|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** |
|  |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** |

|  |
| --- |
| **КУРСОВОЙ ПРОЕКТ** |
|  |
| по междисциплинарному курсу: МДК.02.02. Технология разработки и зашиты баз данных. |
|  |
| на тему: Разработка информационной системы книжного магазина. |
|  |
| студента группы 090203-9о-20/2  специальности 09.02.03 Программирование в компьютерных системах |
| Якимова Дмитрия Романовича |

|  |  |  |
| --- | --- | --- |
| Студент |  | Д.Р. Якимов |
| Руководитель курсового проекта |  | Е.А. Ларионова |
| Председатель ПЦК специальности 09.02.03 Программирование в компьютерных системах |  | А.И. Глускер |

|  |  |  |
| --- | --- | --- |
| Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | |
| Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Заведующий отделением № 1 |  | И.А. Миланова |

Москва

2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc158318606)

[1 ОСНОВНАЯ ЧАСТЬ 5](#_Toc158318607)

[1.1 Обзор и анализ предметной области 5](#_Toc158318608)

[1.2 Жизненный цикл базы данных 5](#_Toc158318609)

[1.3 Защита базы данных 7](#_Toc158318610)

[1.4 Выбор и характеристика СУБД 8](#_Toc158318611)

[1.5 Выбор и характеристика среды разработки приложения 8](#_Toc158318612)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 9](#_Toc158318613)

[2.1 Постановка задачи 9](#_Toc158318614)

[2.2 Архитектура информационной системы 9](#_Toc158318615)

[2.3 Логическая модель базы данных 10](#_Toc158318616)

[2.4 Нормализация таблиц 10](#_Toc158318617)

[2.5 Описание таблиц 11](#_Toc158318618)

[2.7 Разработка приложения 14](#_Toc158318619)

[2.7.1 Структурная схема приложения 14](#_Toc158318620)

[2.7.2 Используемая технология доступа к данным 15](#_Toc158318621)

[2.7.3 Описание процесса отладки приложения 15](#_Toc158318622)

[2.7.4 Защита информационной системы 15](#_Toc158318623)

[2.8 Инструкция пользователю 15](#_Toc158318624)

[2.8.1 Общие сведения об информационной системе 15](#_Toc158318625)

[2.8.2 Требования к техническим средствам 16](#_Toc158318626)

[2.8.3 Требования к программным средствам 16](#_Toc158318627)

[2.8.4 Настройка информационной системы 16](#_Toc158318628)

[2.8.5 Формы ввода 17](#_Toc158318629)

[ЗАКЛЮЧЕНИЕ 27](#_Toc158318630)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 28](#_Toc158318631)

[ПРИЛОЖЕНИЕ А 29](#_Toc158318632)

[ПРИЛОЖЕНИЕ Б 41](#_Toc158318633)

ВВЕДЕНИЕ

В современном мире стремительно происходит развитие информационных технологий, за счет чего появляются различные интернет-магазины, которые ранее были невозможны, но в текущее время представить без них свою жизнь почти невозможно. По данным сервиса DataInsight который составил отчет о количестве продаж книг на различных площадках для продажи, а также специализированных интернет-магазинах. За 12 месяцев, в период с июля 2020 года по июнь 2021 года было оформлено 42,6 миллионов заказов на книги. В связи с чем, для развития темы просмотра информациях о различных книгах, были разработаны приложение и база данных, в которую будет собираться подробная информация о книгах.

Основная цель исследования – это приобретение и закрепление навыков в области объектно-ориентированного анализа и проектирования. Также исследование преследует цели:

* ­получение навыков в работе со средой MS Visual Studio и WPF,
* закрепление знаний SQL при работе с ним в различных СУБД.

Основные задачи проекта:

* анализ предметной области,
* проектирование базы данных средствами ERWin Data Modeler,
* реализация защиты базы данных в среде PostgreSQL,
* разработка основных функций приложения в среде MS Visual Studio,
* тестирование приложения.

В рассматриваемой предметной области представлены следующие приложения:

* Лабиринт – книжный интернет магазин это сайт, где можно купить книги, игрушки, аудио, видео,
* Book24 – интернет-магазин издательской группы Эксмо-АСТ,
* Читай-город – это самая большая в России сеть книжных магазинов и интернет-магазин. Помимо книг можно найти канцтовары, сладости, подарочную упаковку и идеи для сюрпризов,
* Буквоед – сеть книжных магазинов, кроме книг, в магазинах сети представлены канцелярские товары, игры, сувениры, товары для хобби и творчества,
* Республика – Сеть розничных магазинов книг, в интернет-магазине можно найти и заказать интересные книги для чтения, а также товары для творчества,

Разработка базы данных книжного магазина позволит:

* хранить информацию о книгах, издателях, авторах, жанрах, сериях,
* просматривать информацию о книгах.

1 ОСНОВНАЯ ЧАСТЬ

* 1. Обзор и анализ предметной области

База данных создается для обеспечения пользователя возможностью поиска книг и добавления их в список желаемых. База данных должна содержать данные о книгах, авторах, жанрах, издателях и сериях, а также для пользовательской части данные о пользователях, добавленных книг в список желаемых и ролях пользователей.

В соответствии с предметной областью система строится с учетом следующих особенностей:

* у каждой книги должен быть автор, издатель, жанр, серия.

Выделим базовые сущности этой предметной области:

* книги,
* авторы,
* издатели,
* жанры,
* серии,
* пользователи,
* книги пользователей,
* роли пользователей,
* корзины.

Атрибуты книг – название, автор, жанр, серия, описание, год публикации, количество страниц, стоимость.

Атрибуты авторов – наименование автора.

Атрибуты жанров – наименование жанра.

Атрибуты серии – наименование серии.

Атрибуты жанров – название.

Атрибуты пользователей – логин, пароль, дата регистрации, роль.

Атрибуты книг пользователей – пользователь, книга, дата добавления.

Атрибуты ролей – наименование роли, наличие административных прав,

Атрибуты корзины – ссылка на книгу, ссылка на пользователя, количество товаров.

* 1. Жизненный цикл базы данных

В жизненном цикле базы данных можно выделить пять этапов:

1) анализ,

2) проектирование,

3) реализация,

4) тестирование,

5) эксплуатация.

На первом этапе был произведен анализ предметной области, в результате которого выяснилось, что необходимо создать следующие таблицы:

* книги,
* авторы,
* жанры,
* издатели,
* серии,
* пользователи,
* книги пользователей,
* роли,
* корзины.

На этапе проектирования была создана концептуальная модель базы данных:

* books (book\_id, genre\_id, author\_id, book\_name, book\_description, publisher\_id, series\_id, book\_publish\_year, book\_pages, book\_price)
* authors (author\_id, author\_name)
* genres (genre\_id, genre\_name)
* publishers (publisher\_id, publisher\_name)
* series (series\_id, series\_id)
* users (user\_id, user\_login, user\_password, user\_register\_date, role\_id)
* users\_books (ub\_id, user\_id, book\_id, ub\_add\_date)
* roles (role\_id, role\_name, role\_is\_admin)
* carts (primary\_id, user\_id, book\_id, cart\_values)

Также на этом этапе были спроектированы логическая модель базы данных, которая представлена в пункте 2.3, а также физическая модель, показанная в рисунке 1.

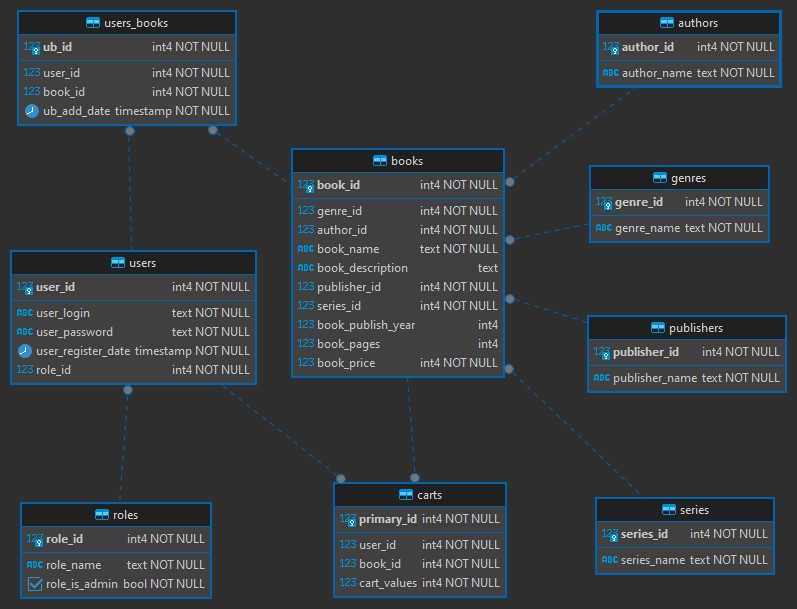


Рисунок 1 – Физическая модель базы данных

На этапе реализации был написан запрос для создания базы данных, создания и распределения прав пользователей, а также создания таблиц и других объектов БД. Код запроса указан в приложении А.

На этапе тестирования БД была заполнена тестовыми данными. Результаты тестирования показали, что схема БД является пригодной для ввода в эксплуатацию.

На этапе эксплуатации для взаимодействия с БД было разработано приложение. Код приложения представлен в приложении Б.

* 1. Защита базы данных

Для защиты базы данных используются параметризированные запросы. Использование параметризированных запросов позволяет избежать SQL-инъекций, которые могут стать причиной несанкционированного доступа к базе данных. Также для защиты базы данных используется вход по логину/паролю SQL, который позволяет ограничить права пользователя, использующего базу данных.

* 1. Выбор и характеристика СУБД

Для разработки информационной системы была выбрана СУБД PostgreSQL Server разработанная корпорацией PostgreSQL Global Development Group. Основной используемый язык запросов – Transact SQL. Transact SQL является реализацией стандарта ANSI/ISO по SQL с расширениями.

* 1. Выбор и характеристика среды разработки приложения

В качестве среды разработки приложения был выбран Microsoft Visual Studio 2022, в качестве технологии разработки использовалась технология WPF. У данной технологии можно выделить следующие преимущества:

* возможность декларативного определения графического интерфейса с помощью языка разметки XAML,
* объектная модель, которая позволяет не указывать явно размеры объектов, что позволяет создавать, интерфейсы, которые масштабируются в зависимости от размеров окна,
* аппаратное ускорение графики.

1. СПЕЦИАЛЬНАЯ ЧАСТЬ
   1. Постановка задачи

Необходимо средствами PostgreSQL Server и MS Visual Studio разработать приложение, позволяющее производить поиск и просматривать информацию о книгах.

Программа должна обеспечивать следующие функциональные возможности:

* просмотр информации о книге, пользователе,
* добавление книг, авторов, жанров, издателей, серий, пользователей и книг пользователей,
* удаление книг, авторов, жанров, издателей, серий, пользователей и книг пользователей,
* изменение информации о книгах, авторах, жанрах, издателях, сериях, пользователях.
  1. Архитектура информационной системы

Архитектура разрабатываемого приложения является двухуровневой, то есть клиент-сервер (Рисунок 2).

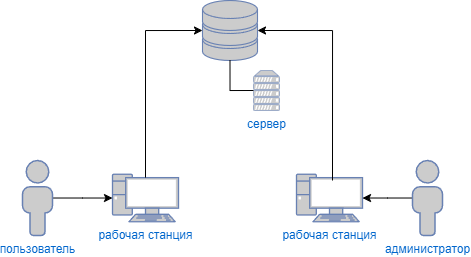


Рисунок 2 – Архитектура разрабатываемого приложения

Особенностью этой архитектуры является то, что все данные хранятся на сервере, все запросы также выполняются на стороне сервера. Это позволяет снизить требования к рабочим системам пользователей, но повышает нагрузку на сервер.

* 1. Логическая модель базы данных

На рисунке 3 представлена логическая модель базы данных книг.

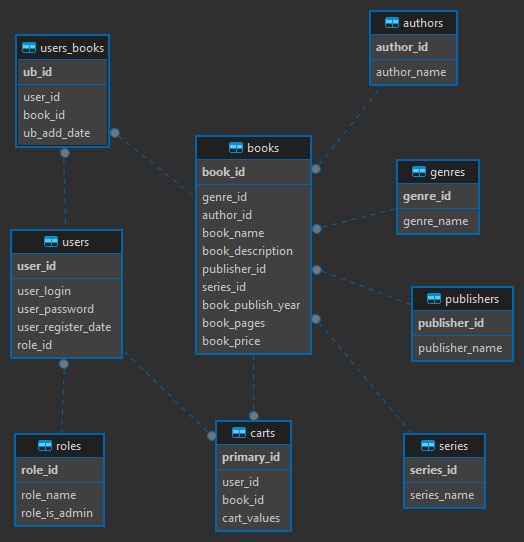


Рисунок 3 – Логическая модель базы данных

* 1. Нормализация таблиц

Отношение находится в первой нормальной форме, если все атрибуты данного отношения атомарны.

Отношение находится во второй нормальной форме, если оно находится в первой нормальной форме и каждый не ключевой атрибут зависит от первичного.

Отношение находится в третьей нормальной форме, если оно находится во второй нормальной форме и между не ключевыми атрибутами нет функциональных зависимостей.

Каждый атрибут таблиц, которые указаны в разделе 2.5 настоящего документа, является атомарным, зависит от первичного ключа, только от первичного ключа и не от чего кроме первичного ключа. В связи с чем можно сделать вывод что таблицы соответствуют третьей нормальной форме.

* 1. Описание таблиц

В таблицах 1 – 11 указано описание атрибутов таблиц базы данных.

Таблица 1 – Описание таблицы Книги

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Описание** | **Тип данных** | **Дополнительно** |
| book\_id | Идентификатор | INT | NOT NULL,  PRIMARY KEY |
| genre\_id | Идентификатор жанра | INT | NOT NULL, FOREIGN KEY |
| author\_id | Идентификатор автора | INT | NOT NULL, FOREIGN KEY |
| book\_name | Наименование книги | TEXT | NOT NULL |
| book\_description | Описание | TEXT | NULL |
| publisher\_id | Идентификатор издателя | INT | NOT NULL, FOREIGN KEY |
| series\_id | Идентификатор серии | INT | NOT NULL, FOREIGN KEY |
| book\_publish\_year | Год издания книги | INT | NULL |
| book\_pages | Количество страниц в книге | INT | NULL |
| book\_price | Стоимость книги | INT | NOT NULL |

Таблица 2 – Описание таблицы Авторы

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Описание** | **Тип данных** | **Дополнительно** |
| author\_id | Идентификатор | INT | NOT NULL,  PRIMARY KEY |
| author\_name | Наименование автора | TEXT | NOT NULL |

Таблица 3 – Описание таблицы Жанры

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Описание** | **Тип данных** | **Дополнительно** |
| genre\_id | Идентификатор | INT | NOT NULL,  PRIMARY KEY |
| genre\_name | Наименование жанра | TEXT | NOT NULL, |

Таблица 4 – Описание таблицы Издатели

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Описание** | **Тип данных** | **Дополнительно** |
| publisher\_id | Идентификатор | INT | NOT NULL  PRIMARY KEY, |
| publisher\_name | Наименование издателя | TEXT | NOT NULL |

Таблица 5 – Описание таблицы Серии

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Описание** | **Тип данных** | **Дополнительно** |
| series\_id | Идентификатор | INT | NOT NULL,  FOREIGN KEY |
| series\_name | Наименование серии | TEXT | NOT NULL,  FOREIGN KEY |

Таблица 6 – Описание таблицы Пользователи

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Описание** | **Тип данных** | **Дополнительно** |
| user\_id | Идентификатор | INT | NOT NULL, PRIMARY KEY |
| user\_login | Логин | TEXT | NOT NULL |
| user\_password | Пароль | TEXT | NOT NULL |
| user\_register\_date | Дата регистрации | TIMESTAMP | NOT NULL, CURRENT\_TIMESTAMP |
| role\_id | Идентификатор роли | INT | NOT NULL |

Таблица 7 – Описание таблицы Книги пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Описание** | **Тип данных** | **Дополнительно** |
| ub\_id | Идентификатор | INT | NOT NULL,  PRIMARY\_KEY |
| user\_id | Идентификатор пользователя | INT | NOT NULL,  FOREIGN KEY |
| book\_id | Идентификатор книги | INT | NOT NULL, FOREIGN\_KEY |
| ub\_add\_date | Дата добавления | TIMESTAMP | NOT NULL, CURRENT\_TIMESTAMP |

Таблица 8 – Описание таблицы Роли

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Описание** | **Тип данных** | **Дополнительно** |
| role\_id | Идентификатор | INT | NOT NULL,  PRIMARY KEY |
| role\_name | Наименование роли | TEXT | NOT NULL |
| role\_is\_admin | Наличие административных прав | BOOL | NOT NULL |

Таблица 8 – Описание таблицы Корзины

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Описание** | **Тип данных** | **Дополнительно** |
| primary\_id | Идентификатор | INT | NOT NULL,  PRIMARY KEY |
| user\_id | Идентификатор пользователя | INT | NOT NULL,  FOREIGN KEY |
| book\_id | Идентификатор книги | INT | NOT NULL,  FOREIGN KEY |
| cart\_values | Количество товаров в корзине | INT | NOT NULL |

2.7 Разработка приложения

2.7.1 Структурная схема приложения

Структурная схема приложения соответствует диаграммам использования для пользователя (Рисунок 4) и администратора (Рисунок 5).

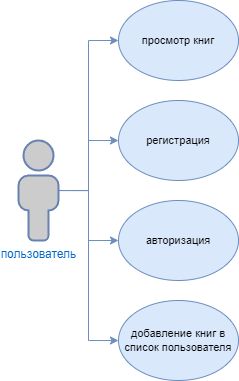


Рисунок 4 – Диаграмма использования для пользователя

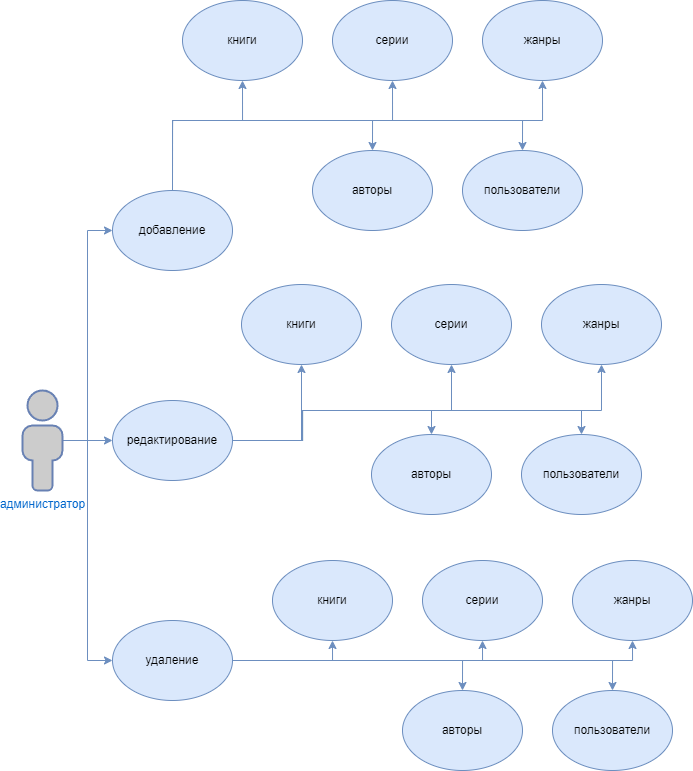


Рисунок 5 – Диаграмма использования для администратора

2.7.2 Используемая технология доступа к данным

Для доступа к данным используется расширение NPgSQL, эта технология представляет нам набор классов, через которые мы можем взаимодействовать с базой данных.

Основу интерфейса взаимодействия с базами данных в NPgSQL составляют следующие объекты: NpgsqlConnection, NpgsqlCommand, NpgsqlDataReader.

2.7.3 Описание процесса отладки приложения

В процессе разработки ИС возникали следующие ошибки:

1. ошибки при выполнении запросов,
2. ошибки при входе по логину/паролю.

Для устранения ошибок, возникающих в процессе выполнения запросов, понадобилось проверять корректность написания запросов SQL.

Ошибки при входе были связаны с тем, что при установке пустой строки подключения, расширение NPgSQL выдаёт ошибку об отсутствии определения параметров:

1. адрес хоста базы данных,
2. название базы данных,
3. пользователь базы данных,
4. пароль пользователя базы данных, при наличии.

Для решения проблемы необходимо указать требуемые данные в строке подключения к базе данных.

2.7.4 Защита информационной системы

Для защиты информационной системы были использованы параметризированные запросы, а также вход по логину/паролю SQL.

