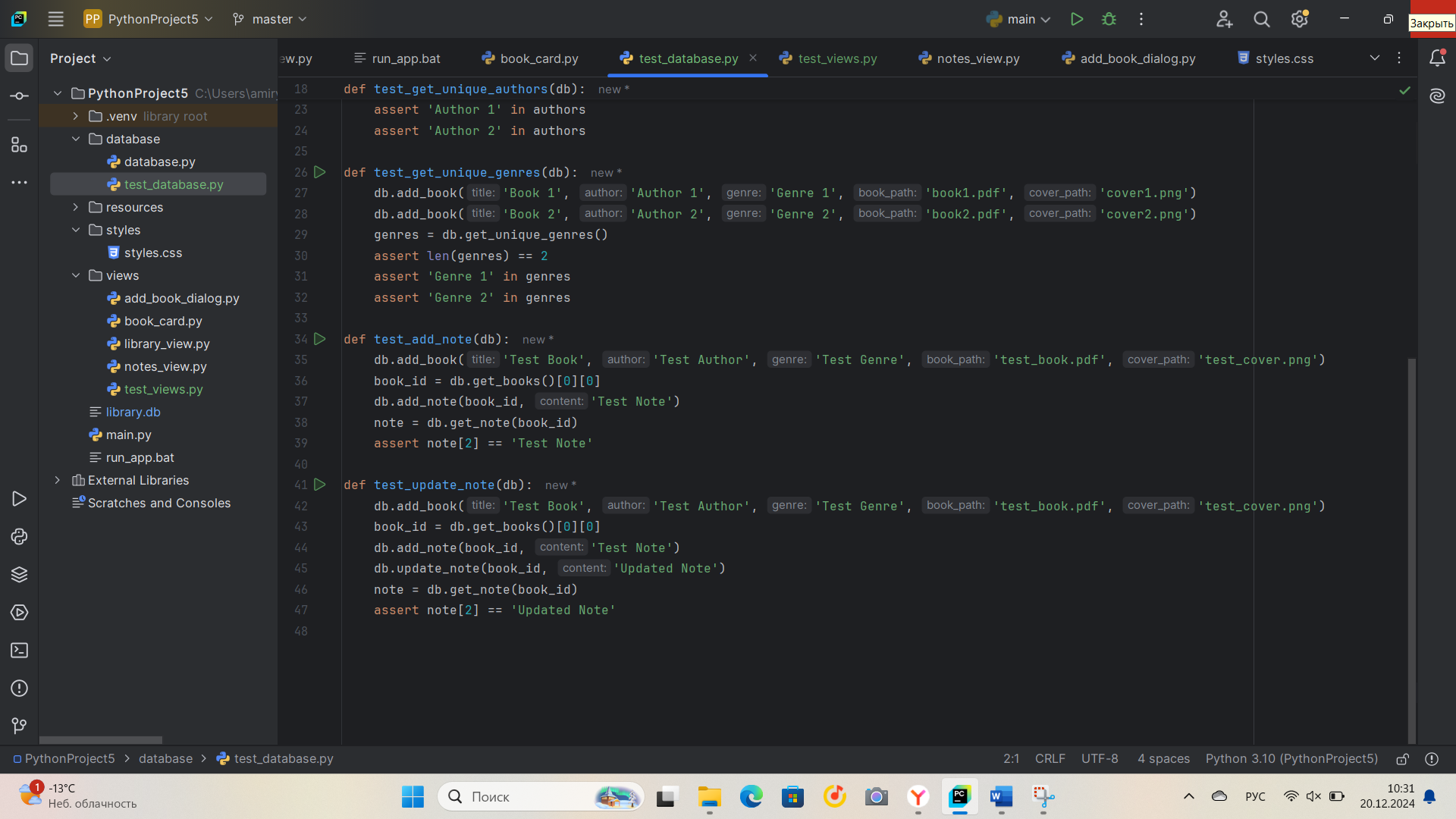
# Тестирование

**Тесты базы данных**

Тесты для базы данных включают проверку основных операций, таких как добавление книг, получение списка книг, фильтрация и поиск книг, а также создание и редактирование заметок.

