****

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

КОЛЛЕДЖ МНОГОУРОВНЕВОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ОТЧЕТ

По учебной практике УП.01.01 Разработка программных модулей

программного обеспечения для компьютерных систем

специальность 09.02.03 Программирование в компьютерных системах

Выполнил студент:

Капитонов Р.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Гусятинер Л. Б.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

Москва, 2023

**Содержание**

[**Раздел 3. Разработка проекта с графическим интерфейсом** 2](#_Toc151979131)

[3.2 Разработка требований к проекту. Построение диаграммы использования. 4](#_Toc151979132)

[1.3 Разработка сценария проекта 5](#_Toc151979133)

[1.4 Построение диаграммы классов 5](#_Toc151979134)

[1.5 Разработка базы данных 6](#_Toc151979135)

[1.6 Разработка главного модуля 10](#_Toc151979136)

[1.7 Разработка входящих модулей 11](#_Toc151979137)

[1.8 Тестирование и отладка 17](#_Toc151979138)

[1.9 Дневник 18](#_Toc151979139)

# **Раздел 3. Разработка проекта с графическим интерфейсом**

В данном проекте мы работаем в основе с двумя библиотеками “MAUI” и “EntityFramework”.

Для проектов использующих MAUI необходимо скачать и установить расширение для Visual Studio.

Для использования EntityFramework нужно установить пакет NuGet.

Использование через using:

using Microsoft.EntityFrameworkCore;

1. База данных

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис. 1. Таблица “Authors”

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рис. 2. Таблица “Books”

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис. 3. Таблица “Genres”

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис. 4. Таблица “Users”

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рис. 5. Таблица “BooksAndGenres”

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рис. 6. Таблица “BooksAndUsers”

3.2 Разработка требований к проекту. Построение диаграммы использования.

Изображение выглядит как текст, диаграмма, рисунок, зарисовка

Автоматически созданное описание

Рис. 7. Диаграмма использования

* 1. Разработка сценария проекта

В данном разделе приведен сценарий использования программы пользователем (Рисунок 8).

Изображение выглядит как диаграмма, линия, текст

Автоматически созданное описание

Рис. 8. Сценарий проекта

* 1. Построение диаграммы классов

В данном разделе находятся диаграммы классов приложения.(Рисунок 9)

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рис. 9. Диаграмма классов

* 1. Разработка базы данных

В проекте в качестве базы данных используется **SQLite**. Причины использования:

**Простота отладки**: Базу данных SQLite легче редактировать и изменять. В случае проблем с миграцией её легче удалить, т.к. БД по своей сути файл.

Для работы с бд мы решили написать отдельный класс AppDbContext.

Листинг 1: [AppDbContext.cs](https://github.com/prog-kkkmt/p1-18/blob/Gymrasimov/%D0%A3%D0%9F/project_w_bd/sqlighter.py)

using Microsoft.EntityFrameworkCore;

namespace Classes;

public class AppDbContext : DbContext

{

public AppDbContext(DbContextOptions options) : base(options)

{

}

protected AppDbContext()

{

}

public DbSet<User> Users { get; set; }

public DbSet<Book> Books { get; set; }

public DbSet<Author> Authors { get; set; }

public DbSet<Genre> Genres { get; set; }

public DbSet<BooksAndGenres> BooksAndGenres { get; set; }

public DbSet<BooksAndUsers> BooksAndUsers { get; set; }

//public string databasepath = $"Data Source={@"C:\Projects\CourseWork\MyDatabase.db"}";

//"Host=127.0.0.1;Port=5432;Database=KRVBooks;Username=TestGroupLocalhost;Password=postgres;";

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

// Путь к базе данных SQLite. Файл будет создан в текущей директории приложения. (bin/Debug/)

//var databasePath = System.IO.Path.Combine(System.Environment.CurrentDirectory, "MyDatabase.db");

//optionsBuilder.UseSQlite($"Data Source={databasePath}");

//optionsBuilder.UseNpgsql(databasepath);

//optionsBuilder.UseSqlite(databasepath);

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<BooksAndGenres>().HasKey(x => new { x.BookId, x.GenreId });

modelBuilder.Entity<BooksAndGenres>()

.HasOne(bg => bg.Book)

.WithMany(b => b.Genres)

.HasForeignKey(bg => bg.BookId);

modelBuilder.Entity<BooksAndUsers>().HasKey(x => new { x.BookId, x.UserId });

modelBuilder.Entity<BooksAndUsers>()

.HasOne(bu => bu.User)

.WithMany(u => u.Books)

.HasForeignKey(bu => bu.UserId);

}

}

* 1. Разработка главного модуля

Главный модуль состоит из файла Index. Перенаправление на него выполняется сразу при запуске программы. Его также можно назвать главным меню.