2.8 Инструкция пользователю

2.8.1 Общие сведения об информационной системе

Данная ИС предназначена для поиска и отображения информации о книгах. ИС позволяет добавлять выбранные книги в список пользователя, а также удалять их оттуда.

Для администратора ИС также предоставляет возможность:

* добавления книг, жанров, серий, авторов и пользователей,
* удаления книг, жанров, серий, авторов и пользователей,
* изменения книг, жанров, серий, авторов и пользователей.

2.8.2 Требования к техническим средствам

Системные требования к рабочему месту:

* операционная система Windows (начиная с Windows 7),
* процессор x86 или x64,
* ОЗУ объёмом не менее 4ГБ,
* дисковой накопитель (не менее 100 МБ свободного места),
* стабильное подключение к сети,
* видеоадаптер,
* монитор,
* клавиатура и мышь.

2.8.3 Требования к программным средствам

Системные требования к серверу:

* процессор x86 или x64,
* ОЗУ объёмом не менее 4ГБ,
* дисковой накопитель (не менее 1 ТБ свободного места),
* сеть 100/1000 Мбит/с.

2.8.4 Настройка информационной системы

Для запуска информационной системы необходимо:

* в экземпляре PostgreSQL Server запустить скрипт для создания базы данных книг,
* запустить проект в Visual Studio и заменить значение conntectionString в файле database.cs (Заменить Server=localhost на Server=[Адрес вашего сервера], Port=5432 на Port=[Порт СУБД на вашем сервере], User Id=postgres на User Id=[Пользователь на вашем сервере], Password= на Password[Пароль от пользователя на вашем сервере]).

2.8.5 Формы ввода

В приложении представлены следующие формы ввода:

* в окне авторизации (Рисунок 6), поля ввода логина и пароля,
* в окне регистрации (Рисунок 7), поля ввода логина, пароля и повторного пароля,
* в окне просмотра авторов (Рисунок 8), список с отображением авторов, а также контекстным меню пункты добавления, редактирования и удаления,
* в окне добавления автора (Рисунок 9), поле ввода наименования автора,
* в окне редактирования автора (Рисунок 10), поле ввода наименования автора,
* в окне просмотра жанров (Рисунок 11), список с отображением жанров, а также контекстным меню пункты добавления, редактирования и удаления,
* в окне добавления жанра (Рисунок 12), поле ввода наименования жанра,
* в окне редактирования жанра (Рисунок 13), поле ввода наименования жанра,
* в окне просмотра издателей (Рисунок 14), список с отображением издателей, а также контекстным меню пункты добавления, редактирования и удаления,
* в окне добавления издателя (Рисунок 15), поле ввода наименования издателя,
* в окне редактирования издателя (Рисунок 16), поле ввода наименования издателя,
* в окне просмотра серий (Рисунок 17), список с отображением серий, а также контекстным мню пункты добавления, редактирования и удаления,
* в окне добавления серии (Рисунок 18), поле ввода наименования серии,
* в окне редактирования серии (Рисунок 19), поле ввода наименования серии,
* в окне просмотра пользователей (Рисунок 20), список с отображением пользователей, а также контекстным меню пункты добавления, редактирования и удаления,
* в окне добавления пользователя (Рисунок 21), поля ввода логина, пароля, повтора пароля и выпадающий список с выбором роли пользователя,
* в окне редактирования пользователя (Рисунок 22), поля ввода пароля, повтора пароля и выпадающий список с выбором роли пользователя,
* в окне главного меню (Рисунок 23), список с отображением книг, а также контекстным меню пункты добавления, редактирования и удаления,
* в окне добавления книги (Рисунок 24), поля ввода названия, года издания, количества страниц, описания, выпадающие списки с выбором автора, жанра, издателя, серии,
* в окне редактирования книги (Рисунок 25), поля ввода названия, года издания, количества страниц, описания, выпадающие списки с выбором автора, жанра, издателя, серии,
* в окне просмотра книги (Рисунок 26), поля с текстом названия, автора, id товара, жанра, издателя, серии, года издания, количества страниц, описания, кнопка с добавлением или удалением книги в список пользователя,
* в окне просмотра профиля (Рисунок 27), поля с текстом id, логина, даты регистрации и роли, список с отображением добавленных книг пользователем,
* в окне просмотра корзины (Рисунок 28), чек-бокс с печатью чека, кнопка с оформлением заказа, кнопка с очисткой корзины, список с отображением добавленных книг в корзину,
* в окне печати чека (Рисунок 29), выбор устройства для печати содержимого чека,
* в содержимом чека (Рисунок 30), номер чека, дата оформления заказа, наименование книг.

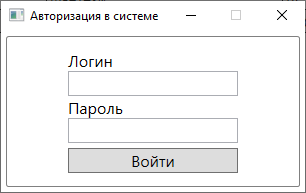


Рисунок 6 – Окно авторизации

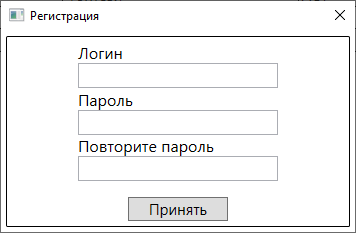


Рисунок 7 – Окно регистрации

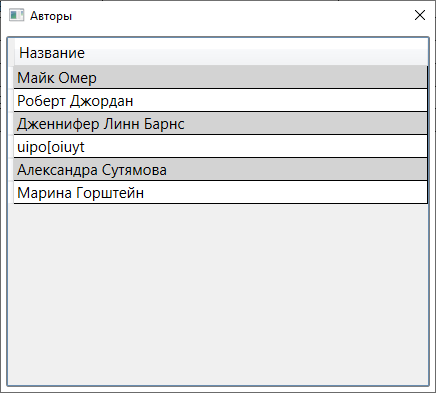


Рисунок 8 – Окно просмотра авторов

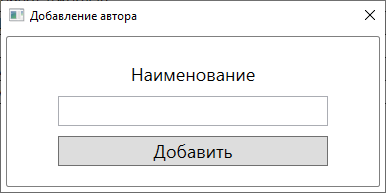


Рисунок 9 – Окно добавления автора

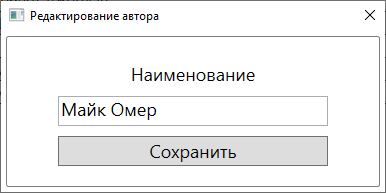


Рисунок 10 – Окно редактирования автора

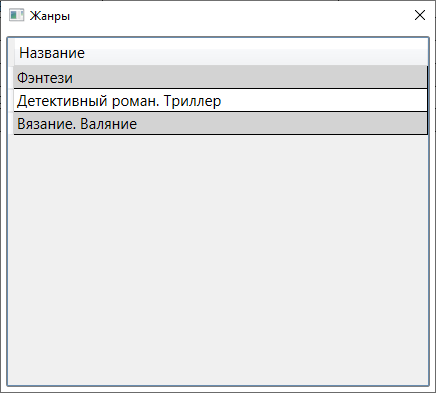


Рисунок 11 – Окно просмотра жанров

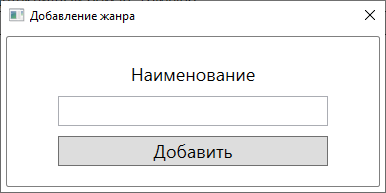


Рисунок 12 – Окно добавления жанра

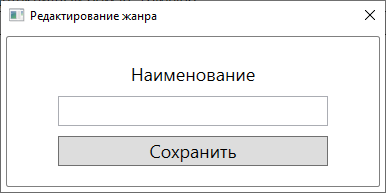


Рисунок 13 – Окно редактирования жанра

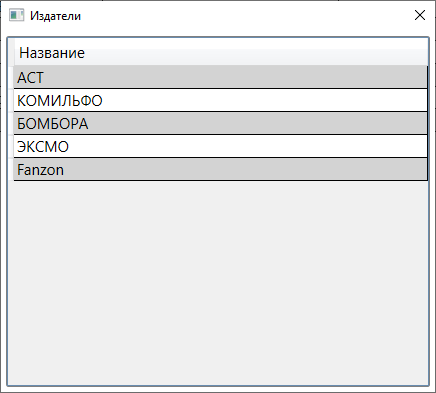


Рисунок 14 – Окно просмотра издателей

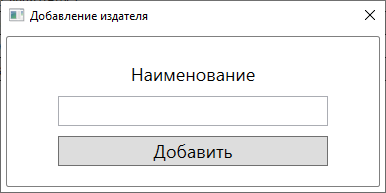


Рисунок 15 – Окно добавления издателя

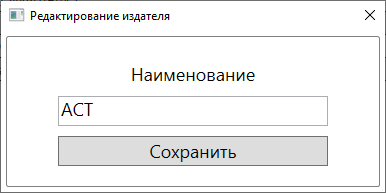


Рисунок 16 – Окно редактирования издателя

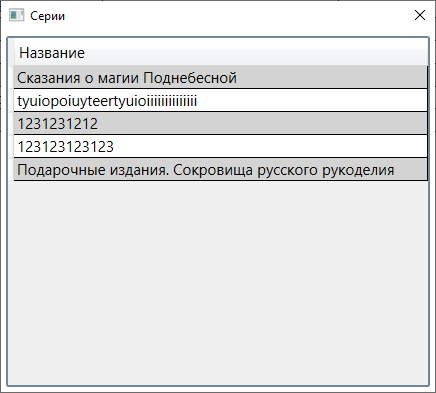


Рисунок 17 – Окно просмотра серий

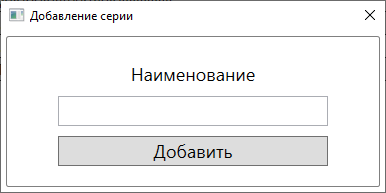


Рисунок 18 – Окно добавления серии

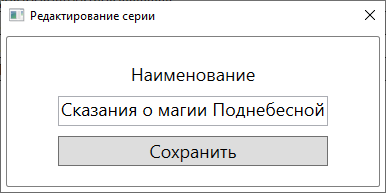


Рисунок 19 – Окно редактирования серии

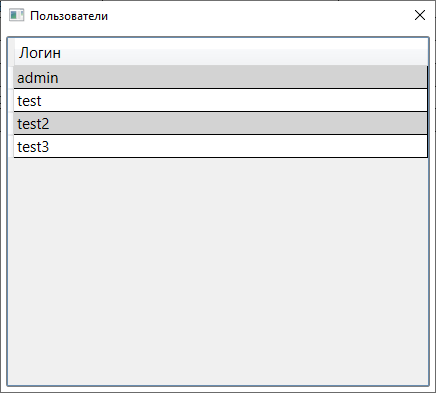


Рисунок 20 – Окно просмотра пользователей

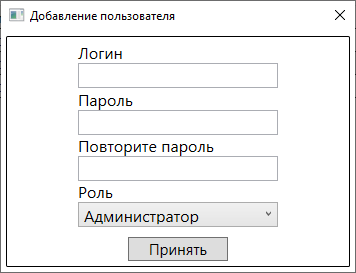


Рисунок 21 – Окно добавления пользователя

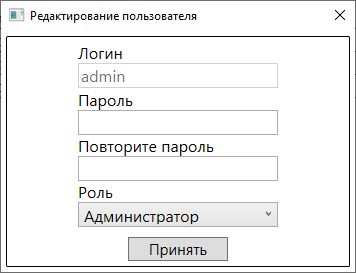


Рисунок 22 – Окно редактирования пользователя

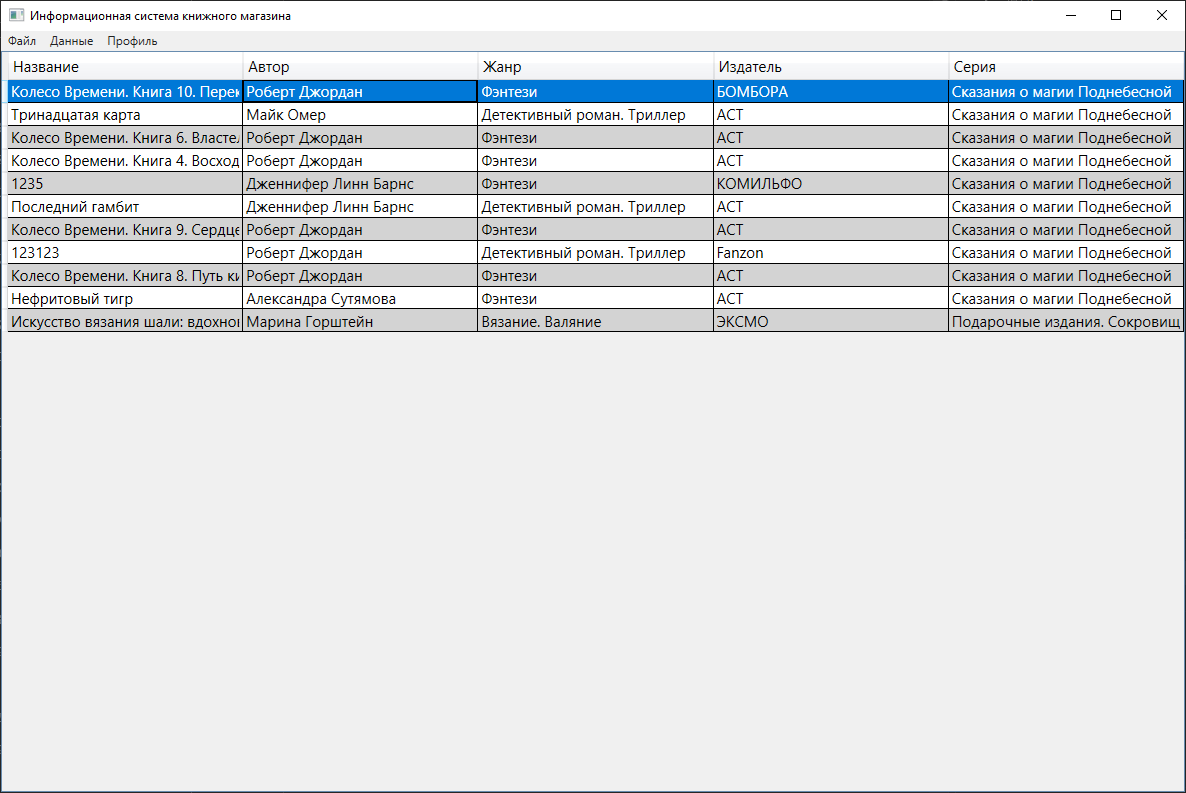


Рисунок 23 – Окно главного меню

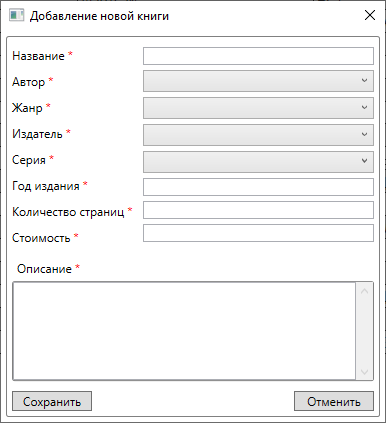


Рисунок 24 – окно добавления книги

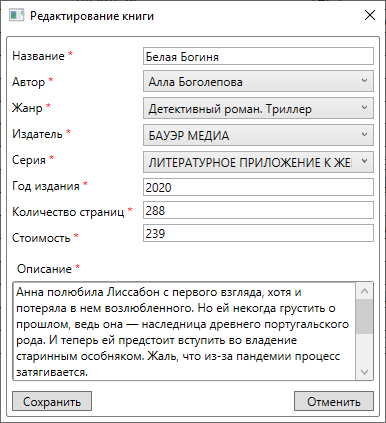


Рисунок 24 – окно редактирования книги

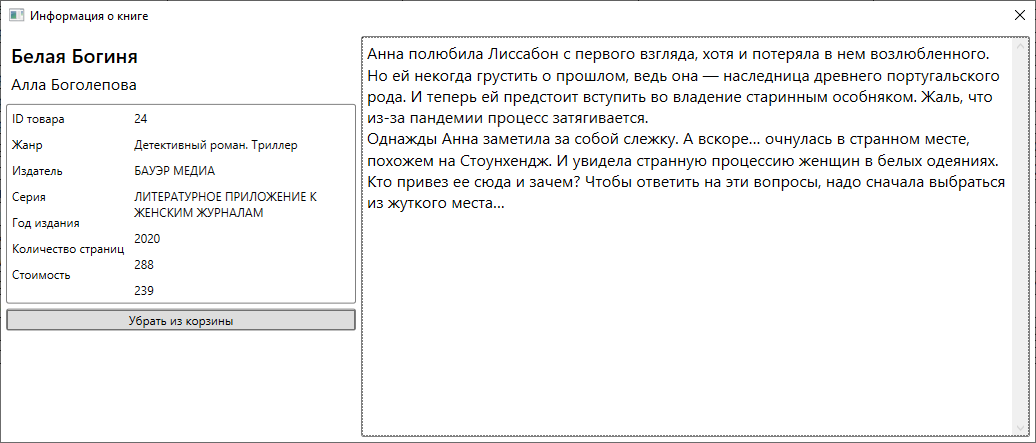


Рисунок 25 – окно просмотра книги

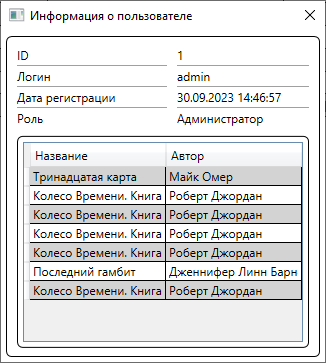


Рисунок 26 – окно просмотра профиля

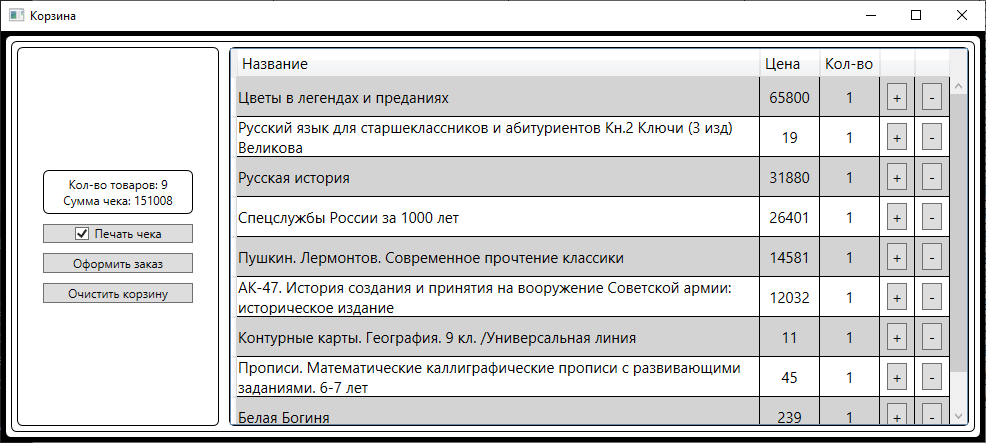


Рисунок 27 – окно просмотра корзины

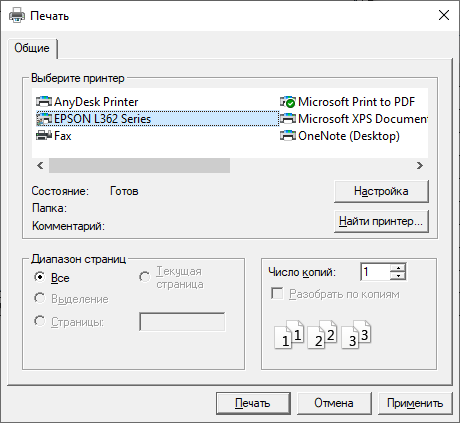


Рисунок 28 – окно печати чека

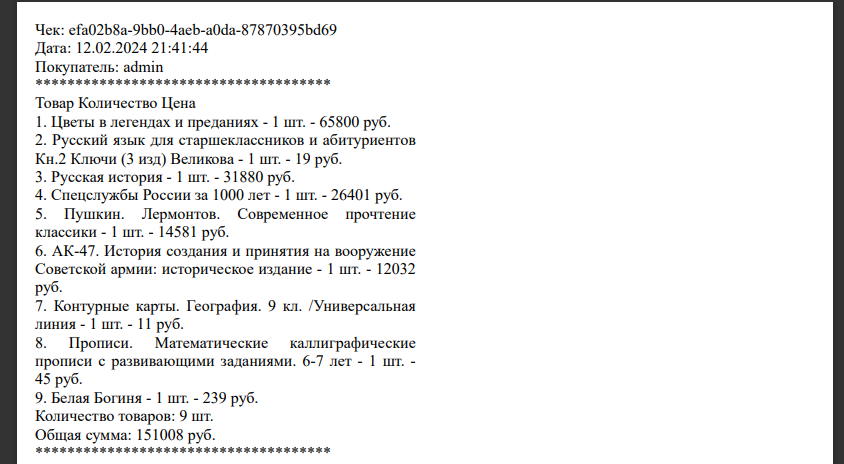


Рисунок 29 – содержимое чека

ЗАКЛЮЧЕНИЕ

Во время разработки курсового проекта были получены навыки:

* проектирования БД для ИС,
* создания физической модели в СУБД PostgreSQL Server,
* работы с системой для построения клиентских приложений WPF,
* работы с технологией NPgSQL.