**Пример тестов базы данных (database/test\_database.py)**



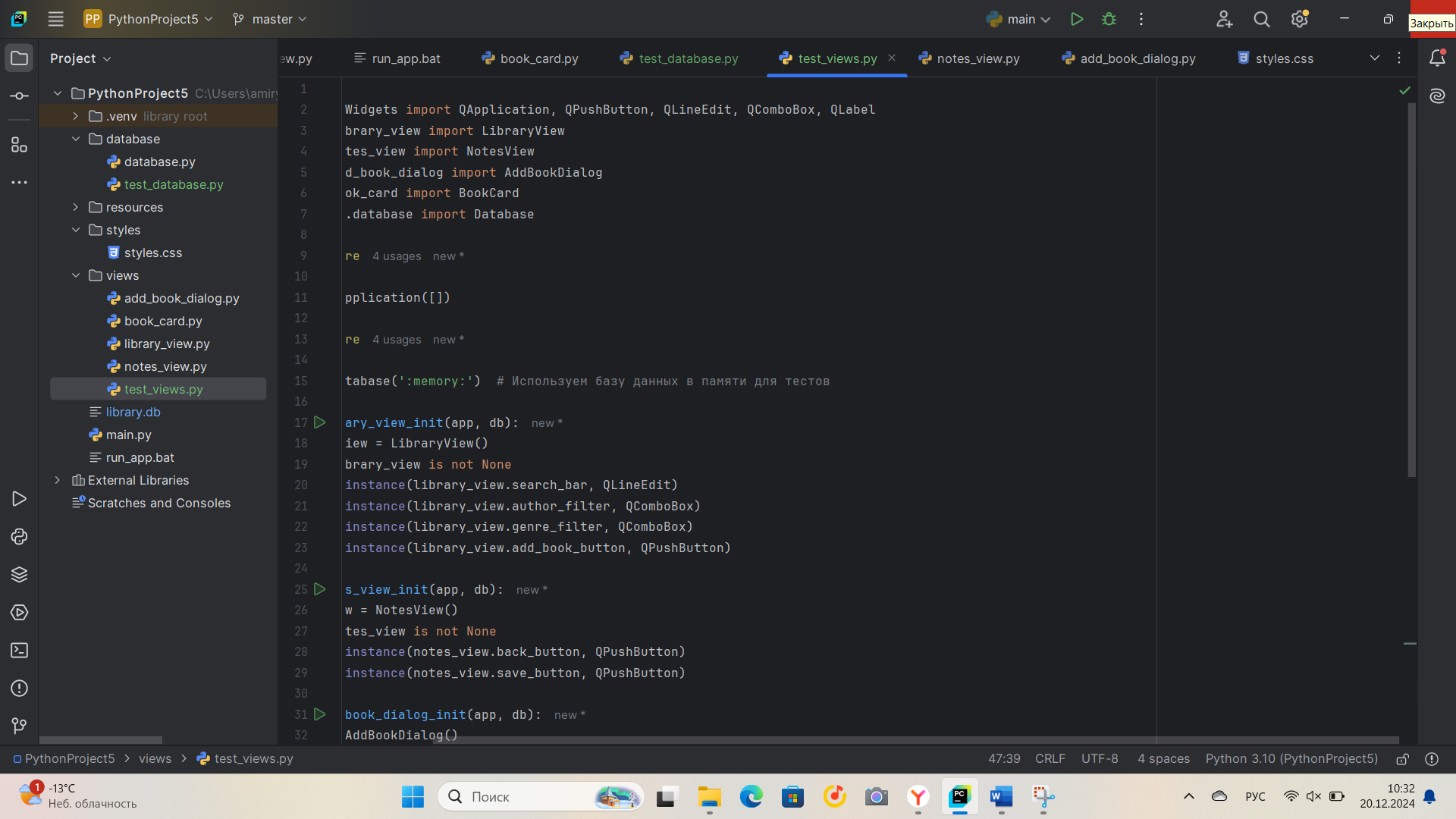


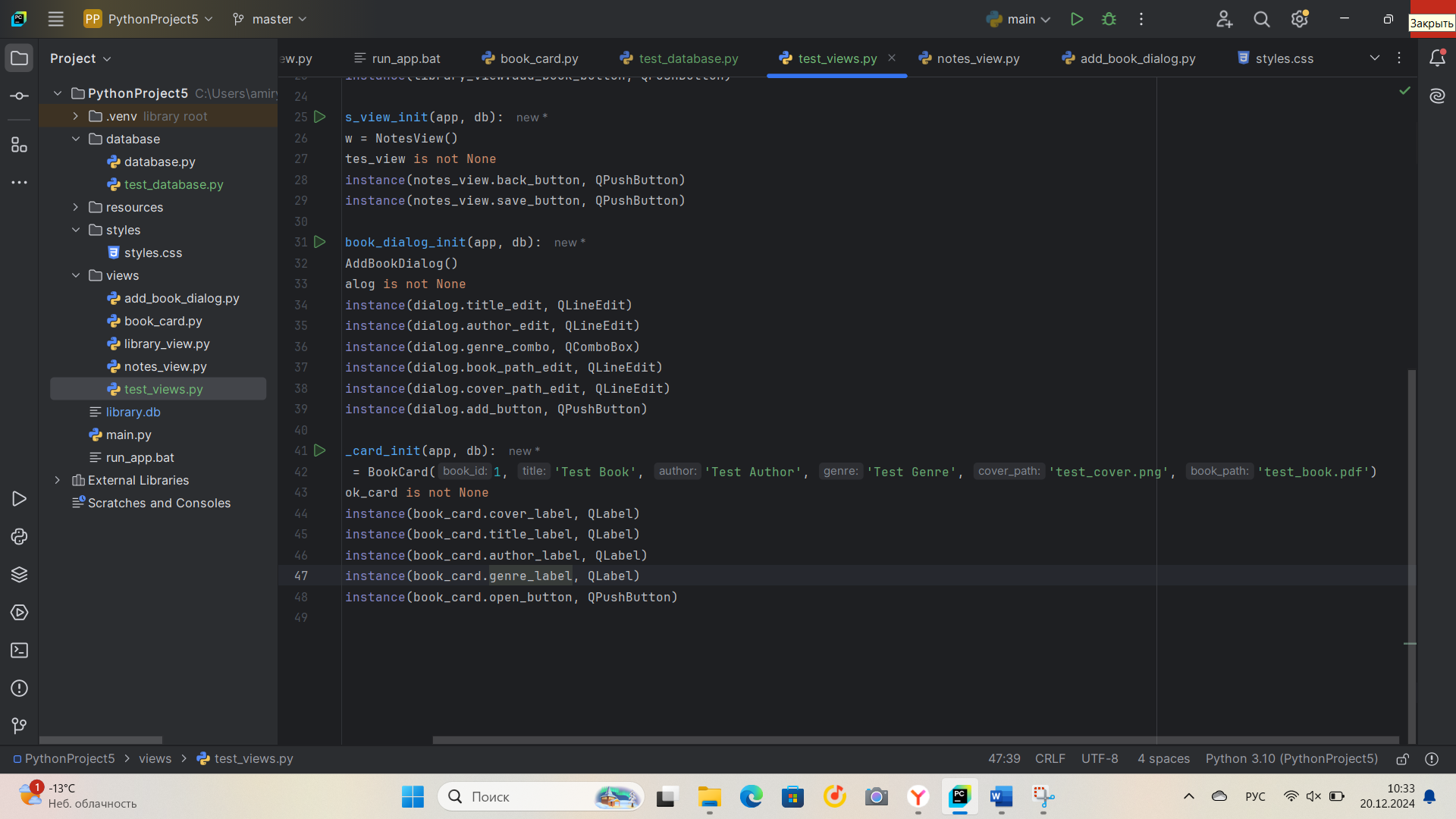
Мы создаем тестовую базу данных и с помощью **test\_views.py** добавляем в нее заполненные книги, проверяем на корректность использования данных и корректность отображения.

1. Инициализация базы данных:
   * **db**: Фикстура для создания базы данных в памяти. Это позволяет изолировать тесты от реальной базы данных и обеспечивает быструю инициализацию и очистку данных между тестами.
2. Тест добавления книги:
   * **test\_add\_book**: Проверяет, что книга корректно добавляется в базу данных. После добавления книги проверяется, что книга действительно была добавлена и что все её атрибуты (название, автор, жанр, путь к книге и обложке) сохранены правильно.
3. Тест получения уникальных авторов:
   * **test\_get\_unique\_authors**: Проверяет, что метод **get\_unique\_authors** корректно возвращает список уникальных авторов. Добавляются две книги с разными авторами, и проверяется, что список уникальных авторов содержит оба автора.
4. Тест получения уникальных жанров:
   * **test\_get\_unique\_genres**: Проверяет, что метод **get\_unique\_genres** корректно возвращает список уникальных жанров. Добавляются две книги с разными жанрами, и проверяется, что список уникальных жанров содержит оба жанра.
5. Тест добавления заметки:
   * **test\_add\_note**: Проверяет, что заметка корректно добавляется к книге. После добавления заметки проверяется, что заметка действительно была добавлена и что её содержимое сохранено правильно.
6. Тест обновления заметки:
   * **test\_update\_note**: Проверяет, что заметка корректно обновляется. После добавления заметки она обновляется, и проверяется, что содержимое заметки было обновлено правильно.

**Тесты виджетов**

Тесты для виджетов включают проверку инициализации и основных функций виджетов.