Листинг 2. Главный модуль Index

@page "/"

@inject AuthService authService

@using BookDesktop.Models

@using BookDesktop.Service

<div class="w-50 m-auto">

<h3>Авторизация</h3>

<EditForm Model="@Person" OnValidSubmit="@Submit">

<DataAnnotationsValidator/>

<ValidationSummary/>

<div class="mb-3">

<label for="exampleInputEmail1" class="form-label">Логин</label>

<InputText class="form-control" @bind-Value="Person!.Login" id="exampleInputEmail1"/>

</div>

<div class="mb-3">

<label for="exampleInputPassword1" class="form-label">Пароль</label>

<InputText type="password" class="form-control" @bind-Value="Person!.Password" id="exampleInputPassword1"/>

</div>

<button type="submit" class="btn btn-primary">Войти</button>

</EditForm>

</div>

@code

{

Person Person = default!;

private void Submit()

{

if (Person.Login != "" && Person.Password != "")

authService.Login(Person.Login, Person.Password);

}

protected override void OnInitialized()

{

#if DEBUG

authService.Login("root", "root");

#else

base.OnInitialized();

Person = new();

#endif

}

}

* 1. Разработка входящих модулей

Описание спецификаций к модулям

Всего разработано 2 модуля:

* Окно каталога книг (Books)
* Окно редактирования корзины (Bin)

Описание модулей:

**Окно каталога книг** – на данном окне находится список книг, которые можно добавить в корзину по нажатию кнопки.

Рассмотрим функцию обработки добавления/удаления книги в/из корзинына и её блок-схему:

Листинг 3. Функция обработки квитанции на оплату

private void ToggleBookBin(Book book)

{

if (bin.books.Contains(book))

{

bin.books.Remove(book);

}

else

{

if (book.Amount > 0)

bin.books.Add(book);

}

InvokeAsync(StateHasChanged);

}

Изображение выглядит как диаграмма, Технический чертеж, линия, План

Автоматически созданное описание

Рисунок 10. Блок-схема к функции ToggleBookBin.

**Окно редактирования корзины** – на этом окне располагается меню для управления содержимым корзины. В этом меню есть следующие кнопки:

* Удалить из корзины
* Оформить заказ
  1. Тестирование и отладка

В ходе написания проекта при попытке запустить программу были получены ошибки (Рисунок 11):

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Автоматически созданное описание

Рис. 11. Ошибки

При проверке кода были исправлены найденные ошибки, в результате при запуске программы ошибок не было (Рисунок 12):

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рис. 12. Успешная сборка

* 1. Дневник

|  |  |  |
| --- | --- | --- |
| Дата | Содержание работ | Отметка о выполнении |
| 05.09.2023 | Выполнение тестового задания с чтением из файла. |  |
| 09.09.2023 | Модификация задачи и добавление интерфейса. |  |
| 12.09.2023 | Подготовка к тестовому заданию с файлами и работой с БД.(Канцелярия) |  |
| 15.09.2023 | Задачи со списками на Ulearn.me. |  |
| 16.09.2023 | Разработка функционала на OpenSource проекте. |  |
| 19.09.2023 | Разработка ТЗ для проекта. |  |
| 26.09.2023 | Выполнение заданий на заданном языке(Python): калькулятор и консольная игра. |  |
| 02.10.2023 | Изучение финализаторов и дженериков. |  |
| 09.10.2023 | Выполнение тестового задания и работа с БД внутри кода. |  |
| 16.10.2023 | Полноценная реализация тестовое задание с файлами и работой с БД и интерфейсом.(Канцелярия) |  |
| 23.10.2023 | Пройдены разделы 1 главы на Ulearn.me до алгоритмов.(строки, циклы, тестирование) |  |
| 30.10.2023 | Изучал асинхронность и многопоточность. |  |
| 06.11.2023 | Изучал графики в WPF. |  |
| 20.10.2023 | Изучал boxing/unboxing, структуры с отличиями от классов, абстрактные классы и интерфейсы. |  |
| 27.10.2023 | Составление отчёта по практике. |  |