В процессе выполнения курсового проекта:

* изучена предметная область,
* спроектирована база данных,
* разработано приложение,
* осуществлена защита от SQL инъекций.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Руководство по WPF [Электронный ресурс] – 2016 – URL: metanit.com/sharp/wpf (дата обращения 29.11.2023),
2. Полное руководство по языку программирования С# 8.0 и платформе .NET Core 3 [Электронный ресурс] – 2019 – URL: metanit.com/sharp/tutorial (дата обращения 29.11.2023),
3. Руководство по PostgreSQL Server 13 [Электронный ресурс] – 2017 – URL: https://metanit.com/sql/postgresql/ (дата обращения 29.11.2023),
4. Руководство по NPgSQL и работе с базами данных [Электронный ресурс] – 2015 – URL: https://metanit.com/sharp/entityframeworkcore/7.1.php (дата обращения 29.11.2023),
5. Уроки по C# и платформе .NET Framework [Электронный ресурс] – 2019 – URL: professorweb.ru (дата обращения 29.11.2023),
6. Техническая документация Microsoft [Электронный ресурс] – 2019 – URL: docs.microsoft.com/ru-ru (дата обращения 29.11.2023),
7. C# 6.0 Справочник. Полное описание языка. Джозеф Албахари и Бен Албахари,
8. WPF в .NET 4 с примерами на C# 2010 для профессионалов. Мэтью Мак-Дональд.

ПРИЛОЖЕНИЕ А

SQL скрипты на создание и заполнение базы данных, на создание учетных записей пользователей, на назначение привилегий

Запрос на создание базы данных:

**SET** statement\_timeout = 0;

**SET** lock\_timeout = 0;

**SET** idle\_in\_transaction\_session\_timeout = 0;

**SET** client\_encoding = 'UTF8';

**SET** standard\_conforming\_strings = **on**;

**SELECT** pg\_catalog.set\_config('search\_path', '', **false**);

**SET** check\_function\_bodies = **false**;

**SET** xmloption = content;

**SET** client\_min\_messages = warning;

**SET** row\_security = off;

**DROP** DATABASE booksystem;

--

-- TOC entry 3091 (class 1262 OID 1244241)

-- Name: booksystem; Type: DATABASE; Schema: -; Owner: postgres

--

**CREATE** DATABASE booksystem **WITH** TEMPLATE = template0 ENCODING = 'UTF8' LOCALE = 'Russian\_Russia.1251';

**ALTER** DATABASE booksystem OWNER **TO** postgres;

\**connect** booksystem

**SET** statement\_timeout = 0;

**SET** lock\_timeout = 0;

**SET** idle\_in\_transaction\_session\_timeout = 0;

**SET** client\_encoding = 'UTF8';

**SET** standard\_conforming\_strings = **on**;

**SELECT** pg\_catalog.set\_config('search\_path', '', **false**);

**SET** check\_function\_bodies = **false**;

**SET** xmloption = content;

**SET** client\_min\_messages = warning;

**SET** row\_security = off;

--

-- TOC entry 3 (class 2615 OID 2200)

-- Name: public; Type: SCHEMA; Schema: -; Owner: postgres

--

**CREATE** **SCHEMA** public;

**ALTER** **SCHEMA** public OWNER **TO** postgres;

--

-- TOC entry 3092 (class 0 OID 0)

-- Dependencies: 3

-- Name: SCHEMA public; Type: COMMENT; Schema: -; Owner: postgres

--

COMMENT **ON** **SCHEMA** public **IS** 'standard public schema';

**SET** default\_tablespace = '';

**SET** default\_table\_access\_method = heap;

--

-- TOC entry 203 (class 1259 OID 1244251)

-- Name: authors; Type: TABLE; Schema: public; Owner: postgres

--

**CREATE** **TABLE** public.authors (

author\_id **integer** **NOT** **NULL**,

author\_name text **NOT** **NULL**

);

**ALTER** **TABLE** public.authors OWNER **TO** postgres;

--

-- TOC entry 202 (class 1259 OID 1244249)

-- Name: authors\_author\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

**ALTER** **TABLE** public.authors **ALTER** **COLUMN** author\_id **ADD** **GENERATED** **ALWAYS** **AS** **IDENTITY** (

**SEQUENCE** NAME public.authors\_author\_id\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

CACHE 1

);

--

-- TOC entry 201 (class 1259 OID 1244244)

-- Name: books; Type: TABLE; Schema: public; Owner: postgres

--

**CREATE** **TABLE** public.books (

book\_id **integer** **NOT** **NULL**,

genre\_id **integer** **NOT** **NULL**,

author\_id **integer** **NOT** **NULL**,

book\_name text **NOT** **NULL**,

book\_description text,

publisher\_id **integer** **NOT** **NULL**,

series\_id **integer** **NOT** **NULL**,

book\_publish\_year **integer**,

book\_pages **integer**,

book\_price **integer** **DEFAULT** 0 **NOT** **NULL**

);

**ALTER** **TABLE** public.books OWNER **TO** postgres;

--

-- TOC entry 200 (class 1259 OID 1244242)

-- Name: books\_book\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

**ALTER** **TABLE** public.books **ALTER** **COLUMN** book\_id **ADD** **GENERATED** **ALWAYS** **AS** **IDENTITY** (

**SEQUENCE** NAME public.books\_book\_id\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

CACHE 1

);

--

-- TOC entry 217 (class 1259 OID 1301690)

-- Name: carts; Type: TABLE; Schema: public; Owner: postgres

--

**CREATE** **TABLE** public.carts (

user\_id **integer** **NOT** **NULL**,

book\_id **integer** **NOT** **NULL**,

primary\_id **integer** **NOT** **NULL**,

cart\_values **integer** **DEFAULT** 1 **NOT** **NULL**

);

**ALTER** **TABLE** public.carts OWNER **TO** postgres;

--

-- TOC entry 216 (class 1259 OID 1301688)

-- Name: carts\_primary\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

**ALTER** **TABLE** public.carts **ALTER** **COLUMN** primary\_id **ADD** **GENERATED** **ALWAYS** **AS** **IDENTITY** (

**SEQUENCE** NAME public.carts\_primary\_id\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

CACHE 1

);

--

-- TOC entry 205 (class 1259 OID 1244258)

-- Name: genres; Type: TABLE; Schema: public; Owner: postgres

--

**CREATE** **TABLE** public.genres (

genre\_id **integer** **NOT** **NULL**,

genre\_name text **NOT** **NULL**

);

**ALTER** **TABLE** public.genres OWNER **TO** postgres;

--

-- TOC entry 204 (class 1259 OID 1244256)

-- Name: genres\_genre\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

**ALTER** **TABLE** public.genres **ALTER** **COLUMN** genre\_id **ADD** **GENERATED** **ALWAYS** **AS** **IDENTITY** (

**SEQUENCE** NAME public.genres\_genre\_id\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

CACHE 1

);

--

-- TOC entry 207 (class 1259 OID 1244321)

-- Name: publishers; Type: TABLE; Schema: public; Owner: postgres

--

**CREATE** **TABLE** public.publishers (

publisher\_id **integer** **NOT** **NULL**,

publisher\_name text **NOT** **NULL**

);

**ALTER** **TABLE** public.publishers OWNER **TO** postgres;

--

-- TOC entry 206 (class 1259 OID 1244319)

-- Name: publishers\_publisher\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

**ALTER** **TABLE** public.publishers **ALTER** **COLUMN** publisher\_id **ADD** **GENERATED** **ALWAYS** **AS** **IDENTITY** (

**SEQUENCE** NAME public.publishers\_publisher\_id\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

CACHE 1

);

--

-- TOC entry 215 (class 1259 OID 1293504)

-- Name: roles; Type: TABLE; Schema: public; Owner: postgres

--

**CREATE** **TABLE** public.roles (

role\_id **integer** **NOT** **NULL**,

role\_name text **NOT** **NULL**,

role\_is\_admin **boolean** **NOT** **NULL**

);

**ALTER** **TABLE** public.roles OWNER **TO** postgres;

--

-- TOC entry 214 (class 1259 OID 1293502)

-- Name: roles\_role\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

**ALTER** **TABLE** public.roles **ALTER** **COLUMN** role\_id **ADD** **GENERATED** **ALWAYS** **AS** **IDENTITY** (

**SEQUENCE** NAME public.roles\_role\_id\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

CACHE 1

);

--

-- TOC entry 209 (class 1259 OID 1252513)

-- Name: series; Type: TABLE; Schema: public; Owner: postgres

--

**CREATE** **TABLE** public.series (

series\_id **integer** **NOT** **NULL**,

series\_name text **NOT** **NULL**

);

**ALTER** **TABLE** public.series OWNER **TO** postgres;

--

-- TOC entry 208 (class 1259 OID 1252511)

-- Name: series\_series\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

**ALTER** **TABLE** public.series **ALTER** **COLUMN** series\_id **ADD** **GENERATED** **ALWAYS** **AS** **IDENTITY** (

**SEQUENCE** NAME public.series\_series\_id\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

CACHE 1

);

--

-- TOC entry 211 (class 1259 OID 1293473)

-- Name: users; Type: TABLE; Schema: public; Owner: postgres

--

**CREATE** **TABLE** public.users (

user\_id **integer** **NOT** **NULL**,

user\_login text **NOT** **NULL**,

user\_password text **NOT** **NULL**,

user\_register\_date **timestamp** **without** **time** **zone** **DEFAULT** CURRENT\_TIMESTAMP **NOT** **NULL**,

role\_id **integer** **NOT** **NULL**

);

**ALTER** **TABLE** public.users OWNER **TO** postgres;

--

-- TOC entry 213 (class 1259 OID 1293485)

-- Name: users\_books; Type: TABLE; Schema: public; Owner: postgres

--

**CREATE** **TABLE** public.users\_books (

ub\_id **integer** **NOT** **NULL**,

user\_id **integer** **NOT** **NULL**,

book\_id **integer** **NOT** **NULL**,

ub\_add\_date **timestamp** **without** **time** **zone** **DEFAULT** CURRENT\_TIMESTAMP **NOT** **NULL**

);

**ALTER** **TABLE** public.users\_books OWNER **TO** postgres;

--

-- TOC entry 212 (class 1259 OID 1293483)

-- Name: users\_books\_ub\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

**ALTER** **TABLE** public.users\_books **ALTER** **COLUMN** ub\_id **ADD** **GENERATED** **ALWAYS** **AS** **IDENTITY** (

**SEQUENCE** NAME public.users\_books\_ub\_id\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

CACHE 1

);

--

-- TOC entry 210 (class 1259 OID 1293471)

-- Name: users\_user\_id\_seq; Type: SEQUENCE; Schema: public; Owner: postgres

--

**ALTER** **TABLE** public.users **ALTER** **COLUMN** user\_id **ADD** **GENERATED** **ALWAYS** **AS** **IDENTITY** (

**SEQUENCE** NAME public.users\_user\_id\_seq

**START** **WITH** 1

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

CACHE 1

);

--

-- TOC entry 3071 (class 0 OID 1244251)

-- Dependencies: 203

-- Data for Name: authors; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.authors (author\_id, author\_name) **FROM** stdin;

1 Майк Омер

2 Роберт Джордан

3 Дженнифер Линн Барнс

5 uipo[oiuyt

6 Александра Сутямова

7 Марина Горштейн

8 Без автора

9 Василий Ключевский

10 Иосиф Линдер

11 Джоан Кэтлин Роулинг

12 Елена Бурцева

13 Руслан Чумак

14 Людмила Великова

15 Алла Боголепова

\.

--

-- TOC entry 3069 (class 0 OID 1244244)

-- Dependencies: 201

-- Data for Name: books; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.books (book\_id, genre\_id, author\_id, book\_name, book\_description, publisher\_id, series\_id, book\_publish\_year, book\_pages, book\_price) **FROM** stdin;

7 2 2 Колесо Времени. Книга 6. Властелин хаоса Среди сообщества Айз Седай раскол. Признать Ранда ал’Тора Возрожденным Драконом или заклеймить его как самозванца? Темный властелин все силы прикладывает к тому, чтобы убедить Престол Амерлин в том, что Ранд — Лжедракон. Он внедряет в ряды Айз Седай своих приспешниц, наделяя их новыми телами, и плетет интригу за интригой. Мало того, Темный наводит на мир засуху и безумную жару, представляя все это происками Лжедракона Ранда ал’Тора. Найнив и Илэйн, оставившие Белую Башню после переворота, пускаются в долгий путь, чтобы отыскать истину… 1 1 4 4 1359

11 2 3 1235 123456 2 1 5 5 599

3 1 3 Последний гамбит Все, что осталось Эйвери Грэмбс сделать, чтобы получить наследство – прожить несколько недель в особняке Хоторнов до своего совершеннолетия. Но это не так-то просто, ведь с каждым днем напряжение нарастает. Папарацци преследуют ее на каждом шагу, и ее жизни постоянно грозит опасность. Только поддержка братьев помогает Эйвери все это выдержать. Она считала, что раскрыла все тайны семьи Хоторнов, пока на пороге ее дома внезапно не появился незваный гость. Осталось решить последнюю загадку. Эйвери и братьям вновь предстоит сразиться с могущественным противником. На этот раз на кон поставлено все. Игра началась. 1 1 6 6 619

1 1 1 Тринадцатая карта Детективы Ханна Шор и Бернард Глэдвин начинают расследование смерти Жаклин Мьюн, погибшей от двух пулевых ранений. Но то, что поначалу выглядит случайной стрельбой в неблагополучном районе, оказывается намного сложнее. Жаклин была экстрасенсом, предсказательницей судьбы и знахаркой, гадала на картах Таро и продавала экзотические снадобья, травы и масла. И очень много у кого вокруг были причины желать ее смерти.\r\nГадалка не смогла прочесть свое будущее по картам. Теперь детективам предстоит заглянуть в ее прошлое и выяснить личность загадочного убийцы… 1 1 2 2 469

14 2 6 Нефритовый тигр Древний Китай. Пять государств, что некогда были единым целым. Для Соны все это чуждо, но она оказалась втянута в войну двух династий, военные заговоры и дворцовые интриги. Ей предстоит распутать клубок истории и полностью переосмыслить прошлое. Может ли она доверять своим знаниям, окружающим ее странным людям или хотя бы самой себе? Так ли живет и дышит этот древний мир, как она себе представляла? И что за судьбу уготовил ей загадочный Белый бог, о котором она ничего не знает? 1 1 2023 320 599

2 2 2 Колесо Времени. Книга 10. Перекрестки сумерек В мире царят хаос и всеобщее недоверие. В раздоре и пожарах войны страны и континенты. Многим очевидно, что грядет Последняя битва с Темным, решающая для судьбы человечества, и в ней на стороне Света должен выступить Дракон Возрожденный. Но остается загадкой — что он задумал, что предпримет и где затаились его враги…\r\nЭгвейн, возведенная на Престол Амерлин теми из Айз Седай, что отказались признать Элайду, нынешнюю Амерлин в Белой Башне, встала с армией под стенами Тар Валона и грозит силой отобрать у Элайды власть… .\r\nИлэйн Траканд в Кэймлине, столице Андора, ведет борьбу за трон своей матери, стремясь заручиться поддержкой сторонников и отбить посягательства претендентки, которая, вступив в коалицию со знатными лордами королевства, взяла город в осаду…\r\nСам же Ранд, Дракон Возрожденный, в преддверии неминуемой схватки с Темным, стоит на пороге тяжелейшего выбора: заключать или нет перемирие с шончан, ведь сражаться одновременно с вторгшимися из-за океана захватчиками и с Тенью себе дороже…\r\nА еще — и это пугает всех, независимо от политических предпочтений — произошел чудовищный выброс Силы, последствия которого не знает никто…\r\nВ настоящем издании текст романа заново отредактирован и исправлен. 3 1 1 1 1249

12 1 2 123123 123123123 5 1 2023 338 599

5 2 2 Колесо Времени. Книга 8. Путь кинжалов: роман Монархи Пограничных земель, собрав войска и по древнему обычаю дав взаимную клятву, отправляются походом на юг, чтобы разобраться с Рандом ал’Тором, который взбаламутил полмира и объявил себя Возрожденным Драконом.\r\nТем временем Илэйн и Найнив, заполучив Чашу Ветров и заключив сделку с Морским народом, собираются избавить страну от безумной жары и засухи, наведенных Темным.\r\nПоложение поистине безысходное. Эбу Дар, важнейший морской порт континента, захватили явившиеся с моря шончанские войска и идут теперь с боями на Иллиан.\r\nРанд ал’Тор вступает в схватку с врагом. Но почему саидин, часть силы, которая правит Вселенной и вращает Колесо Времени, не подчиняется его воле, а Калландор, меч, который дается в руки только истинному Дракону, разит всех подряд — и чужих, и своих?..\r\nВ настоящем издании текст романа заново отредактирован и исправлен. 1 1 7 9 1249

6 2 2 Колесо Времени. Книга 4. Восходящая Тень Ранд ал’Тор, наконец-то завладевший созданным в древности мечом Калландором, который наделяет его обладателя невиданной силой, объявлен Драконом Возрожденным.\r\nИсполнилось одно из пророчеств, явился тот, кто призван спасти народы от Темного, источника и породителя хаоса. Но возрождение Дракона-спасителя создает не только новые нити в плетении мирового Узора, оно притягивает к Ранду ал’Тору пузыри зла, и Ранд становится мишенью для темных сил. Сам мир меняется резко, рвутся родственные связи, распадаются союзы, воины из Айильской пустыни захватывают Тирскую Твердыню, а в Белой Башне города Тар Валон происходит кровавый переворот. И есть много таких людей, кто не готов признавать за Рандом его новую ипостась, и эти люди опасны не менее прислужников Темного.\r\nВ настоящем издании текст романа «Дракон Возрожденный», как и в других романах, составивших знаменитую эпопею «Колесо Времени», заново отредактирован и исправлен. 1 1 3 3 1359

8 2 2 Колесо Времени. Книга 9. Сердце зимы Ранд ал’Тор, объявивший себя Возрожденным Драконом, скрываясь от прислужников Темного, собирается нанести ответный удар по Тени. В Белой Башне Тар Валона раскол: предательницы из Черных Айя готовят заговор, ну а друзья Ранда озабочены делами насущными. Перрин желает лишь одного — вырвать свою жену из айильского плена. Илэйн стремится уберечь от пожара войны родную страну. Мэт оказался в городе, захваченном явившейся из-за океана шончанской армией, и там судьба сводит его с Дочерью Девяти Лун, которая, как предначертано, должна стать его женой. И никому не ведомо, что ждет их всех впереди... В настоящем издании текст романа заново отредактирован и исправлен. 1 1 8 7 1249

15 4 7 Искусство вязания шали: вдохновение сибирского леса, 12 авторских проектов со схемами для вязания на спицах «Сибирская сойка», «Птицелов», «Сон-трава», «Марьин корень» — даже просто читая названия, чувствуешь смолистый запах кедровых шишек, шелк полевых трав под ногами, а лучи солнца с трудом прорываются сквозь высоченные кроны деревьев замысловатыми бликами. Чистота, ясность, многогранность, свежесть... Вся прелесть сибирской природы, порой слегка брутальной, а порой — сама нежность, в работах замечательного российского дизайнера Марины Горштейн!\r\n12 роскошных авторских шалей самых разных форм и моделей, связанных спицами с огромной любовью к сибирской природе. Формы шалей разнообразны: классическая треугольная, полукруглая, с горизонтальной каймой, с нупами, с бисером... Если начать от простого к сложному, с помощью книги вы станете асом и виртуозом. Независимо от вашего пола, возраста или навыков, каждый сможет связать свою собственную уникальную шаль!\r\n\r\nОб авторе:\r\nМарина Горштейн — замечательный российский мастер, дизайнер, вяжет уже много лет, более 5 лет обучает вязанию всех желающих. Работы Марины регулярно публикуются в СМИ, участвуют в творческих проектах и в выставках по всему миру. В 2021 году коллекции дизайнера были представлены на Детской неделе моды в Канаде. 4 6 2023 830 1779