Проверяем корректность инициализации и основных функций виджетов, используемых в приложении "Моя библиотека". Эти тесты помогают убедиться, что виджеты работают правильно и соответствуют требованиям.

1. Инициализация приложения:
   * **app**: Фикстура для инициализации приложения **QApplication**. Это необходимо для создания виджетов в тестах.
2. Инициализация базы данных:
   * **db**: Фикстура для создания базы данных в памяти. Это позволяет изолировать тесты от реальной базы данных.
3. Тест инициализации **LibraryView**:
   * **test\_library\_view\_init**: Проверяет, что виджет **LibraryView** инициализируется корректно и содержит все необходимые элементы (поле поиска, фильтры по автору и жанру, кнопка добавления книги).
4. Тест инициализации **NotesView**:
   * **test\_notes\_view\_init**: Проверяет, что виджет **NotesView** инициализируется корректно и содержит все необходимые элементы (кнопки "Назад к библиотеке" и "Сохранить заметку").
5. Тест инициализации **AddBookDialog**:
   * **test\_add\_book\_dialog\_init**: Проверяет, что диалоговое окно **AddBookDialog** инициализируется корректно и содержит все необходимые элементы (поля ввода для названия, автора, жанра, пути к книге и обложке, а также кнопка добавления книги).
6. Тест инициализации **BookCard**:
   * **test\_book\_card\_init**: Проверяет, что виджет **BookCard** инициализируется корректно и содержит все необходимые элементы (метки для отображения обложки, названия, автора, жанра и кнопка "Открыть книгу").

**Руководство по установке**

**ребования**

* Операционная система: Windows, macOS или Linux.
* Python: Убедитесь, что у вас установлен Python 3.6 или выше.
* PyQt5: Библиотека для создания графического интерфейса.
* SQLite: База данных для хранения информации о книгах и заметках.

**Шаги установки**

1. Установите Python:
   * Скачайте и установите Python с [официального сайта Python](https://www.python.org/downloads/).
2. Установите PyQt5:
   * Откройте командную строку или терминал и выполните следующую команду:

**pip install PyQt5**

1. Скачайте проект:
   * Скачайте архив с проектом с [ссылки на скачивание] (замените на фактическую ссылку).
2. Распакуйте архив:
   * Распакуйте скачанный архив в удобное для вас место на вашем компьютере.
3. Установите зависимости:
   * Перейдите в директорию проекта и выполните следующую команду для установки всех необходимых зависимостей:

**pip install -r requirements.txt**

**Запуск приложения**

1. Перейдите в директорию проекта:
   * Откройте командную строку или терминал и перейдите в директорию проекта:

**cd path/to/your/project**

1. Запустите приложение:
   * Выполните следующую команду для запуска приложения:

**python main.py**

**Руководство по использованию**

**Основные функции**

1. Добавление новой книги:
   * Нажмите кнопку "Добавить книгу" в правом нижнем углу окна.
   * Откроется диалоговое окно для добавления новой книги.
   * Введите информацию о книге: название, автор, жанр, путь к книге и обложке.
   * Нажмите кнопку "Добавить книгу", чтобы сохранить новую книгу в библиотеке.
2. Просмотр библиотеки книг:
   * Книги отображаются в виде карточек, содержащих название, автора, жанр и обложку книги.
   * Карточки книг распределяются по горизонтали по 4 штуки в одной строке.
3. Фильтрация и поиск книг:
   * Используйте выпадающие списки для фильтрации книг по автору и жанру.
   * Введите текст в поле поиска для поиска книг по названию и автору.
4. Создание и редактирование заметок:
   * Дважды щелкните на карточку книги, чтобы перейти к заметкам для этой книги.
   * Откроется окно читательского дневника, где отображаются заметки для выбранной книги.
   * Введите или отредактируйте заметки в текстовом поле.
   * Нажмите кнопку "Сохранить заметку", чтобы сохранить изменения.
5. Возврат к библиотеке:
   * Нажмите кнопку "Назад к библиотеке", чтобы вернуться к библиотеке книг.
6. Открытие книги:
   * Нажмите кнопку "Открыть книгу" на карточке книги, чтобы открыть книгу на вашем устройстве.

**Заключение**

Приложение "Моя библиотека" успешно реализовано и протестировано. Оно предоставляет удобный интерфейс для управления личной библиотекой книг и создания заметок к книгам. Пользователи могут добавлять новые книги, фильтровать и искать книги, а также создавать и редактировать заметки. Все данные хранятся в базе данных, что обеспечивает надежное хранение и управление информацией.