16 5 8 Цветы в легендах и преданиях Не все дороги устланы цветами, но без цветов нет жизни и нет истории. «Там, где вырождаются цветы, не может жить человек» (Гегель).\r\n\r\nНе все люди праведники, не все тпицы райские, но все цветы восхитительны и совершенны. Садовые, полевые, лесные, даже числящиеся сорняками,\r\n\r\n«С поезда выйдешь — как окликают!\r\n\r\nПо полю дрожь.\r\n\r\nПоле пришпорено васильками,\r\n\r\nкак ни уходишь — все не уйдешь» (А Вознесенский).\r\n\r\nНаверное, можно было бы обойтись без цветов, но они свидетельствуют о нашей свободе, «ибо доказывают, что мы не скованы по рукам и ногам будничными заботами» (Р. Тагор).\r\n\r\nНикогда с такой любовью и целомудрием люди не думают о Мире и Мироздании, как тогда, когда смотрят на цветы. Любой из них заставляет вспомнить о небе. Любой из них наполняет нас восхищением пред удивительным замыслом Творца. И в любом из них великая тайна, недоступная и самым выдающимся мудрецам. Тайна, рождающая предания и легенды... 6 7 2009 544 65800

22 8 14 Русский язык для старшеклассников и абитуриентов Кн.2 Ключи (3 изд) Великова Специальный агент ФБР Мария Паркес, специалист по составлению психологических портретов, неутомимо идет по следу серийных убийц. Мария обладает даром медиума, каждую ночь она видит во сне убийства, точно передачи в прямом эфире, не имея возможности предотвратить ужасное действо. Благодаря своему дару она уже выследила несколько душегубов. На этот раз пропала помощница шерифа Рейчел, которая занималась расследованием исчезновения четырех молодых официанток. Следы Рейчел приводят Марию в лес, к развалинам старой церкви. То, что она увидела там, в подземелье, заставило ее похолодеть... 11 8 2016 168 19

17 6 9 Русская история Книга в коллекционном переплёте, изготовленном из натуральной гладкой кожи, украшенном тончайшим тиснением в русском стиле. На предней сторонке изображен св. Георгий Победоносец, поражающий змия, на корешке использован рукописный шрифт. Книга отпечатана на белой бумаге с лёгкой текстурой. Более 150-ти редких графических иллюстраций в тексте.\r\nЦветные иллюстрации отпечатаны на специальных листах из высокобелой мелованной бумаги, вшитых в книгу.\r\n4 вида тиснения - золото, серебро, красный натуральный пигмент и рельефное тиснение.\r\nФорзацы из дизайнерской тонированной бумаги. Обрез окрашен в цвет переплёта, торшонирован и вызолочен вручную.\r\nЛента для закладывания страниц и каптал, скрепляющий верх и низ объемного тома, изготовлены из шелка. К книге прилагается красочный сертификат, подтверждающий уникальность издания, он вложен в каждый экземпляр и пронумерован вручную. 4 8 2021 896 31880

18 6 10 Спецслужбы России за 1000 лет Любое государство только тогда может называться государством, когда оно способно обеспечить безопасность - свою и своих граждан - доступными ему методами. . .Книга, ставшая результатом продолжительного изучения архивных материалов, подробно рассказывает об истоках, формировании и развитии отечественных силовых, разведывательных и контртеррористических подразделений, о методах работы и практическом опыте спецслужб на различных этапах истории Российского государства. 7 9 2017 784 26401

19 7 12 Пушкин. Лермонтов. Современное прочтение классики В книге кандидата филологических наук Е.А. Бурцевой "Пушкин. Лермонтов. Современное прочтения классики" собраны статьи о жизни и творчестве великих поэтов 1 половины XIX века, сыгравших ключевую роль в становлении большой русской литературы. Автор рассматривает как хрестоматийные, так и малоизвестные широкому кругу читателей произведения в новом ключе, избегая характерного для прошлых литературоведческих эпох вульгарного социологизма. Особенно пристально автор анализирует некоторые литературные произведения в сравнительно-сопоставительном и текстологическом аспекте, такие как драму А.С. Пушкина "Борис Годунов", произведение, которое сам поэт называл главным в своем творчестве, романтическую поэму М.Ю. Лермонтова "Демон" над которой Лермонтов работал с юности и в которую вложил все свои самые сокровенные представления о добре и зле. При этом проводятся параллели и говорится о влиянии Пушкина и Лермонтова на писателей и поэтов последующих эпох - от Ф.М. Достоевского до М.А. Булгакова. 9 8 2017 156 14581

20 6 13 АК-47. История создания и принятия на вооружение Советской армии: историческое издание Автомат конструкции М. Т. Калашникова занимает особое место в истории стрелкового оружия мира. Он состоит на вооружении армии России и многих других стран мира уже более 70 лет. Несмотря на столь солидный срок службы, этот автомат и сейчас соответствует главным требованиям армии, а по безотказности остается непревзойденным до сих пор. Автомат Калашникова интересен огромной аудитории читателей, ему посвящено множество публикаций. В то же время за семь десятилетий боевого использования автомата Калашникова по-настоящему исчерпывающей истории его создания до сих пор не было написано, что не соответствует роли этого оружия в развитии отечественной оружейной школы. Настоящая книга призвана устранить это упущение. В книге описаны события начиная от раннего этапа творчества М. Т. Калашникова как конструктора-оружейника до включения его в конкурс на разработку автомата в 1946 г., запуска автомата АК-47 в серийное производство на Ижевских оружейных заводах в 1948-1948 гг. и его принятия на вооружение cоветской армии в 1949 г. Эти события освещены с учетом комплекса тактических, технических, экономических, военно-политических факторов и других событий, воздействовавших на разработку ручного автоматического оружия в СССР в начале-конце 1940-х гг. Книга насыщена изображениями опытных образцов оружия и другим документальным и иллюстративным материалом, раскрывающим суть и ход описываемых событий. В книге проведен технический анализ опытных автоматов М. Т. Калашникова и автоматов, разработанных другими конструкторами, приведены изображения их внешнего вида и внутреннего устройства. Часть опубликованных в книге образцов оружия представляется общественности впервые. Книга написана на основе уникальной коллекции опытного советского автоматического оружия, хранящейся в Военно-историческом музее артиллерии, инженерных войск и войск связи. Документальную основу материала книги составляют источники из архивных фондов Военно-исторического музея артиллерии, инженерных войск и войск связи, Центрального архива Министерства обороны РФ, других архивов и музеев РФ, а также из Архива Президента Республики Казахстан. Рекомендуем книгу всем читателям, интересующимся историей советского стрелкового оружия. 10 10 2021 600 12032

21 8 8 Контурные карты. География. 9 кл. /Универсальная линия Контурные карты содержат необходимый набор карт по курсу географии соответствующего класса и могут быть использованы с любым УМК. 11 8 2017 16 11

23 9 8 Прописи. Математические каллиграфические прописи с развивающими заданиями. 6-7 лет В пособии предложены задания, направленные на формирование навыков написания цифр у дошкольников и младших школьников. По этой тетради ребенок научится правильно и красиво писать цифры и математические элементы. Выполняя занимательные задания и графические упражнения в игровой форме, дошкольник научится считать в пределах десяти, писать цифры по клеточкам и точкам. В процессе выполнения разнообразных заданий развиваются внимание. Увлекательные задания, собранные в книжке, развивают мелкую моторику, внимание, память, логическое мышление - качества, которые помогут хорошо учиться в школе, расширяют кругозор и словарный запас.\r\nПредназначено дошкольникам и учащимся начальной школы; полезно родителям, которые хотят помочь своим детям подготовиться к школе. 12 11 2018 19 45

24 1 15 Белая Богиня Анна полюбила Лиссабон с первого взгляда, хотя и потеряла в нем возлюбленного. Но ей некогда грустить о прошлом, ведь она — наследница древнего португальского рода. И теперь ей предстоит вступить во владение старинным особняком. Жаль, что из-за пандемии процесс затягивается.\r\nОднажды Анна заметила за собой слежку. А вскоре… очнулась в странном месте, похожем на Стоунхендж. И увидела странную процессию женщин в белых одеяниях.\r\nКто привез ее сюда и зачем? Чтобы ответить на эти вопросы, надо сначала выбраться из жуткого места… 13 12 2020 288 239

\.

--

-- TOC entry 3085 (class 0 OID 1301690)

-- Dependencies: 217

-- Data for Name: carts; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.carts (user\_id, book\_id, primary\_id, cart\_values) **FROM** stdin;

\.

--

-- TOC entry 3073 (class 0 OID 1244258)

-- Dependencies: 205

-- Data for Name: genres; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.genres (genre\_id, genre\_name) **FROM** stdin;

2 Фэнтези

1 Детективный роман. Триллер

4 Вязание. Валяние

5 Художественная литература. Фольклор

6 История. Общество

7 Литературоведение. Фольклористика

8 Образование

9 Дошкольное образование

\.

--

-- TOC entry 3075 (class 0 OID 1244321)

-- Dependencies: 207

-- Data for Name: publishers; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.publishers (publisher\_id, publisher\_name) **FROM** stdin;

1 АСТ

2 КОМИЛЬФО

3 БОМБОРА

4 ЭКСМО

5 Fanzon

6 ИЗДАТЕЛЬСКИЙ ДОМ "ЭКОНОМИЧЕСКА"

7 РИПОЛ-КЛАССИК

8 BLOOMSBURY

9 LAP LAMBERT ACADEMIC PUBLISHIN

10 АТЛАНТ

11 ПРОСВЕЩЕНИЕ

12 УЧИТЕЛЬ

13 БАУЭР МЕДИА

\.

--

-- TOC entry 3083 (class 0 OID 1293504)

-- Dependencies: 215

-- Data for Name: roles; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.roles (role\_id, role\_name, role\_is\_admin) **FROM** stdin;

1 Администратор t

3 Пользователь f

\.

--

-- TOC entry 3077 (class 0 OID 1252513)

-- Dependencies: 209

-- Data for Name: series; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.series (series\_id, series\_name) **FROM** stdin;

1 Сказания о магии Поднебесной

2 tyuiopoiuyteertyuioiiiiiiiiiiiiii

3 1231231212

5 123123123123

6 Подарочные издания. Сокровища русского рукоделия

7 РУССКАЯ КЛАССИЧЕСКАЯ БИБЛИОТЕКА "ЭКОНОМИКА И ДУХОВНОСТЬ"

8 Без серии

9 Золотая коллекция

10 ОРУЖЕЙНАЯ АКАДЕМИЯ

11 Дошкольный тренажер

12 ЛИТЕРАТУРНОЕ ПРИЛОЖЕНИЕ К ЖЕНСКИМ ЖУРНАЛАМ

\.

--

-- TOC entry 3079 (class 0 OID 1293473)

-- Dependencies: 211

-- Data for Name: users; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.users (user\_id, user\_login, user\_password, user\_register\_date, role\_id) **FROM** stdin;

1 admin admin 2023-09-30 14:46:57.797922 1

2 test test 2023-10-04 12:21:24.197097 3

3 test2 test 2023-11-17 14:33:47.240031 1

4 test3 test 2023-11-17 14:37:22.734188 3

6 123123 123123 2024-02-02 13:02:32.824017 3

\.

--

-- TOC entry 3081 (class 0 OID 1293485)

-- Dependencies: 213

-- Data for Name: users\_books; Type: TABLE DATA; Schema: public; Owner: postgres

--

COPY public.users\_books (ub\_id, user\_id, book\_id, ub\_add\_date) **FROM** stdin;

2 1 1 2023-11-21 20:23:27.996091

3 1 6 2023-11-24 17:42:58.806824

6 1 7 2023-11-24 17:48:47.178263

9 1 5 2023-11-24 17:49:05.984353

11 1 8 2023-11-24 17:49:12.214326

12 1 3 2023-11-24 17:49:14.535901

15 2 12 2023-11-24 19:06:39.550127

16 1 15 2024-02-08 20:54:54.672475

17 1 8 2024-02-08 20:54:54.676232

18 1 14 2024-02-08 20:54:54.676814

19 1 2 2024-02-08 20:54:54.677337

20 1 3 2024-02-08 20:54:54.677908

21 1 14 2024-02-08 21:16:43.686865

22 1 8 2024-02-08 21:16:43.691821

23 1 6 2024-02-08 21:16:43.69235

24 1 6 2024-02-08 21:17:08.203655

25 1 5 2024-02-08 21:17:08.210228

26 1 14 2024-02-08 21:17:08.210923

27 1 14 2024-02-08 21:18:12.255071

28 1 15 2024-02-08 21:18:12.262062

29 1 1 2024-02-08 21:18:12.262685

30 1 15 2024-02-09 12:58:49.344401

31 1 8 2024-02-12 20:04:23.294153

32 1 6 2024-02-12 20:04:23.298989

33 1 15 2024-02-12 20:04:23.29944

34 1 2 2024-02-12 20:04:23.299797

35 1 14 2024-02-12 20:04:23.300263

36 1 1 2024-02-12 20:04:23.300617

37 1 3 2024-02-12 20:04:23.301149

38 1 16 2024-02-12 21:41:52.915794

39 1 22 2024-02-12 21:41:52.921853

40 1 17 2024-02-12 21:41:52.922401

41 1 18 2024-02-12 21:41:52.922873

42 1 19 2024-02-12 21:41:52.923243

43 1 20 2024-02-12 21:41:52.923651

44 1 21 2024-02-12 21:41:52.924095

45 1 23 2024-02-12 21:41:52.924545

46 1 24 2024-02-12 21:41:52.925015

\.

--

-- TOC entry 3093 (class 0 OID 0)

-- Dependencies: 202

-- Name: authors\_author\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

**SELECT** pg\_catalog.setval('public.authors\_author\_id\_seq', 15, **true**);

--

-- TOC entry 3094 (class 0 OID 0)

-- Dependencies: 200

-- Name: books\_book\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

**SELECT** pg\_catalog.setval('public.books\_book\_id\_seq', 24, **true**);

--

-- TOC entry 3095 (class 0 OID 0)

-- Dependencies: 216

-- Name: carts\_primary\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

**SELECT** pg\_catalog.setval('public.carts\_primary\_id\_seq', 41, **true**);

--

-- TOC entry 3096 (class 0 OID 0)

-- Dependencies: 204

-- Name: genres\_genre\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

**SELECT** pg\_catalog.setval('public.genres\_genre\_id\_seq', 9, **true**);

--

-- TOC entry 3097 (class 0 OID 0)

-- Dependencies: 206

-- Name: publishers\_publisher\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

**SELECT** pg\_catalog.setval('public.publishers\_publisher\_id\_seq', 13, **true**);

--

-- TOC entry 3098 (class 0 OID 0)

-- Dependencies: 214

-- Name: roles\_role\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

**SELECT** pg\_catalog.setval('public.roles\_role\_id\_seq', 3, **true**);

--

-- TOC entry 3099 (class 0 OID 0)

-- Dependencies: 208

-- Name: series\_series\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

**SELECT** pg\_catalog.setval('public.series\_series\_id\_seq', 12, **true**);

--

-- TOC entry 3100 (class 0 OID 0)

-- Dependencies: 212

-- Name: users\_books\_ub\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

**SELECT** pg\_catalog.setval('public.users\_books\_ub\_id\_seq', 46, **true**);

--

-- TOC entry 3101 (class 0 OID 0)

-- Dependencies: 210

-- Name: users\_user\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres

--

**SELECT** pg\_catalog.setval('public.users\_user\_id\_seq', 6, **true**);

--

-- TOC entry 2912 (class 2606 OID 1244255)

-- Name: authors authors\_pk; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.authors

**ADD** **CONSTRAINT** authors\_pk **PRIMARY** **KEY** (author\_id);

--

-- TOC entry 2910 (class 2606 OID 1244248)

-- Name: books books\_pk; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.books

**ADD** **CONSTRAINT** books\_pk **PRIMARY** **KEY** (book\_id);

--

-- TOC entry 2928 (class 2606 OID 1301694)

-- Name: carts carts\_pk; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.carts

**ADD** **CONSTRAINT** carts\_pk **PRIMARY** **KEY** (primary\_id);

--

-- TOC entry 2914 (class 2606 OID 1244262)

-- Name: genres genres\_pk; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.genres

**ADD** **CONSTRAINT** genres\_pk **PRIMARY** **KEY** (genre\_id);

--

-- TOC entry 2916 (class 2606 OID 1244328)

-- Name: publishers publishers\_pk; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.publishers

**ADD** **CONSTRAINT** publishers\_pk **PRIMARY** **KEY** (publisher\_id);

--

-- TOC entry 2926 (class 2606 OID 1293511)

-- Name: roles roles\_pk; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.roles

**ADD** **CONSTRAINT** roles\_pk **PRIMARY** **KEY** (role\_id);

--

-- TOC entry 2918 (class 2606 OID 1252520)

-- Name: series series\_pk; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.series

**ADD** **CONSTRAINT** series\_pk **PRIMARY** **KEY** (series\_id);

--

-- TOC entry 2924 (class 2606 OID 1293490)

-- Name: users\_books users\_books\_pk; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.users\_books

**ADD** **CONSTRAINT** users\_books\_pk **PRIMARY** **KEY** (ub\_id);

--

-- TOC entry 2920 (class 2606 OID 1293480)

-- Name: users users\_pk; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.users

**ADD** **CONSTRAINT** users\_pk **PRIMARY** **KEY** (user\_id);

--

-- TOC entry 2922 (class 2606 OID 1293482)

-- Name: users users\_un; Type: CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.users

**ADD** **CONSTRAINT** users\_un **UNIQUE** (user\_login);

--

-- TOC entry 2929 (class 2606 OID 1244272)

-- Name: books books\_fk; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.books

**ADD** **CONSTRAINT** books\_fk **FOREIGN** **KEY** (genre\_id) **REFERENCES** public.genres(genre\_id) **ON** **UPDATE** **CASCADE** **ON** **DELETE** **CASCADE**;

--

-- TOC entry 2930 (class 2606 OID 1244282)

-- Name: books books\_fk1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.books

**ADD** **CONSTRAINT** books\_fk1 **FOREIGN** **KEY** (author\_id) **REFERENCES** public.authors(author\_id) **ON** **UPDATE** **CASCADE** **ON** **DELETE** **CASCADE**;

--

-- TOC entry 2931 (class 2606 OID 1244329)

-- Name: books books\_fk2; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.books

**ADD** **CONSTRAINT** books\_fk2 **FOREIGN** **KEY** (publisher\_id) **REFERENCES** public.publishers(publisher\_id) **ON** **UPDATE** **CASCADE** **ON** **DELETE** **CASCADE**;

--

-- TOC entry 2932 (class 2606 OID 1252521)

-- Name: books books\_fk3; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.books

**ADD** **CONSTRAINT** books\_fk3 **FOREIGN** **KEY** (series\_id) **REFERENCES** public.series(series\_id) **ON** **UPDATE** **CASCADE** **ON** **DELETE** **CASCADE**;

--

-- TOC entry 2936 (class 2606 OID 1301695)

-- Name: carts carts\_fk; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.carts

**ADD** **CONSTRAINT** carts\_fk **FOREIGN** **KEY** (user\_id) **REFERENCES** public.users(user\_id) **ON** **UPDATE** **CASCADE** **ON** **DELETE** **CASCADE**;

--

-- TOC entry 2937 (class 2606 OID 1301700)

-- Name: carts carts\_fk\_1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.carts

**ADD** **CONSTRAINT** carts\_fk\_1 **FOREIGN** **KEY** (book\_id) **REFERENCES** public.books(book\_id) **ON** **UPDATE** **CASCADE** **ON** **DELETE** **CASCADE**;

--

-- TOC entry 2934 (class 2606 OID 1293491)

-- Name: users\_books users\_books\_fk; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.users\_books

**ADD** **CONSTRAINT** users\_books\_fk **FOREIGN** **KEY** (book\_id) **REFERENCES** public.books(book\_id) **ON** **UPDATE** **CASCADE** **ON** **DELETE** **CASCADE**;

--

-- TOC entry 2935 (class 2606 OID 1293496)

-- Name: users\_books users\_books\_fk\_1; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.users\_books

**ADD** **CONSTRAINT** users\_books\_fk\_1 **FOREIGN** **KEY** (user\_id) **REFERENCES** public.users(user\_id) **ON** **UPDATE** **CASCADE** **ON** **DELETE** **CASCADE**;

--

-- TOC entry 2933 (class 2606 OID 1293512)

-- Name: users users\_fk; Type: FK CONSTRAINT; Schema: public; Owner: postgres

--

**ALTER** **TABLE** **ONLY** public.users

**ADD** **CONSTRAINT** users\_fk **FOREIGN** **KEY** (role\_id) **REFERENCES** public.roles(role\_id) **ON** **UPDATE** **CASCADE** **ON** **DELETE** **CASCADE**;

ПРИЛОЖЕНИЕ Б

Код программы

Код программы состоит из следующих модулей:

* authorization.xaml
* authorization.xaml.cs
* authorsAdd.xaml
* authorsAdd.xaml.cs
* authorsEdit.xaml
* authorsEdit.xaml.cs
* authorsView.xaml
* authorsView.xaml.cs
* bookAdd.xaml
* bookAdd.xaml.cs
* bookEdit.xaml
* bookEdit.xaml.cs
* bookView.xaml
* bookView.xaml.cs
* data.cs
* database.cs
* genresAdd.xaml
* genresAdd.xaml.cs
* genresEdit.xaml
* genresEdit.xaml.cs
* genresView.xaml
* genresView.xaml.cs
* MainWindow.xaml
* MainWindow.xaml.cs
* publisherAdd.xaml
* publisherAdd.xaml.cs
* publisherEdit.xaml
* publisherEdit.xaml.cs
* publishersView.xaml
* publishersView.xaml.cs
* seriesAdd.xaml
* seriesAdd.xaml.cs
* seriesEdit.xaml
* seriesEdit.xaml.cs
* seriesView.xaml
* seriesView.xaml.cs
* userAdd.xaml
* userAdd.xaml.cs
* userEdit.xaml
* userEdit.xaml.cs
* userProfile.xaml
* userProfile.xaml.cs
* userRegister.xaml
* userRegister.xaml.cs
* usersView.xaml
* usersView.xaml.cs
* cartView.xaml
* cartView.xaml.cs

В модуле authorization.xaml содержится декларативное описание интерфейса окна авторизации на языке XAML:

<Window x:Class="bookSystem.authorization"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Авторизация в системе" Height="200" Width="320" ResizeMode="CanMinimize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<StackPanel VerticalAlignment="Center" Width="170">

<TextBlock Text="Логин" FontSize="16"/>

<TextBox x:Name="authorizationLogin" FontSize="18" Height="25"/>

<TextBlock Text="Пароль" FontSize="16"/>

<PasswordBox FontSize="18" Height="25" x:Name="AuthorizationPassword"/>

<Button Content="Войти" FontSize="16" Margin="0,5,0,0" x:Name="logIn" Click="logIn\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле authorization.xaml.cs содержится описание работы окна авторизации на языке C#:

using System.Windows;

namespace bookSystem {

public partial class authorization : Window {

public authorization() {

InitializeComponent();

}

private void logIn\_Click(object sender, RoutedEventArgs e) {

if (AuthorizationPassword.Password != string.Empty && authorizationLogin.Text != string.Empty) {

data.Users user = data.users.Find(u => u.User\_Login == authorizationLogin.Text && u.User\_Password == AuthorizationPassword.Password);

if (user.User\_Login != null) {

this.Close();

data.currentUser = user;

MainWindow mainWindow = (MainWindow)this.Owner;

mainWindow.enableUserActivity();

}

else {

MessageBox.Show("Неправильный логин или пароль!");

}

}

}

}

}

В модуле authorsAdd.xaml содержится декларативное описание интерфейса окна добавления автора на языке XAML:

<Window x:Class="bookSystem.authorsAdd"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Добавление автора" Height="200" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<StackPanel VerticalAlignment="Center" Width="270">

<Label Content="Наименование" FontSize="18" HorizontalContentAlignment="Center"/>

<TextBox x:Name="authorName" FontSize="18" Height="30" Margin="0,5,0,0"/>

<Button Content="Добавить" FontSize="18" Height="30" Margin="0,10,0,0" x:Name="buttonAdd" Click="buttonAdd\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле authorsAdd.xaml.cs содержится описание работы добавления автора на языке C#:

using System.Windows;

namespace bookSystem {

public partial class authorsAdd : Window {

public authorsAdd() {

InitializeComponent();

}

private void buttonAdd\_Click(object sender, RoutedEventArgs e) {

if (authorName.Text != null) {

data.Author author = data.authors.Find(a => a.Author\_Name == authorName.Text);

if (author.Author\_Name != null) {

MessageBox.Show("Автор уже существует в базе данных!");

}

else {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.authorAdd(authorName.Text);

db.loadAuthors();

db.closeConnection();

authorsView authorsView = (authorsView)this.Owner;

authorsView.dataGridSetItemSource();

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

}

else {

MessageBox.Show("Необходимо указать автора!");

}

}

}

}

В модуле authorsEdit.xaml содержится декларативное описание интерфейса окна редактирования автора на языке XAML:

<Window x:Class="bookSystem.authorsEdit"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Редактирование автора" Height="200" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<StackPanel VerticalAlignment="Center" Width="270">

<Label Content="Наименование" FontSize="18" HorizontalContentAlignment="Center"/>

<TextBox x:Name="authorName" FontSize="18" Height="30" Margin="0,5,0,0"/>

<Button Content="Сохранить" FontSize="18" Height="30" Margin="0,10,0,0" x:Name="buttonSave" Click="buttonSave\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле authorsEdit.xaml.cs содержится описание работы окна редактирования автора на языке C#:

using System.Windows;

namespace bookSystem {

public partial class authorsEdit : Window {

data.Author \_\_author;

public authorsEdit(data.Author author) {

InitializeComponent();

this.\_\_author = author;

authorName.Text = \_\_author.Author\_Name;

}

private void buttonSave\_Click(object sender, RoutedEventArgs e) {

if (authorName.Text != null) {

data.Author author = data.authors.Find(a => a.Author\_Name == authorName.Text);

if (author.Author\_Name != null) {

MessageBox.Show("Автор уже существует в базе данных!");

}

else {

\_\_author.Author\_Name = authorName.Text;

database db = new database();

if (db.openConnection(db.connectionString)) {

db.authorEdit(\_\_author);

db.loadAuthors();

db.loadBooks();

db.closeConnection();

authorsView authorsView = (authorsView)this.Owner;

authorsView.dataGridSetItemSource();

MainWindow mainWindow = (MainWindow)authorsView.Owner;

mainWindow.dataGridSetItemSource(data.books);

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

}

else {

MessageBox.Show("Необходимо указать автора!");

}

}

}

}

В модуле authorsView.xaml содержится декларативное описание интерфейса окна просмотра авторов на языке XAML:

<Window x:Class="bookSystem.authorsView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Авторы" Height="400" Width="450" MinHeight="400" MinWidth="350" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<DockPanel>

<DataGrid x:Name="dataGrid" IsReadOnly="True" AutoGenerateColumns="False" FontSize="15" SelectionMode="Single" RowBackground="LightGray" AlternatingRowBackground="White">

<DataGrid.ContextMenu>

<ContextMenu x:Name="dataGridContextMenu" Visibility="Visible">

<MenuItem Header="Добавить" x:Name="dataGridAuthorAdd" Click="dataGridAuthorAdd\_Click"/>

<Separator/>

<MenuItem Header="Редактировать" x:Name="dataGridAuthorEdit" Click="dataGridAuthorEdit\_Click" />

<MenuItem Header="Удалить" x:Name="dataGridAuthorDelete" Click="dataGridAuthorDelete\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Binding="{Binding Path=Author\_Name}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле authorsView.xaml.cs содержится описание работы окна просмотра авторов на языке C#:

using System.Windows;

using System.Windows.Controls;

namespace bookSystem {

public partial class authorsView : Window {

public authorsView() {

InitializeComponent();

dataGrid.ItemsSource = data.authors;

}

private void dataGridAuthorAdd\_Click(object sender, RoutedEventArgs e) {

authorsAdd authorsAdd = new authorsAdd();

authorsAdd.Owner = this;

authorsAdd.ShowDialog();

}

public void dataGridSetItemSource() {

dataGrid.ClearValue(ItemsControl.ItemsSourceProperty);

dataGrid.ItemsSource = data.authors;

}

private void dataGridAuthorEdit\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

authorsEdit authorsEdit = new authorsEdit((data.Author)dataGrid.SelectedItem);

authorsEdit.Owner = this;

authorsEdit.ShowDialog();

}

}

private void dataGridAuthorDelete\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

data.Author author = (data.Author)dataGrid.SelectedItem;

database db = new database();

if (db.openConnection(db.connectionString)) {

db.authorDelete(author.Author\_Id);

db.loadAuthors();

db.loadBooks();

dataGridSetItemSource();

MainWindow mainWindow = (MainWindow)this.Owner;

mainWindow.dataGridSetItemSource(data.books);

}

}

}

}

}

В модуле bookAdd.xaml содержится декларативное описание интерфейса окна добавления книги на языке XAML:

<Window x:Class="bookSystem.bookAdd"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Добавление новой книги" Height="430" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<DockPanel>

<DockPanel DockPanel.Dock="Bottom">

<StackPanel DockPanel.Dock="Left" Margin="5,5,5,5">

<Button x:Name="buttonSave" Content="Сохранить" Width="80" Click="buttonSave\_Click"/>

</StackPanel>

<StackPanel DockPanel.Dock="Right" Margin="5,5,5,5" HorizontalAlignment="Right">

<Button x:Name="buttonCancel" Content="Отменить" Width="80"/>

</StackPanel>

</DockPanel>

<Border DockPanel.Dock="Bottom" Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2" Height="100">

<DockPanel>

<ScrollViewer>

<TextBox x:Name="bookDescription" TextWrapping="Wrap" AcceptsReturn="True"/>

</ScrollViewer>

</DockPanel>

</Border>

<TextBlock DockPanel.Dock="Bottom" Padding="10,0,0,0">

<Run Text="Описание"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<StackPanel Margin="5,5,5,5" DockPanel.Dock="Left">

<TextBlock Margin="0,5,0,5">

<Run Text="Название"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Автор"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Жанр"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Издатель"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Серия"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Год издания"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Количество страниц"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Стоимость"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

</StackPanel>

<StackPanel Margin="5,5,5,5" DockPanel.Dock="Right">

<TextBox x:Name="bookName" Margin="0,5,0,5"/>

<ComboBox x:Name="bookAuthor" Margin="0,0,0,0" DisplayMemberPath="Author\_Name"/>

<ComboBox x:Name="bookGenre" Margin="0,5,0,0" DisplayMemberPath="Genre\_Name"/>

<ComboBox x:Name="bookPublisher" Margin="0,5,0,0" DisplayMemberPath="Publisher\_Name"/>

<ComboBox x:Name="bookSeries" Margin="0,5,0,0" DisplayMemberPath="Series\_Name"/>

<TextBox x:Name="bookPublishYear" Margin="0,5,0,0"/>

<TextBox x:Name="bookPages" Margin="0,5,0,5"/>

<TextBox x:Name="bookPrice" Margin="0,0,0,5"/>

</StackPanel>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле bookAdd.xaml.cs содержится описание работы окна добавления книги на языке C#:

using System;

using System.Windows;

namespace bookSystem {

public partial class bookAdd : Window {

public bookAdd() {

InitializeComponent();

bookAuthor.ItemsSource = data.authors;

bookGenre.ItemsSource = data.genres;

bookPublisher.ItemsSource = data.publishers;

bookSeries.ItemsSource = data.series;

}

private void buttonSave\_Click(object sender, RoutedEventArgs e) {

if (bookName.Text.Length > 0) {

if (bookAuthor.SelectedItem != null) {

if (bookGenre.SelectedItem != null) {

if (bookPublisher.SelectedItem != null) {

if (bookSeries.SelectedItem != null) {

if (bookPublishYear.Text.Length > 0) {

if (bookPages.Text.Length > 0) {

if (bookDescription.Text.Length > 0) {

if (Convert.ToInt32(bookPrice.Text) > 0) {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.bookAdd(new data.Book {

Book\_Name = bookName.Text

, Book\_Description = bookDescription.Text

, Author = (data.Author)bookAuthor.SelectedItem

, Genre = (data.Genre)bookGenre.SelectedItem

, Publisher = (data.Publisher)bookPublisher.SelectedItem

, Series = (data.Series)bookSeries.SelectedItem

, Book\_Publish\_Year = Convert.ToInt32(bookPublishYear.Text)

, Book\_Pages = Convert.ToInt32(bookPages.Text)

, Book\_Price = Convert.ToInt32(bookPrice.Text)

});

db.loadBooks();

db.closeConnection();

MainWindow mainWindow = (MainWindow)this.Owner;

mainWindow.dataGridSetItemSource(data.books);

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

else {

}

}

else {

MessageBox.Show("Необходимо указать издателя для книги!");

}

}

else {

MessageBox.Show("Необходимо указать количество страниц для книги!");

}

}

else {

MessageBox.Show("Необходимо указать год издания для книги!");

}

}

else {

MessageBox.Show("Необходимо указать серию для книги!");

}

}

else {

MessageBox.Show("Необходимо указать описание для книги!");

}

}

else {

MessageBox.Show("Необходимо указать жанр книги!");

}

}

else {

MessageBox.Show("Необходимо указать автора книги!");

}

}

else {

MessageBox.Show("Необходимо указать название для книги!");

}

}

}

}

В модуле bookEdit.xaml содержится декларативное описание интерфейса окна редактирования книги на языке XAML:

<Window x:Class="bookSystem.bookEdit"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Редактирование книги" Height="430" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<DockPanel>

<DockPanel DockPanel.Dock="Bottom">

<StackPanel DockPanel.Dock="Left" Margin="5,5,5,5">

<Button x:Name="buttonSave" Content="Сохранить" Width="80" Click="buttonSave\_Click"/>

</StackPanel>

<StackPanel DockPanel.Dock="Right" Margin="5,5,5,5" HorizontalAlignment="Right">

<Button x:Name="buttonCancel" Content="Отменить" Width="80"/>

</StackPanel>

</DockPanel>

<Border DockPanel.Dock="Bottom" Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2" Height="100">

<DockPanel>

<ScrollViewer>

<TextBox x:Name="bookDescription" TextWrapping="Wrap" AcceptsReturn="True"/>

</ScrollViewer>

</DockPanel>

</Border>

<TextBlock DockPanel.Dock="Bottom" Padding="10,0,0,0">

<Run Text="Описание"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<StackPanel Margin="5,5,5,5" DockPanel.Dock="Left">

<TextBlock Margin="0,5,0,5">

<Run Text="Название"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Автор"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Жанр"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Издатель"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Серия"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Год издания"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Количество страниц"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

<TextBlock Margin="0,5,0,5">

<Run Text="Стоимость"/>

<Run Text="\*" Foreground="Red"/>

</TextBlock>

</StackPanel>

<StackPanel Margin="5,5,5,5" DockPanel.Dock="Right">

<TextBox x:Name="bookName" Margin="0,5,0,5"/>

<ComboBox x:Name="bookAuthor" Margin="0,0,0,0" DisplayMemberPath="Author\_Name"/>

<ComboBox x:Name="bookGenre" Margin="0,5,0,0" DisplayMemberPath="Genre\_Name"/>

<ComboBox x:Name="bookPublisher" Margin="0,5,0,0" DisplayMemberPath="Publisher\_Name"/>

<ComboBox x:Name="bookSeries" Margin="0,5,0,0" DisplayMemberPath="Series\_Name"/>

<TextBox x:Name="bookPublishYear" Margin="0,5,0,0"/>

<TextBox x:Name="bookPages" Margin="0,5,0,5"/>

<TextBox x:Name="bookPrice" Margin="0,0,0,5"/>

</StackPanel>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле bookEdit.xaml.cs содержится описание работы окна редактирования книги на языке C#:

using System;

using System.Windows;

namespace bookSystem {

public partial class bookEdit : Window {

private int bookId { get; set; }

public bookEdit(data.Book book) {

InitializeComponent();

bookGenre.ItemsSource = data.genres;

bookGenre.Text = book.Genre.Genre\_Name;

bookAuthor.ItemsSource = data.authors;

bookAuthor.Text = book.Author.Author\_Name;

bookName.Text = book.Book\_Name;

bookDescription.Text = book.Book\_Description;

bookPublisher.ItemsSource = data.publishers;

bookPublisher.Text = book.Publisher.Publisher\_Name;

bookSeries.ItemsSource = data.series;

bookSeries.Text = book.Series.Series\_Name;

bookPublishYear.Text = book.Book\_Publish\_Year.ToString();

bookPages.Text = book.Book\_Pages.ToString();

bookPrice.Text = book.Book\_Price.ToString();

bookId = book.Book\_Id;

}

private void buttonSave\_Click(object sender, RoutedEventArgs e) {

if (bookName.Text.Length > 0) {

if (bookAuthor.SelectedItem != null) {

if (bookGenre.SelectedItem != null) {

if (bookPublisher.SelectedItem != null) {

if (bookSeries.SelectedItem != null) {

if (bookPublishYear.Text.Length > 0) {

if (bookPages.Text.Length > 0) {

if (bookDescription.Text.Length > 0) {

if (Convert.ToInt32(bookPrice.Text) > 0) {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.bookEdit(new data.Book {

Book\_Id = bookId

, Book\_Name = bookName.Text

, Book\_Description = bookDescription.Text

, Author = (data.Author)bookAuthor.SelectedItem

, Genre = (data.Genre)bookGenre.SelectedItem

, Publisher = (data.Publisher)bookPublisher.SelectedItem

, Series = (data.Series)bookSeries.SelectedItem

, Book\_Publish\_Year = Convert.ToInt32(bookPublishYear.Text)

, Book\_Pages = Convert.ToInt32(bookPages.Text)

, Book\_Price = Convert.ToInt32(bookPrice.Text)

});

db.loadBooks();

db.closeConnection();

MainWindow mainWindow = (MainWindow)this.Owner;

mainWindow.dataGridSetItemSource(data.books);

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

}

else {

MessageBox.Show("Необходимо указать описание для книги!");

}

}

else {

MessageBox.Show("Необходимо указать количество страниц для книги!");

}

}

else {

MessageBox.Show("Необходимо указать год издания для книги!");

}

}

else {

MessageBox.Show("Необходимо указать издателя для книги!");

}

}

else {

MessageBox.Show("Необходимо указать издателя книги!");

}

}

else {

MessageBox.Show("Необходимо указать жанр книги!");

}

}

else {

MessageBox.Show("Необходимо указать автора книги!");

}

}

else {

MessageBox.Show("Необходимо указать название для книги!");

}

}

}

}

В модуле bookView.xaml содержится декларативное описание интерфейса окна просмотра книги на языке XAML:

<Window x:Class="bookSystem.bookView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Информация о книге" Height="450" Width="1050" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<DockPanel>

<StackPanel Margin="5,5,0,5" DockPanel.Dock="Left" Width="350">

<TextBlock x:Name="bookName" TextWrapping="Wrap" Text="Последний гамбит" FontSize="20" FontWeight="DemiBold" Margin="5,5,5,5"/>

<TextBlock x:Name="bookAuthorName" TextWrapping="Wrap" Text="Дженнифер Линн Барнс" FontSize="16" Margin="5,0,0,10"/>

<Border BorderBrush="Gray" BorderThickness="1" CornerRadius="2">

<DockPanel>

<StackPanel DockPanel.Dock="Left">

<Label Content="ID товара"/>

<Label Content="Жанр"/>

<Label Content="Издатель"/>

<Label Content="Серия"/>

<Label Content="Год издания"/>

<Label Content="Количество страниц"/>

<Label Content="Стоимость"/>

</StackPanel>

<StackPanel DockPanel.Dock="Right">

<TextBlock x:Name="bookId" TextWrapping="Wrap" Text="2" Margin="5"/>

<TextBlock x:Name="bookGenreName" TextWrapping="Wrap" Text="Детективный роман. Триллер" Margin="5"/>

<TextBlock x:Name="bookPublisherName" TextWrapping="Wrap" Text="АСТ" Margin="5"/>

<TextBlock x:Name="bookSeriesName" TextWrapping="Wrap" Text="Сказания о магии Поднебесной" Margin="5"/>

<TextBlock x:Name="bookPublishYear" TextWrapping="Wrap" Text="2023" Margin="5"/>

<TextBlock x:Name="bookPages" TextWrapping="Wrap" Text="222" Margin="5"/>

<TextBlock x:Name="bookPrice" TextWrapping="Wrap" Text="1249" Margin="5"/>

</StackPanel>

</DockPanel>

</Border>

<Border BorderBrush="Gray" BorderThickness="1" CornerRadius="2" Margin="0,5,0,0">

<Button x:Name="buttonAddBook" Content="Добавить в корзину" Click="buttonAddBook\_Click"/>

</Border>

</StackPanel>

<Border DockPanel.Dock="Right" Margin="5" BorderBrush="Gray" BorderThickness="1" CornerRadius="2">

<ScrollViewer>

<TextBlock x:Name="bookDescription" TextWrapping="Wrap" FontSize="16" Margin="5"/>

</ScrollViewer>

</Border>

</DockPanel>

</Grid>

</Window>

В модуле bookView.xaml.cs содержится описание работы окна просмотра книги на языке C#:

using System.Windows;

namespace bookSystem {

public partial class bookView : Window {

data.Book book;

data.Carts checkCart;

public bookView(data.Book book) {

InitializeComponent();

this.book = book;

bookName.Text = book.Book\_Name;

bookAuthorName.Text = book.Author.Author\_Name;

bookId.Text = book.Book\_Id.ToString();

bookGenreName.Text = book.Genre.Genre\_Name;

bookPublisherName.Text = book.Publisher.Publisher\_Name;

bookSeriesName.Text = book.Series.Series\_Name;

bookDescription.Text = book.Book\_Description;

bookPublishYear.Text = book.Book\_Publish\_Year.ToString();

bookPages.Text = book.Book\_Pages.ToString();

bookPrice.Text = book.Book\_Price.ToString();

if (data.currentUser.User\_Login != null) {

buttonAddBook.IsEnabled = true;

checkCart = data.carts.Find(c => c.User\_Id == data.currentUser.User\_Id && c.Book\_Id == book.Book\_Id);

if (checkCart.Primary\_Id != 0) {

buttonAddBook.Content = "Убрать из корзины";

}

else {

buttonAddBook.Content = "Добавить в корзину";

}

}

else {

buttonAddBook.IsEnabled = false;

}

}

private void buttonAddBook\_Click(object sender, RoutedEventArgs e) {

if (checkCart.Primary\_Id != 0) {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.cartRemove(checkCart);

db.loadCart();

}

}

else {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.cartAdd(data.currentUser.User\_Id, book.Book\_Id);

db.loadCart();

}

}

this.Close();

}

}

}

В модуле data.cs содержится описание работы с хранимыми в памяти программы данными на языке C#:

using System;

using System.Collections.Generic;

namespace bookSystem {

public class data {

public struct Book {

public int Book\_Id { get; set; }

public Genre Genre { get; set; }

public Author Author { get; set; }

public Publisher Publisher { get; set; }

public Series Series { get; set; }

public string Book\_Name { get; set; }

public string Book\_Description { get; set; }

public int Book\_Publish\_Year { get; set; }

public int Book\_Pages { get; set; }

public int Book\_Price { get; set; }

}

public struct Genre {

public int Genre\_Id { get; set; }

public string Genre\_Name { get; set; }

}

public struct Author {

public int Author\_Id { get; set; }

public string Author\_Name { get; set; }

}

public struct Publisher {

public int Publisher\_Id { get; set; }

public string Publisher\_Name { get; set; }

}

public struct Series {

public int Series\_Id { get; set; }

public string Series\_Name { get; set;}

}

public struct Users {

public int User\_Id { get; set; }

public string User\_Login { get; set; }

public string User\_Password { get; set; }

public DateTime User\_Register\_Date { get; set; }

public Roles Role { get; set; }

}

public struct Roles {

public int Role\_Id { get; set; }

public string Role\_Name { get; set; }

public bool Role\_Is\_Admin { get; set; }

}

public struct UsersBooks {

public int UB\_Id { get; set; }

public int UB\_User\_Id { get; set; }

public int UB\_Book\_Id { get; set; }

public DateTime UB\_Add\_Date { get; set; }

}

public struct Carts {

public int Primary\_Id { get; set; }

public int User\_Id { get; set; }

public int Book\_Id { get; set; }

public int Cart\_Values { get; set; }

public Book Book { get; set; }

}

public static List<Book> books = new List<Book>();

public static List<Genre> genres = new List<Genre>();

public static List<Author> authors = new List<Author>();

public static List<Publisher> publishers = new List<Publisher>();

public static List<Series> series = new List<Series>();

public static List<Users> users = new List<Users>();

public static List<Roles> roles = new List<Roles>();

public static List<UsersBooks> usersBooks = new List<UsersBooks>();

public static List<Carts> carts = new List<Carts>();

public static Users currentUser;

}

}

В модуле database.cs содержится описание работы с базой данных на языке C#:

using Npgsql;

using System;

using System.Windows;

namespace bookSystem {

internal class database {

public NpgsqlConnection Connection { get; set; }

public NpgsqlCommand Command { get; set; }

public NpgsqlDataReader Reader { get; set; }

public string connectionString { get; set; } = "Server=localhost;Port=5432;User Id=postgres;Password=;Database=booksystem";

public bool openConnection(string connectionString) {

Connection = new NpgsqlConnection(connectionString);

try {

Connection.Open();

return true;

}

catch {

return false;

}

}

public bool closeConnection() {

try {

Connection.Close();

return true;

}

catch {

return false;

}

}

public bool executeQuery(string query) {

try {

Command = new NpgsqlCommand(query, Connection);

Command.ExecuteNonQuery();

return true;

}

catch (NpgsqlException e) {

MessageBox.Show($"[E] Ошибка при выполнении запроса в базу данных!\n\n\n{e.Message}");

return false;

}

}

public bool executeReader(string query) {

try {

Command = new NpgsqlCommand(query, Connection);

Reader = Command.ExecuteReader();

return true;

}

catch (NpgsqlException e) {

MessageBox.Show($"[R] Ошибка при выполнении запроса в базу данных!\n\n\n{e.Message}");

return false;

}

}

public void loadGenres() {

data.genres.Clear();

string sql\_query = "select \* from public.genres";

executeReader(sql\_query);

if (Reader.HasRows) {

while (Reader.Read()) {

data.genres.Add(new data.Genre {

Genre\_Id = (int)Reader["genre\_id"]

, Genre\_Name = (string)Reader["genre\_name"]

});

}

}

Reader.Close();

}

public void loadAuthors() {

data.authors.Clear();

string sql\_query = "select \* from public.authors";

executeReader(sql\_query);

if (Reader.HasRows) {

while (Reader.Read()) {

data.authors.Add(new data.Author {

Author\_Id = (int)Reader["author\_id"]

, Author\_Name = (string)Reader["author\_name"]

});

}

}

Reader.Close();

}

public void loadPublishers() {

data.publishers.Clear();

string sql\_query = "select \* from public.publishers";

executeReader(sql\_query);

if (Reader.HasRows) {

while (Reader.Read()) {

data.publishers.Add(new data.Publisher {

Publisher\_Id = (int)Reader["publisher\_id"]

, Publisher\_Name = (string)Reader["publisher\_name"]

});

}

}

Reader.Close();

}

public void loadSeries() {

data.series.Clear();

string sql\_query = "select \* from public.series";

executeReader(sql\_query);

if (Reader.HasRows) {

while (Reader.Read()) {

data.series.Add(new data.Series {

Series\_Id = (int)Reader["series\_id"]

, Series\_Name = (string)Reader["series\_name"]

});

}

}

Reader.Close();

}

public void loadBooks() {

data.books.Clear();

string sql\_query = "select \* from public.books";

executeReader(sql\_query);

if (Reader.HasRows) {

while (Reader.Read()) {

data.books.Add(new data.Book {

Book\_Id = (int)Reader["book\_id"]

, Book\_Name = (string)Reader["book\_name"]

, Book\_Description = (string)Reader["book\_description"]

, Genre = data.genres.Find(g => g.Genre\_Id == (int)Reader["genre\_id"])

, Author = data.authors.Find(a => a.Author\_Id == (int)Reader["author\_id"])

, Publisher = data.publishers.Find(p => p.Publisher\_Id == (int)Reader["publisher\_id"])

, Series = data.series.Find(s => s.Series\_Id == (int)Reader["series\_id"])

, Book\_Publish\_Year = (int)Reader["book\_publish\_year"]

, Book\_Pages = (int)Reader["book\_pages"]

, Book\_Price = (int)Reader["book\_price"]

});

}

}

Reader.Close();

}

public void loadRoles() {

data.roles.Clear();

string sql\_query = "select \* from public.roles";

executeReader(sql\_query);

if (Reader.HasRows) {

while (Reader.Read()) {

data.roles.Add(

new data.Roles {

Role\_Id = (int)Reader["role\_id"]

, Role\_Name = (string)Reader["role\_name"]

, Role\_Is\_Admin = (bool)Reader["role\_is\_admin"]

}

);

}

}

Reader.Close();

}

public void loadUsers() {

data.users.Clear();

string sql\_query = "select \* from public.users";

executeReader(sql\_query);

if (Reader.HasRows) {

while (Reader.Read()) {

data.users.Add( new data.Users {

User\_Id = (int)Reader["user\_id"]

, User\_Login = (string)Reader["user\_login"]

, User\_Password = (string)Reader["user\_password"]

, User\_Register\_Date = (DateTime)Reader["user\_register\_date"]

, Role = data.roles.Find(r => r.Role\_Id == (int)Reader["role\_id"])

} );

}

}

Reader.Close();

}

public void loadUserBooks() {

data.usersBooks.Clear();

string sql\_query = "select \* from public.users\_books";

executeReader(sql\_query);

if (Reader.HasRows) {

while (Reader.Read()) {

data.usersBooks.Add(new data.UsersBooks {

UB\_Id = (int)Reader["ub\_id"]

, UB\_User\_Id = (int)Reader["user\_id"]

, UB\_Book\_Id = (int)Reader["book\_id"]

, UB\_Add\_Date = (DateTime)Reader["ub\_add\_date"]

});

}

}

Reader.Close();

}

public void bookAdd(data.Book book) {

book.Book\_Name = checkEscapeChars(book.Book\_Name);

book.Book\_Description = checkEscapeChars(book.Book\_Description);

string sql\_query = "insert into public.books" +

"(book\_name, author\_id, genre\_id, book\_description, publisher\_id, series\_id, book\_publish\_year, book\_pages, book\_price) values" +

$"('{book.Book\_Name}', '{book.Author.Author\_Id}', '{book.Genre.Genre\_Id}', '{book.Book\_Description}', '{book.Publisher.Publisher\_Id}', '{book.Series.Series\_Id}', '{book.Book\_Publish\_Year}', '{book.Book\_Pages}', '{book.Book\_Price}')";

executeQuery(sql\_query);

}

public void bookEdit(data.Book book) {

book.Book\_Name = checkEscapeChars(book.Book\_Name);

book.Book\_Description = checkEscapeChars(book.Book\_Description);

string sql\_query = "update public.books " +

$"set genre\_id = '{book.Genre.Genre\_Id}', author\_id = '{book.Author.Author\_Id}', book\_name = '{book.Book\_Name}', book\_description = '{book.Book\_Description}', publisher\_id = '{book.Publisher.Publisher\_Id}', series\_id = '{book.Series.Series\_Id}', book\_publish\_year = '{book.Book\_Publish\_Year}', book\_pages = '{book.Book\_Pages}', book\_price = '{book.Book\_Price}' " +

$"where book\_id = '{book.Book\_Id}'";

executeQuery(sql\_query);

}

public void bookDelete(int bookId) {

string sql\_query = $"delete from public.books where book\_id = '{bookId}'";

executeQuery(sql\_query);

}

private string checkEscapeChars(string query) {

return query.Replace("'", "''");

}

public void authorAdd(string authorName) {

authorName = checkEscapeChars(authorName);

string sql\_query = $"insert into public.authors (author\_name) values ('{authorName}')";

executeQuery(sql\_query);

}

public void authorEdit(data.Author author) {

author.Author\_Name = checkEscapeChars(author.Author\_Name);

string sql\_query = $"update public.authors set author\_name = '{author.Author\_Name}' where author\_id = '{author.Author\_Id}'";

executeQuery(sql\_query);

}

public void authorDelete(int authorId) {

string sql\_query = $"delete from public.authors where author\_id = '{authorId}'";

executeQuery(sql\_query);

}

public void genreAdd(string genreName) {

genreName = checkEscapeChars(genreName);

string sql\_query = $"insert into public.genres (genre\_name) values ('{genreName}')";

executeQuery(sql\_query);

}

public void genreEdit(data.Genre genre) {

genre.Genre\_Name = checkEscapeChars(genre.Genre\_Name);

string sql\_query = $"update public.genres set genre\_name = '{genre.Genre\_Name}' where genre\_id = '{genre.Genre\_Id}'";

executeQuery(sql\_query);

}

public void genreDelete(int genreId) {

string sql\_query = $"delete from public.genres where genre\_id = '{genreId}'";

executeQuery(sql\_query);

}

public void publisherAdd(string publisherName) {

publisherName = checkEscapeChars(publisherName);

string sql\_query = $"insert into public.publishers (publisher\_name) values ('{publisherName}')";

executeQuery(sql\_query);

}

public void publisherEdit(data.Publisher publisher) {

publisher.Publisher\_Name = checkEscapeChars(publisher.Publisher\_Name);

string sql\_query = $"update public.publishers set publisher\_name = '{publisher.Publisher\_Name}' where publisher\_id = '{publisher.Publisher\_Id}'";

executeQuery(sql\_query);

}

public void publisherDelete(int publisherId) {

string sql\_query = $"delete from public.publishers where publisher\_id = '{publisherId}'";

executeQuery(sql\_query);

}

public void seriesAdd(string seriesName) {

seriesName = checkEscapeChars(seriesName);

string sql\_query = $"insert into public.series (series\_name) values ('{seriesName}')";

executeQuery(sql\_query);

}

public void seriesEdit(data.Series series) {

series.Series\_Name = checkEscapeChars(series.Series\_Name);

string sql\_query = $"update public.series set series\_name = '{series.Series\_Name}' where series\_id = '{series.Series\_Id}'";

executeQuery(sql\_query);

}

public void seriesDelete(int seriesId) {

string sql\_query = $"delete from public.series where series\_id = '{seriesId}'";

executeQuery(sql\_query);

}

public void userAdd(data.Users user) {

user.User\_Login = checkEscapeChars(user.User\_Login);

user.User\_Password = checkEscapeChars(user.User\_Password);

string sql\_query = $"insert into public.users (user\_login, user\_password, role\_id) values ('{user.User\_Login}', '{user.User\_Password}', '{user.Role.Role\_Id}')";

executeQuery(sql\_query);

}

public void userEdit(data.Users user) {

user.User\_Password = checkEscapeChars(user.User\_Password);

string sql\_query = $"update public.users set user\_password = '{user.User\_Password}', role\_id = '{user.Role.Role\_Id}' where user\_id = '{user.User\_Id}'";

executeQuery(sql\_query);

}

public void userDelete(int userId) {

string sql\_query = $"delete from public.users where user\_id = '{userId}'";

executeQuery(sql\_query);

}

public void userBookAdd(int userId, int bookId) {

string sql\_query = $"insert into public.users\_books (user\_id, book\_id) values ('{userId}', '{bookId}')";

executeQuery(sql\_query);

}

public void userBookDelete(data.UsersBooks userBook) {

string sql\_query = $"delete from public.users\_books where ub\_id = '{userBook.UB\_Id}'";

executeQuery(sql\_query);

}

public void cartAdd(int userId, int bookId) {

string sql\_query = $"insert into public.carts (user\_id, book\_id) values ('{userId}', '{bookId}')";

executeQuery(sql\_query);

}

public void cartRemove(data.Carts cart) {

string sql\_query = $"delete from public.carts where primary\_id = '{cart.Primary\_Id}'";

executeQuery(sql\_query);

}

public void cartAddValue(data.Carts cart) {

string sql\_query = $"update public.carts set cart\_values = cart\_values + '1' where primary\_id = '{cart.Primary\_Id}'";

executeQuery(sql\_query);

}

public void cartRemoveValue(data.Carts cart) {

string sql\_query = $"update public.carts set cart\_values = cart\_values - '1' where primary\_id = '{cart.Primary\_Id}'";

executeQuery(sql\_query);

}

public void loadCart() {

data.carts.Clear();

string sql\_query = "select \* from public.carts order by primary\_id";

executeReader(sql\_query);

if (Reader.HasRows) {

while (Reader.Read()) {

data.carts.Add(new data.Carts {

Primary\_Id = (int)Reader["primary\_id"]

, User\_Id = (int)Reader["user\_id"]

, Book\_Id = (int)Reader["book\_id"]

, Cart\_Values = (int)Reader["cart\_values"]

, Book = data.books.Find(b => b.Book\_Id == (int)Reader["book\_id"])

});

}

}

Reader.Close();

}

}

}

В модуле genresAdd.xaml содержится декларативное описание интерфейса окна добавления жанра на языке XAML:

<Window x:Class="bookSystem.genresAdd"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Добавление жанра" Height="200" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<StackPanel VerticalAlignment="Center" Width="270">

<Label Content="Наименование" FontSize="18" HorizontalContentAlignment="Center"/>

<TextBox x:Name="genreName" FontSize="18" Height="30" Margin="0,5,0,0"/>

<Button Content="Добавить" FontSize="18" Height="30" Margin="0,10,0,0" x:Name="buttonAdd" Click="buttonAdd\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле genresAdd.xaml.cs содержится описание работы окна добавления жанра на языке C#:

using System.Windows;

namespace bookSystem {

public partial class genresAdd : Window {

public genresAdd() {

InitializeComponent();

}

private void buttonAdd\_Click(object sender, RoutedEventArgs e) {

if (genreName.Text != null) {

data.Genre genre = data.genres.Find(a => a.Genre\_Name == genreName.Text);

if (genre.Genre\_Name != null) {

MessageBox.Show("Жанр уже существует в базе данных!");

}

else {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.genreAdd(genreName.Text);

db.loadGenres();

db.closeConnection();

genresView genresView = (genresView)this.Owner;

genresView.dataGridSetItemSource();

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

}

else {

MessageBox.Show("Необходимо указать жанр!");

}

}

}

}

В модуле genresEdit.xaml содержится декларативное описание интерфейса окна редактирования жанра на языке XAML:

<Window x:Class="bookSystem.genresEdit"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Редактирование жанра" Height="200" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<StackPanel VerticalAlignment="Center" Width="270">

<Label Content="Наименование" FontSize="18" HorizontalContentAlignment="Center"/>

<TextBox x:Name="genreName" FontSize="18" Height="30" Margin="0,5,0,0"/>

<Button Content="Сохранить" FontSize="18" Height="30" Margin="0,10,0,0" x:Name="buttonSave" Click="buttonSave\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле genresEdit.xaml.cs содержится описание работы окна редактирования жанра на языке C#:

using System.Windows;

namespace bookSystem {

public partial class genresEdit : Window {

data.Genre \_\_genre;

public genresEdit(data.Genre genre) {

InitializeComponent();

this.\_\_genre = genre;

}

private void buttonSave\_Click(object sender, RoutedEventArgs e) {

if (genreName.Text != null) {

data.Genre genre = data.genres.Find(a => a.Genre\_Name == genreName.Text);

if (genre.Genre\_Name != null) {

MessageBox.Show("Жанр уже существует в базе данных!");

}

else {

\_\_genre.Genre\_Name = genreName.Text;

database db = new database();

if (db.openConnection(db.connectionString)) {

db.genreEdit(\_\_genre);

db.loadGenres();

db.loadBooks();

db.closeConnection();

genresView genresView = (genresView)this.Owner;

genresView.dataGridSetItemSource();

MainWindow mainWindow = (MainWindow)genresView.Owner;

mainWindow.dataGridSetItemSource(data.books);

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

}

else {

MessageBox.Show("Необходимо указать жанр!");

}

}

}

}

В модуле genresView.xaml содержится декларативное описание интерфейса окна просмотра жанров на языке XAML:

<Window x:Class="bookSystem.genresView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Жанры" Height="400" Width="450" MinHeight="400" MinWidth="350" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<DockPanel>

<DataGrid x:Name="dataGrid" IsReadOnly="True" AutoGenerateColumns="False" FontSize="15" SelectionMode="Single" RowBackground="LightGray" AlternatingRowBackground="White">

<DataGrid.ContextMenu>

<ContextMenu x:Name="dataGridContextMenu" Visibility="Visible">

<MenuItem Header="Добавить" x:Name="dataGridGenreAdd" Click="dataGridGenreAdd\_Click"/>

<Separator/>

<MenuItem Header="Редактировать" x:Name="dataGridGenreEdit" Click="dataGridGenreEdit\_Click"/>

<MenuItem Header="Удалить" x:Name="dataGridGenreDelete" Click="dataGridGenreDelete\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Binding="{Binding Path=Genre\_Name}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле genresView.xaml.cs содержится описание работы окна просмотра жанров на языке C#:

using System.Windows;

using System.Windows.Controls;

namespace bookSystem {

public partial class genresView : Window {

public genresView() {

InitializeComponent();

dataGrid.ItemsSource = data.genres;

}

public void dataGridSetItemSource() {

dataGrid.ClearValue(ItemsControl.ItemsSourceProperty);

dataGrid.ItemsSource = data.genres;

}

private void dataGridGenreAdd\_Click(object sender, RoutedEventArgs e) {

genresAdd genresAdd = new genresAdd();

genresAdd.Owner = this;

genresAdd.ShowDialog();

}

private void dataGridGenreEdit\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

genresEdit genresEdit = new genresEdit((data.Genre)dataGrid.SelectedItem);

genresEdit.Owner = this;

genresEdit.ShowDialog();

}

}

private void dataGridGenreDelete\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

data.Genre genre = (data.Genre)dataGrid.SelectedItem;

database db = new database();

if (db.openConnection(db.connectionString)) {

db.genreDelete(genre.Genre\_Id);

db.loadGenres();

db.loadBooks();

dataGridSetItemSource();

MainWindow mainWindow = (MainWindow)this.Owner;

mainWindow.dataGridSetItemSource(data.books);

}

}

}

}

}

В модуле MainWindow.xaml содержится декларативное описание интерфейса главного окна на языке XAML:

<Window x:Class="bookSystem.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Информационная система книжного магазина" Height="800" Width="1200" MinHeight="800" MinWidth="1200">

<Grid>

<DockPanel>

<Menu Height="20" DockPanel.Dock="Top">

<MenuItem Header="Файл">

<MenuItem Header="Авторизация" x:Name="authorizationMenuItem" Click="authorizationMenuItem\_Click"/>

<MenuItem Header="Регистрация" x:Name="registerMenuItem" Click="registerMenuItem\_Click"/>

<Separator/>

<MenuItem Header="Выход" x:Name="exitMenuItem" Click="exitMenuItem\_Click"/>

</MenuItem>

<MenuItem Header="Данные" x:Name="dataMenuItem" Visibility="Collapsed">

<MenuItem Header="Авторы" x:Name="authorsViewMenuItem" Click="authorsViewMenuItem\_Click"/>

<MenuItem Header="Жанры" x:Name="genresView" Click="genresView\_Click"/>

<MenuItem Header="Издатели" x:Name="publishersView" Click="publishersView\_Click"/>

<MenuItem Header="Серии" x:Name="seriesView" Click="seriesView\_Click"/>

<MenuItem Header="Пользователи" x:Name="usersView" Click="usersView\_Click"/>

</MenuItem>

<MenuItem Header="Пользователь" Visibility="Collapsed" x:Name="userMenuItem">

<MenuItem Header="Профиль" x:Name="profileMenuItem" Click="profileMenuItem\_Click"/>

<MenuItem Header="Корзина" x:Name="cartMenuItem" Click="cartMenuItem\_Click"/>

</MenuItem>

</Menu>

<StackPanel DockPanel.Dock="Left" Margin="5,5,5,5" x:Name="searchPanel" Visibility="Collapsed" Width="200">

<StackPanel>

<TextBlock Text="Поиск" TextAlignment="Center"/>

<TextBlock Text="Название"/>

<TextBox x:Name="searchBookName"/>

<TextBlock Text="Автор" Margin="0,5,0,0"/>

<ComboBox x:Name="searchBookAuthor" Margin="0,5,0,0" DisplayMemberPath="Author\_Name"/>

<TextBlock Text="Жанр" Margin="0,5,0,0"/>

<ComboBox x:Name="searchBookGenre" Margin="0,5,0,0" DisplayMemberPath="Genre\_Name"/>

</StackPanel>

</StackPanel>

<DataGrid x:Name="dataGrid" IsReadOnly="True" DockPanel.Dock="Right" AutoGenerateColumns="False" FontSize="15" SelectionMode="Single" RowBackground="LightGray" AlternatingRowBackground="White" MouseDoubleClick="dataGrid\_MouseDoubleClick" ContextMenuOpening="dataGrid\_ContextMenuOpening">

<DataGrid.ContextMenu>

<ContextMenu x:Name="dataGridContextMenu" Visibility="Collapsed">

<MenuItem Header="Добавить" x:Name="dataGridBookAdd" Click="dataGridBookAdd\_Click"/>

<Separator/>

<MenuItem Header="Редактировать" x:Name="dataGridBookEdit" Click="dataGridBookEdit\_Click"/>

<MenuItem Header="Удалить" x:Name="dataGridBookDelete" Click="dataGridBookDelete\_Click" />

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Binding="{Binding Path=Book\_Name}" Width="\*"/>

<DataGridTextColumn Header="Автор" Binding="{Binding Path=Author.Author\_Name}" Width="\*"/>

<DataGridTextColumn Header="Жанр" Binding="{Binding Path=Genre.Genre\_Name}" Width="\*"/>

<DataGridTextColumn Header="Издатель" Binding="{Binding Path=Publisher.Publisher\_Name}" Width="\*"/>

<DataGridTextColumn Header="Серия" Binding="{Binding Path=Series.Series\_Name}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

</DockPanel>

</Grid>

</Window>

В модуле MainWindow.xaml.cs содержится описание работы главного окна на языке C#:

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

namespace bookSystem {

public partial class MainWindow : Window {

database db = new database();

public MainWindow() {

InitializeComponent();

if (db.openConnection(db.connectionString)) {

db.loadAuthors();

db.loadGenres();

db.loadPublishers();

db.loadSeries();

db.loadBooks();

db.loadRoles();

db.loadUsers();

db.loadUserBooks();

db.loadCart();

dataGridSetItemSource(data.books);

db.closeConnection();

}

else {

MessageBox.Show("Невозможно подключиться к базе данных!");

this.Close();

}

}

private void authorizationMenuItem\_Click(object sender, RoutedEventArgs e) {

authorization authorization = new authorization();

authorization.Owner = this;

authorization.ShowDialog();

}

private void dataGrid\_MouseDoubleClick(object sender, System.Windows.Input.MouseButtonEventArgs e) {

if (dataGrid.SelectedItem != null) {

bookView bookView = new bookView((data.Book)dataGrid.SelectedItem);

bookView.Owner = this;

bookView.ShowDialog();

}

}

public void enableUserActivity() {

registerMenuItem.IsEnabled = false;

authorizationMenuItem.IsEnabled = false;

userMenuItem.Visibility = Visibility.Visible;

if (data.currentUser.Role.Role\_Is\_Admin) {

dataGridContextMenu.Visibility = Visibility.Visible;

dataMenuItem.Visibility = Visibility.Visible;

}

}

private void dataGridBookAdd\_Click(object sender, RoutedEventArgs e) {

bookAdd bookAdd = new bookAdd();

bookAdd.Owner = this;

bookAdd.ShowDialog();

}

public void dataGridSetItemSource(List<data.Book> book) {

dataGrid.ClearValue(ItemsControl.ItemsSourceProperty);

dataGrid.ItemsSource = book;

}

private void dataGridBookEdit\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

bookEdit bookEdit = new bookEdit((data.Book)dataGrid.SelectedItem);

bookEdit.Owner = this;

bookEdit.ShowDialog();

}

}

private void dataGrid\_ContextMenuOpening(object sender, ContextMenuEventArgs e) {

if (dataGrid.SelectedItem != null) {

dataGridBookEdit.IsEnabled = true;

dataGridBookDelete.IsEnabled = true;

}

else {

dataGridBookEdit.IsEnabled = false;

dataGridBookDelete.IsEnabled = false;

}

}

private void dataGridBookDelete\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

data.Book book = (data.Book)dataGrid.SelectedItem;

db.bookDelete(book.Book\_Id);

db.loadBooks();

dataGridSetItemSource(data.books);

}

}

private void authorsViewMenuItem\_Click(object sender, RoutedEventArgs e) {

authorsView authorsView = new authorsView();

authorsView.Owner = this;

authorsView.ShowDialog();

}

private void genresView\_Click(object sender, RoutedEventArgs e) {

genresView genresView = new genresView();

genresView.Owner = this;

genresView.ShowDialog();

}

private void publishersView\_Click(object sender, RoutedEventArgs e) {

publishersView publishersView = new publishersView();

publishersView.Owner = this;

publishersView.ShowDialog();

}

private void seriesView\_Click(object sender, RoutedEventArgs e) {

seriesView seriesView = new seriesView();

seriesView.Owner = this;

seriesView.ShowDialog();

}

private void registerMenuItem\_Click(object sender, RoutedEventArgs e) {

userRegister userRegister = new userRegister();

userRegister.Owner = this;

userRegister.ShowDialog();

}

private void usersView\_Click(object sender, RoutedEventArgs e) {

usersView usersView = new usersView();

usersView.Owner = this;

usersView.ShowDialog();

}

private void exitMenuItem\_Click(object sender, RoutedEventArgs e) {

this.Close();

}

private void profileMenuItem\_Click(object sender, RoutedEventArgs e) {

userProfile userProfile = new userProfile();

userProfile.Owner = this;

userProfile.ShowDialog();

}

private void cartMenuItem\_Click(object sender, RoutedEventArgs e) {

cartView cartView = new cartView();

cartView.Owner = this;

cartView.ShowDialog();

}

}

}

В модуле publisherAdd.xaml содержится декларативное описание интерфейса окна добавления издателя на языке XAML:

<Window x:Class="bookSystem.publisherAdd"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Добавление издателя" Height="200" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<StackPanel VerticalAlignment="Center" Width="270">

<Label Content="Наименование" FontSize="18" HorizontalContentAlignment="Center"/>

<TextBox x:Name="publisherName" FontSize="18" Height="30" Margin="0,5,0,0"/>

<Button Content="Добавить" FontSize="18" Height="30" Margin="0,10,0,0" x:Name="buttonAdd" Click="buttonAdd\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле publisherAdd.xaml.cs содержится описание работы окна добавления издателя на языке C#:

using System.Windows;

namespace bookSystem {

public partial class publisherAdd : Window {

public publisherAdd() {

InitializeComponent();

}

private void buttonAdd\_Click(object sender, RoutedEventArgs e) {

if (publisherName.Text != null) {

data.Publisher publisher = data.publishers.Find(a => a.Publisher\_Name == publisherName.Text);

if (publisher.Publisher\_Name != null) {

MessageBox.Show("Издатель уже существует в базе данных!");

}

else {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.publisherAdd(publisherName.Text);

db.loadPublishers();

db.closeConnection();

publishersView publishersView = (publishersView)this.Owner;

publishersView.dataGridSetItemSource();

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

}

else {

MessageBox.Show("Необходимо указать издателя!");

}

}

}

}

В модуле publisherEdit.xaml содержится декларативное описание интерфейса окна редактирования издателя на языке XAML:

<Window x:Class="bookSystem.publisherEdit"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Редактирование издателя" Height="200" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<StackPanel VerticalAlignment="Center" Width="270">

<Label Content="Наименование" FontSize="18" HorizontalContentAlignment="Center"/>

<TextBox x:Name="publisherName" FontSize="18" Height="30" Margin="0,5,0,0"/>

<Button Content="Сохранить" FontSize="18" Height="30" Margin="0,10,0,0" x:Name="buttonAdd" Click="buttonAdd\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле publisherEdit.xaml.cs содержится описание работы окна редактирования издателя на языке C#:

using System.Windows;

namespace bookSystem {

public partial class publisherEdit : Window {

data.Publisher \_\_publisher;

public publisherEdit(data.Publisher publisher) {

InitializeComponent();

this.\_\_publisher = publisher;

publisherName.Text = \_\_publisher.Publisher\_Name;

}

private void buttonAdd\_Click(object sender, RoutedEventArgs e) {

if (publisherName.Text != null) {

data.Publisher publisher = data.publishers.Find(a => a.Publisher\_Name == publisherName.Text);

if (publisher.Publisher\_Name != null) {

MessageBox.Show("Издатель уже существует в базе данных!");

}

else {

\_\_publisher.Publisher\_Name = publisherName.Text;

database db = new database();

if (db.openConnection(db.connectionString)) {

db.publisherEdit(\_\_publisher);

db.loadPublishers();

db.loadBooks();

db.closeConnection();

publishersView publishersView = (publishersView)this.Owner;

publishersView.dataGridSetItemSource();

MainWindow mainWindow = (MainWindow)publishersView.Owner;

mainWindow.dataGridSetItemSource(data.books);

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

}

else {

MessageBox.Show("Необходимо указать издателя!");

}

}

}

}

В модуле publishersView.xaml содержится декларативное описание интерфейса окна просмотра издателей на языке XAML:

<Window x:Class="bookSystem.publishersView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Издатели" Height="400" Width="450" MinHeight="400" MinWidth="350" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<DockPanel>

<DataGrid x:Name="dataGrid" IsReadOnly="True" AutoGenerateColumns="False" FontSize="15" SelectionMode="Single" RowBackground="LightGray" AlternatingRowBackground="White">

<DataGrid.ContextMenu>

<ContextMenu x:Name="dataGridContextMenu" Visibility="Visible">

<MenuItem Header="Добавить" x:Name="dataGridPublisherAdd" Click="dataGridPublisherAdd\_Click"/>

<Separator/>

<MenuItem Header="Редактировать" x:Name="dataGridPublisherEdit" Click="dataGridPublisherEdit\_Click"/>

<MenuItem Header="Удалить" x:Name="dataGridPublisherDelete" Click="dataGridPublisherDelete\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Binding="{Binding Path=Publisher\_Name}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле publishersView.xaml.cs содержится описание работы окна просмотра издателей на языке C#:

using System.Windows;

using System.Windows.Controls;

namespace bookSystem {

public partial class publishersView : Window {

public publishersView() {

InitializeComponent();

dataGrid.ItemsSource = data.publishers;

}

public void dataGridSetItemSource() {

dataGrid.ClearValue(ItemsControl.ItemsSourceProperty);

dataGrid.ItemsSource = data.publishers;

}

private void dataGridPublisherAdd\_Click(object sender, RoutedEventArgs e) {

publisherAdd publisherAdd = new publisherAdd();

publisherAdd.Owner = this;

publisherAdd.ShowDialog();

}

private void dataGridPublisherEdit\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

publisherEdit publisherEdit = new publisherEdit((data.Publisher)dataGrid.SelectedItem);

publisherEdit.Owner = this;

publisherEdit.ShowDialog();

}

}

private void dataGridPublisherDelete\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

data.Publisher publisher = (data.Publisher)dataGrid.SelectedItem;

database db = new database();

if (db.openConnection(db.connectionString)) {

db.publisherDelete(publisher.Publisher\_Id);

db.loadPublishers();

db.loadBooks();

dataGridSetItemSource();

MainWindow mainWindow = (MainWindow)this.Owner;

mainWindow.dataGridSetItemSource(data.books);

}

}

}

}

}

В модуле seriesAdd.xaml содержится декларативное описание интерфейса окна добавления серии на языке XAML:

<Window x:Class="bookSystem.seriesAdd"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Добавление серии" Height="200" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<StackPanel VerticalAlignment="Center" Width="270">

<Label Content="Наименование" FontSize="18" HorizontalContentAlignment="Center"/>

<TextBox x:Name="seriesName" FontSize="18" Height="30" Margin="0,5,0,0"/>

<Button Content="Добавить" FontSize="18" Height="30" Margin="0,10,0,0" x:Name="buttonAdd" Click="buttonAdd\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле seriesAdd.xaml.cs содержится описание работы окна добавления серии на языке C#:

using System.Windows;

namespace bookSystem {

public partial class seriesAdd : Window {

public seriesAdd() {

InitializeComponent();

}

private void buttonAdd\_Click(object sender, RoutedEventArgs e) {

if (seriesName.Text != null) {

data.Series series = data.series.Find(a => a.Series\_Name == seriesName.Text);

if (series.Series\_Name != null) {

MessageBox.Show("Серия уже существует в базе данных!");

}

else {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.seriesAdd(seriesName.Text);

db.loadSeries();

db.closeConnection();

seriesView seriesView = (seriesView)this.Owner;

seriesView.dataGridSetItemSource();

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

}

else {

MessageBox.Show("Необходимо указать серию!");

}

}

}

}

В модуле seriesEdit.xaml содержится декларативное описание интерфейса окна редактирования серии на языке XAML:

<Window x:Class="bookSystem.seriesEdit"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Редактирование серии" Height="200" Width="400" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<StackPanel VerticalAlignment="Center" Width="270">

<Label Content="Наименование" FontSize="18" HorizontalContentAlignment="Center"/>

<TextBox x:Name="seriesName" FontSize="18" Height="30" Margin="0,5,0,0"/>

<Button Content="Сохранить" FontSize="18" Height="30" Margin="0,10,0,0" x:Name="buttonSave" Click="buttonSave\_Click"/>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле seriesEdit.xaml.cs содержится описание работы окна редактирования серии на языке C#:

using System.Windows;

namespace bookSystem {

public partial class seriesEdit : Window {

data.Series \_\_series;

public seriesEdit(data.Series series) {

InitializeComponent();

this.\_\_series = series;

seriesName.Text = \_\_series.Series\_Name;

}

private void buttonSave\_Click(object sender, RoutedEventArgs e) {

if (seriesName.Text != null) {

data.Series series = data.series.Find(a => a.Series\_Name == seriesName.Text);

if (series.Series\_Name != null) {

MessageBox.Show("Издатель уже существует в базе данных!");

}

else {

\_\_series.Series\_Name = seriesName.Text;

database db = new database();

if (db.openConnection(db.connectionString)) {

db.seriesEdit(\_\_series);

db.loadSeries();

db.loadBooks();

db.closeConnection();

seriesView seriesView = (seriesView)this.Owner;

seriesView.dataGridSetItemSource();

MainWindow mainWindow = (MainWindow)seriesView.Owner;

mainWindow.dataGridSetItemSource(data.books);

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

}

else {

MessageBox.Show("Необходимо указать издателя!");

}

}

}

}

В модуле seriesView.xaml содержится декларативное описание интерфейса окна просмотра серий на языке XAML:

<Window x:Class="bookSystem.seriesView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Серии" Height="400" Width="450" MinHeight="400" MinWidth="350" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<DockPanel>

<DataGrid x:Name="dataGrid" IsReadOnly="True" AutoGenerateColumns="False" FontSize="15" SelectionMode="Single" RowBackground="LightGray" AlternatingRowBackground="White">

<DataGrid.ContextMenu>

<ContextMenu x:Name="dataGridContextMenu" Visibility="Visible">

<MenuItem Header="Добавить" x:Name="dataGridSeriesAdd" Click="dataGridSeriesAdd\_Click" />

<Separator/>

<MenuItem Header="Редактировать" x:Name="dataGridSeriesEdit" Click="dataGridSeriesEdit\_Click" />

<MenuItem Header="Удалить" x:Name="dataGridSeriesDelete" Click="dataGridSeriesDelete\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Binding="{Binding Path=Series\_Name}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле seriesView.xaml.cs содержится описание работы окна просмотра серий на языке C#:

using System.Windows;

using System.Windows.Controls;

namespace bookSystem {

public partial class seriesView : Window {

public seriesView() {

InitializeComponent();

dataGrid.ItemsSource = data.series;

}

public void dataGridSetItemSource() {

dataGrid.ClearValue(ItemsControl.ItemsSourceProperty);

dataGrid.ItemsSource = data.series;

}

private void dataGridSeriesAdd\_Click(object sender, RoutedEventArgs e) {

seriesAdd seriesAdd = new seriesAdd();

seriesAdd.Owner = this;

seriesAdd.ShowDialog();

}

private void dataGridSeriesEdit\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

seriesEdit seriesEdit = new seriesEdit((data.Series)dataGrid.SelectedItem);

seriesEdit.Owner = this;

seriesEdit.ShowDialog();

}

}

private void dataGridSeriesDelete\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

data.Series series = (data.Series)dataGrid.SelectedItem;

database db = new database();

if (db.openConnection(db.connectionString)) {

db.seriesDelete(series.Series\_Id);

db.loadSeries();

db.loadBooks();

dataGridSetItemSource();

MainWindow mainWindow = (MainWindow)this.Owner;

mainWindow.dataGridSetItemSource(data.books);

}

}

}

}

}

В модуле userAdd.xaml содержится декларативное описание интерфейса окна добавления пользователя на языке XAML:

<Window x:Class="bookSystem.userAdd"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Добавление пользователя" Height="280" Width="370" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderBrush="Black" BorderThickness="1,1,1,1" CornerRadius="1,1,1,1">

<DockPanel>

<DockPanel DockPanel.Dock="Bottom">

<StackPanel HorizontalAlignment="Center" Margin="0,0,0,5">

<Button Content="Принять" Width="100" FontSize="15" Click="Button\_Click"/>

</StackPanel>

</DockPanel>

<StackPanel Width="200" Margin="0,5,0,5">

<TextBlock Text="Логин" FontSize="16"/>

<TextBox x:Name="addLogin" Height="25" FontSize="16" Width="200"/>

<TextBlock Text="Пароль" FontSize="16"/>

<TextBox x:Name="addPassword" Height="25" FontSize="16" Width="200"/>

<TextBlock Text="Повторите пароль" FontSize="16"/>

<TextBox x:Name="addPasswordRepeat" Height="25" FontSize="16" Width="200"/>

<TextBlock Text="Роль" FontSize="16"/>

<ComboBox x:Name="addRole" Height="25" FontSize="16" DisplayMemberPath="Role\_Name"/>

</StackPanel>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле userAdd.xaml.cs содержится описание работы окна добавления пользователя на языке C#:

using System.Windows;

namespace bookSystem {

public partial class userAdd : Window {

public userAdd() {

InitializeComponent();

addRole.ItemsSource = data.roles;

addRole.SelectedIndex = 0;

}

private void Button\_Click(object sender, RoutedEventArgs e) {

if (addLogin.Text != string.Empty && addPassword.Text != string.Empty && addPasswordRepeat.Text != string.Empty) {

bool exists = false;

foreach (data.Users user in data.users) {

if (addLogin.Text == user.User\_Login) {

exists = true;

break;

}

}

if (exists) {

MessageBox.Show("Пользователь с таким логином уже существует!");

}

else {

if (addPassword.Text == addPasswordRepeat.Text) {

data.Users user = new data.Users {

User\_Login = addLogin.Text

, User\_Password = addPassword.Text

, Role = data.roles.Find(r => r.Role\_Name.Contains(addRole.Text))

};

database db = new database();

if (db.openConnection(db.connectionString)) {

db.userAdd(user);

db.loadUsers();

db.closeConnection();

usersView usersView = (usersView)this.Owner;

usersView.dataGridSetItemSource();

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

else {

MessageBox.Show("Введённые пароли не совпадают!");

}

}

}

else {

MessageBox.Show("Поля \"Логин\", \"Пароль\", \"Повторите пароль\" не должны быть пустыми!");

}

}

}

}

В модуле userEdit.xaml содержится декларативное описание интерфейса окна редактирования пользователя на языке XAML:

<Window x:Class="bookSystem.userEdit"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Редактирование пользователя" Height="280" Width="370" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderBrush="Black" BorderThickness="1,1,1,1" CornerRadius="1,1,1,1">

<DockPanel>

<DockPanel DockPanel.Dock="Bottom">

<StackPanel HorizontalAlignment="Center" Margin="0,0,0,5">

<Button Content="Принять" Width="100" FontSize="15" Click="Button\_Click"/>

</StackPanel>

</DockPanel>

<StackPanel Width="200" Margin="0,5,0,5">

<TextBlock Text="Логин" FontSize="16"/>

<TextBox IsEnabled="False" x:Name="editLogin" Height="25" FontSize="16" Width="200"/>

<TextBlock Text="Пароль" FontSize="16"/>

<TextBox x:Name="editPassword" Height="25" FontSize="16" Width="200"/>

<TextBlock Text="Повторите пароль" FontSize="16"/>

<TextBox x:Name="editPasswordRepeat" Height="25" FontSize="16" Width="200"/>

<TextBlock Text="Роль" FontSize="16"/>

<ComboBox x:Name="editRole" Height="25" FontSize="16" DisplayMemberPath="Role\_Name"/>

</StackPanel>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле userEdit.xaml.cs содержится описание работы окна редактирования пользователя на языке C#:

using System.Windows;

namespace bookSystem {

public partial class userEdit : Window {

data.Users \_\_user;

public userEdit(data.Users user) {

InitializeComponent();

editRole.ItemsSource = data.roles;

this.\_\_user = user;

editLogin.Text = \_\_user.User\_Login;

editRole.Text = \_\_user.Role.Role\_Name;

}

private void Button\_Click(object sender, RoutedEventArgs e) {

if (editLogin.Text != string.Empty && editPassword.Text != string.Empty && editPasswordRepeat.Text != string.Empty) {

data.Users user = data.users.Find(u => u.User\_Login == editLogin.Text);

if (editPassword.Text == editPasswordRepeat.Text) {

\_\_user.User\_Password = editPassword.Text;

\_\_user.Role = data.roles.Find(r => r.Role\_Name.Contains(editRole.Text!));

database db = new database();

if (db.openConnection(db.connectionString)) {

db.userEdit(\_\_user);

db.loadUsers();

db.closeConnection();

usersView usersView = (usersView)this.Owner;

usersView.dataGridSetItemSource();

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

else {

MessageBox.Show("Введённые пароли не совпадают!");

}

}

else {

MessageBox.Show("Поля \"Пароль\", \"Повторите пароль\" не должны быть пустыми!");

}

}

}

}

В модуле userProfile.xaml содержится декларативное описание интерфейса окна просмотра профиля пользователя на языке XAML:

<Window x:Class="bookSystem.userProfile"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Информация о пользователе" Height="370" Width="340" ResizeMode="NoResize" WindowStartupLocation="CenterOwner" Activated="Window\_Activated">

<Grid>

<Border BorderThickness="1" BorderBrush="Black" Margin="5" CornerRadius="5">

<StackPanel Margin="5" Orientation="Vertical" HorizontalAlignment="Left">

<DockPanel>

<StackPanel DockPanel.Dock="Left" Margin="5" Width="150">

<TextBlock Text="ID"/>

<Separator/>

<TextBlock Text="Логин"/>

<Separator/>

<TextBlock Text="Дата регистрации"/>

<Separator/>

<TextBlock Text="Роль"/>

</StackPanel>

<StackPanel DockPanel.Dock="Right" Margin="5" Width="150">

<TextBlock x:Name="profileId" Text="1"/>

<Separator/>

<TextBlock x:Name="profileLogin" Text="admin"/>

<Separator/>

<TextBlock x:Name="profileRegisterDate" Text="2023.11.24 15:00"/>

<Separator/>

<TextBlock x:Name="profileRole" Text="Администратор"/>

</StackPanel>

</DockPanel>

<Border BorderThickness="1" BorderBrush="Black" Margin="5" CornerRadius="5">

<DataGrid x:Name="dataGrid" Height="200" Margin="5" IsReadOnly="True" AutoGenerateColumns="False" SelectionMode="Single" RowBackground="LightGray" AlternatingRowBackground="White" MouseDoubleClick="dataGrid\_MouseDoubleClick">

<DataGrid.Columns>

<DataGridTextColumn Width="\*" Header="Название" Binding="{Binding Path=Book\_Name}"/>

<DataGridTextColumn Width="\*" Header="Автор" Binding="{Binding Path=Author.Author\_Name}"/>

</DataGrid.Columns>

</DataGrid>

</Border>

</StackPanel>

</Border>

</Grid>

</Window>

В модуле userProfile.xaml.cs содержится описание работы окна просмотра профиля пользователя на языке C#:

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

namespace bookSystem {

public partial class userProfile : Window {

List<data.Book> usersBooks = new List<data.Book>();

public userProfile() {

InitializeComponent();

profileId.Text = data.currentUser.User\_Id.ToString();

profileLogin.Text = data.currentUser.User\_Login;

profileRegisterDate.Text = data.currentUser.User\_Register\_Date.ToString("dd.MM.yyyy HH:mm:ss");

profileRole.Text = data.currentUser.Role.Role\_Name;

getUserBooks();

dataGridSetItemSource(usersBooks);

}

public void dataGridSetItemSource(List<data.Book> book) {

dataGrid.ClearValue(ItemsControl.ItemsSourceProperty);

dataGrid.ItemsSource = book;

}

private void dataGrid\_MouseDoubleClick(object sender, System.Windows.Input.MouseButtonEventArgs e) {

if (dataGrid.SelectedItem != null) {

bookView bookView = new bookView((data.Book)dataGrid.SelectedItem);

bookView.Owner = this;

bookView.ShowDialog();

}

}

private void getUserBooks() {

List<data.UsersBooks> \_\_usersBooks = data.usersBooks.FindAll(ub => ub.UB\_User\_Id == data.currentUser.User\_Id);

usersBooks.Clear();

foreach (data.UsersBooks ub in \_\_usersBooks) {

usersBooks.Add(data.books.Find(b => b.Book\_Id == ub.UB\_Book\_Id));

}

}

private void Window\_Activated(object sender, System.EventArgs e) {

getUserBooks();

dataGridSetItemSource(usersBooks);

}

}

}

В модуле userRegister.xaml содержится декларативное описание интерфейса окна регистрации пользователя на языке XAML:

<Window x:Class="bookSystem.userRegister"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Регистрация" Height="240" Width="370" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderBrush="Black" BorderThickness="1,1,1,1" CornerRadius="1,1,1,1">

<DockPanel>

<DockPanel DockPanel.Dock="Bottom">

<StackPanel HorizontalAlignment="Center" Margin="0,0,0,5">

<Button Content="Принять" Width="100" FontSize="15" Click="Button\_Click"/>

</StackPanel>

</DockPanel>

<StackPanel Width="200" Margin="0,5,0,5">

<TextBlock Text="Логин" FontSize="16"/>

<TextBox x:Name="registerLogin" Height="25" FontSize="16" Width="200"/>

<TextBlock Text="Пароль" FontSize="16"/>

<TextBox x:Name="registerPassword" Height="25" FontSize="16" Width="200"/>

<TextBlock Text="Повторите пароль" FontSize="16"/>

<TextBox x:Name="registerPasswordRepeat" Height="25" FontSize="16" Width="200"/>

</StackPanel>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле userRegister.xaml.cs содержится описание работы окна регистрации пользователя на языке C#:

using System.Windows;

namespace bookSystem {

public partial class userRegister : Window {

public userRegister() {

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e) {

if (registerLogin.Text != string.Empty && registerPassword.Text != string.Empty && registerPasswordRepeat.Text != string.Empty) {

bool exists = false;

foreach (data.Users user in data.users) {

if (registerLogin.Text == user.User\_Login) {

exists = true;

break;

}

}

if (exists) {

MessageBox.Show("Пользователь с таким логином уже существует!");

}

else {

if (registerPassword.Text == registerPasswordRepeat.Text) {

data.Users user = new data.Users {

User\_Login = registerLogin.Text

, User\_Password = registerPassword.Text

, Role = data.roles.Find(r => r.Role\_Name.Contains("Пользователь") && r.Role\_Is\_Admin == false)

};

database db = new database();

if (db.openConnection(db.connectionString)) {

db.userAdd(user);

db.loadUsers();

db.closeConnection();

this.Close();

}

else {

MessageBox.Show("Подключение к базе данных неактивно!");

}

}

else {

MessageBox.Show("Введённые пароли не совпадают!");

}

}

}

else {

MessageBox.Show("Поля \"Логин\", \"Пароль\", \"Повторите пароль\" не должны быть пустыми!");

}

}

}

}

В модуле usersView.xaml содержится декларативное описание интерфейса окна добавления книги на языке XAML:

<Window x:Class="bookSystem.usersView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Пользователи" Height="400" Width="450" MinHeight="400" MinWidth="350" ResizeMode="NoResize" WindowStartupLocation="CenterOwner">

<Grid>

<Border Margin="5,5,5,5" BorderThickness="1,1,1,1" BorderBrush="Gray" CornerRadius="2,2,2,2">

<DockPanel>

<DataGrid x:Name="dataGrid" IsReadOnly="True" AutoGenerateColumns="False" FontSize="15" SelectionMode="Single" RowBackground="LightGray" AlternatingRowBackground="White">

<DataGrid.ContextMenu>

<ContextMenu x:Name="dataGridContextMenu" Visibility="Visible">

<MenuItem Header="Добавить" x:Name="dataGridUserAdd" Click="dataGridUserAdd\_Click"/>

<Separator/>

<MenuItem Header="Редактировать" x:Name="dataGridUserEdit" Click="dataGridUserEdit\_Click"/>

<MenuItem Header="Удалить" x:Name="dataGridUserDelete" Click="dataGridUserDelete\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Логин" Binding="{Binding Path=User\_Login}" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле usersView.xaml.cs содержится описание работы окна добавления книги на языке C#:

using System.Windows;

using System.Windows.Controls;

namespace bookSystem {

public partial class usersView : Window {

public usersView() {

InitializeComponent();

dataGrid.ItemsSource = data.users;

}

public void dataGridSetItemSource() {

dataGrid.ClearValue(ItemsControl.ItemsSourceProperty);

dataGrid.ItemsSource = data.users;

}

private void dataGridUserAdd\_Click(object sender, RoutedEventArgs e) {

userAdd userAdd = new userAdd();

userAdd.Owner = this;

userAdd.ShowDialog();

}

private void dataGridUserEdit\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

userEdit userEdit = new userEdit((data.Users)dataGrid.SelectedItem);

userEdit.Owner = this;

userEdit.ShowDialog();

}

}

private void dataGridUserDelete\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.userDelete(((data.Users)dataGrid.SelectedItem).User\_Id);

dataGridSetItemSource();

db.closeConnection();

}

}

}

}

}

В модуле cartView.xaml содержится декларативное описание интерфейса окна просмотра корзины на языке XAML:

<Window x:Class="bookSystem.cartView"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:bookSystem"

mc:Ignorable="d"

Title="Корзина" Height="450" Width="1000" Activated="Window\_Activated">

<Window.Resources>

<Style TargetType="Border">

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="BorderBrush" Value="Black"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="CornerRadius" Value="5"/>

</Style>

<Style TargetType="Button">

<Setter Property="Width" Value="150"/>

<Setter Property="Margin" Value="5"/>

</Style>

</Window.Resources>

<Grid>

<Border>

<DockPanel>

<Border DockPanel.Dock="Left">

<StackPanel Width="200" VerticalAlignment="Center">

<Border Width="150">

<StackPanel Margin="0,5,0,5">

<TextBlock x:Name="totalBooksTextBlock" Text="Кол-во товаров: 999999" HorizontalAlignment="Center"/>

<TextBlock x:Name="totalPriceTextBlock" Text="Сумма чека: 1000000" HorizontalAlignment="Center"/>

</StackPanel>

</Border>

<Button>

<CheckBox Content="Печать чека" x:Name="printChequeCheckBox" IsChecked="True"/>

</Button>

<Button Content="Оформить заказ" x:Name="placeOrderButton" Click="placeOrderButton\_Click"/>

<Button Content="Очистить корзину" x:Name="cleanCartButton" Click="cleanCartButton\_Click"/>

</StackPanel>

</Border>

<Border DockPanel.Dock="Right">

<DataGrid x:Name="dataGrid" IsReadOnly="True" AutoGenerateColumns="False" FontSize="15" SelectionMode="Single" RowBackground="LightGray" AlternatingRowBackground="White" MouseDoubleClick="dataGrid\_MouseDoubleClick" RowHeight="40">

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Binding="{Binding Path=Book.Book\_Name}" Width="\*">

<DataGridTextColumn.ElementStyle>

<Style TargetType="TextBlock">

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="TextWrapping" Value="WrapWithOverflow"/>

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Цена" Binding="{Binding Path=Book.Book\_Price}" Width="60">

<DataGridTextColumn.ElementStyle>

<Style TargetType="TextBlock">

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Кол-во" Width="60" Binding="{Binding Path=Cart\_Values}">

<DataGridTextColumn.ElementStyle>

<Style TargetType="TextBlock">

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTemplateColumn Width="35">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="plusValueButton" Content="+" Width="20" Tag="{Binding SelectedItem, ElementName=dataGrid}" Click="plusValueButton\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="35">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="minusValueButton" Content="-" Width="20" Tag="{Binding SelectedItem, ElementName=dataGrid}" Click="minusValueButton\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Border>

</DockPanel>

</Border>

</Grid>

</Window>

В модуле cartView.xaml содержится описание работы окна просмотра корзины на языке C#:

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

using System;

using System.Windows.Documents;

namespace bookSystem {

public partial class cartView : Window {

List<data.Carts> cart = new List<data.Carts>();

string cheque;

public cartView() {

InitializeComponent();

refreshLists();

refreshTextBlocks();

}

private void getUserBooks() {

cart.Clear();

cart = data.carts.FindAll(c => c.User\_Id == data.currentUser.User\_Id);

}

public void dataGridSetItemSource(List<data.Carts> cart) {

dataGrid.ClearValue(ItemsControl.ItemsSourceProperty);

dataGrid.ItemsSource = cart;

}

private void dataGrid\_MouseDoubleClick(object sender, System.Windows.Input.MouseButtonEventArgs e) {

if (dataGrid.SelectedItem != null) {

data.Carts carts = (data.Carts)dataGrid.SelectedItem;

bookView bookView = new bookView(carts.Book);

bookView.Owner = this;

bookView.ShowDialog();

}

}

private void Window\_Activated(object sender, System.EventArgs e) {

refreshLists();

refreshTextBlocks();

}

private void placeOrderButton\_Click(object sender, RoutedEventArgs e) {

if (cart.Count > 0) {

cheque = generateCheque();

if (printChequeCheckBox.IsChecked == true) {

printCheque();

}

database db = new database();

if (db.openConnection(db.connectionString)) {

foreach (data.Carts c in cart) {

db.userBookAdd(data.currentUser.User\_Id, c.Book\_Id);

}

db.loadUserBooks();

db.closeConnection();

clearCart();

}

}

}

private void cleanCartButton\_Click(object sender, RoutedEventArgs e) {

if (cart.Count > 0) {

clearCart();

}

}

private void refreshLists() {

getUserBooks();

dataGridSetItemSource(cart);

}

private string generateCheque() {

int \_\_count = 1;

int \_\_totalBooks = 0;

int \_\_totalPrice = 0;

string \_\_cheque = $"Чек: {Guid.NewGuid()}\n" +

$"Дата: {DateTime.Now.ToString("G")}\n" +

$"Покупатель: {data.currentUser.User\_Login}\n" +

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" +

$"Товар Количество Цена\n";

foreach (var item in cart) {

\_\_cheque += $"{\_\_count}. {item.Book.Book\_Name} - {item.Cart\_Values} шт. - {item.Book.Book\_Price} руб.\n";

\_\_totalBooks += item.Cart\_Values;

\_\_totalPrice += item.Book.Book\_Price \* item.Cart\_Values;

\_\_count++;

}

\_\_cheque += $"Количество товаров: {\_\_totalBooks} шт.\n";

\_\_cheque += $"Общая сумма: {\_\_totalPrice} руб.\n";

\_\_cheque += "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

return \_\_cheque;

}

private void printCheque() {

PrintDialog printDialog = new PrintDialog();

if (printDialog.ShowDialog() == true) {

FlowDocument flowDocument = new FlowDocument();

foreach (string line in cheque.Split('\n')) {

Paragraph paragraph = new Paragraph();

paragraph.Margin = new Thickness(0);

paragraph.Inlines.Add(new Run(line));

paragraph.FontFamily = new System.Windows.Media.FontFamily("Times New Roman");

flowDocument.Blocks.Add(paragraph);

}

DocumentPaginator paginator = ((IDocumentPaginatorSource)flowDocument).DocumentPaginator;

printDialog.PrintDocument(paginator, this.Title);

}

}

private void clearCart() {

database db = new database();

if (db.openConnection(db.connectionString)) {

foreach (data.Carts c in cart) {

db.cartRemove(c);

}

db.loadCart();

refreshLists();

refreshTextBlocks();

db.closeConnection();

}

}

private void plusValueButton\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.cartAddValue((data.Carts)dataGrid.SelectedItem);

db.loadCart();

refreshLists();

refreshTextBlocks();

db.closeConnection();

}

}

}

private void minusValueButton\_Click(object sender, RoutedEventArgs e) {

if (dataGrid.SelectedItem != null) {

database db = new database();

if (db.openConnection(db.connectionString)) {

db.cartRemoveValue((data.Carts)dataGrid.SelectedItem);

db.loadCart();

refreshLists();

refreshTextBlocks();

db.closeConnection();

}

}

}

private void refreshTextBlocks() {

int totalBooks = 0;

int totalPrice = 0;

foreach (data.Carts c in cart) {

totalBooks += c.Cart\_Values;

totalPrice += c.Book.Book\_Price \* c.Cart\_Values;

}

totalBooksTextBlock.Text = $"Кол-во товаров: {totalBooks}";

totalPriceTextBlock.Text = $"Сумма чека: {totalPrice}";

}

}

}