Практическая работа № 3.  
ASP.Net MVC

**Цель практической работы**

Закрепление теоретических знаний по основам разработки приложений на C# с помощью технологии ASP.NET MVC.

**Постановка задачи**

ASP.NET — это платформа для разработки веб-приложений, созданная компанией Microsoft. Она позволяет разработчикам создавать динамические веб-сайты, веб-приложения и веб-сервисы. ASP.NET поддерживает языки .NET, такие как C#, и предоставляет мощные инструменты и библиотеки для упрощения разработки.

**Основные компоненты ASP.NET**

1. **ASP.NET Web Forms**: Позволяет создавать веб-приложения с использованием событийно-ориентированной модели, аналогичной Windows-приложениям. Подходит для быстрого создания приложений с минимальным количеством кода.
2. **ASP.NET MVC**: Архитектурный паттерн Model-View-Controller, который разделяет приложение на три основных компонента:
   * **Model**: Представляет данные и бизнес-логику.
   * **View**: Отвечает за отображение данных пользователю.
   * **Controller**: Обрабатывает пользовательские запросы, взаимодействует с моделью и выбирает представление для отображения.
3. **ASP.NET Web API**: Позволяет создавать RESTful сервисы, которые могут быть использованы для взаимодействия с клиентскими приложениями, такими как мобильные приложения и одностраничные приложения (SPA).
4. **ASP.NET Core**: Современная версия ASP.NET, которая является кросс-платформенной, высокопроизводительной и более легковесной. Поддерживает создание как веб-приложений, так и API.

**Задание на практическую работу**

1. **Проведите анализ предметной области:**

* Опишите сущности (классы) из вашего варианта, их свойства и связи между ними (1 к 1, 1 ко многим, многие ко многим).
* Определите, какие CRUD-операции нужны для каждой сущности.

1. **Разработайте схему** приложения (в <https://www.drawio.com/>):

* Модели бизнес логики
* Модели БД
* Контроллеры (какие у них будут действия)
* Сервисы (какая бизнес-логика будет вынесена в сервисный слой?)
* Репозитории (как будет организовано взаимодействие с БД через ORM?)

1. **Проектирование API**

* Определите, какие методы API нужны (GET/POST/PUT/DELETE).
* Опишите форматы запросов и ответов (JSON-примеры).

Пример:

*// GET /api/books/1*

{

"id": 1,

"title": "C# для профессионалов",

"year": 2023,

"authorId": 5

}

1. **Определите план тестирования**

* Какие сценарии нужно проверить в лабораторной работе?
* Пример:
  + Создание книги с несуществующим authorId → должна быть ошибка.
  + Удаление автора → каскадное удаление его книг (или запрет, если книги есть).

**Варианты заданий на практическую работу**

1. **Студент и Экзамен**
   * **Студент**: ФИО, Курс, Номер группы, Возраст
   * **Экзамен**: Предмет, Оценка, Студент
2. **Библиотека и Книга**
   * **Библиотека**: Название, Адрес, Телефон
   * **Книга**: Название, Автор, Жанр, ISBN, Библиотека
3. **Интернет-магазин и Продукт**
   * **Интернет-магазин**: Название, Адрес, Телефон
   * **Продукт**: Название, Цена, Описание, Категория, Интернет-магазин
4. **Блог и Пост**
   * **Блог**: Название, Автор, Дата создания
   * **Пост**: Заголовок, Содержание, Дата публикации, Блог
5. **Задачи и Проект**
   * **Проект**: Название, Описание, Дата начала, Дата окончания
   * **Задача**: Название, Описание, Статус, Проект
6. **Клиент и Заказ**
   * **Клиент**: ФИО, Email, Телефон
   * **Заказ**: Дата, Сумма, Статус, Клиент
7. **Сотрудник и Отдел**
   * **Сотрудник**: ФИО, Должность, Возраст
   * **Отдел**: Название, Местоположение, Сотрудники
8. **Фильм и Рецензия**
   * **Фильм**: Название, Жанр, Год выпуска
   * **Рецензия**: Оценка, Содержание, Фильм
9. **Товары и Корзина**
   * **Товар**: Название, Цена, Описание
   * **Корзина**: Пользователь, Товары
10. **Университет и Факультет**
    * **Университет**: Название, Город, Количество студентов
    * **Факультет**: Название, Декан, Университет
11. **Спортсмен и Команда**
    * **Спортсмен**: ФИО, Возраст, Спорт
    * **Команда**: Название, Тренер, Спортсмены
12. **Ресторан и Меню**
    * **Ресторан**: Название, Адрес, Телефон
    * **Меню**: Название блюда, Цена, Ресторан
13. **Курс и Урок**
    * **Курс**: Название, Описание, Дата начала
    * **Урок**: Название, Дата, Курс
14. **Пользователь и Сообщение**
    * **Пользователь**: Логин, Email, Дата регистрации
    * **Сообщение**: Текст, Дата, Пользователь
15. **Автомобиль и Владелец**
    * **Автомобиль**: Марка, Модель, Год выпуска
    * **Владелец**: ФИО, Адрес, Автомобили
16. **Книга и Автор**
    * **Книга**: Название, Жанр, Год выпуска
    * **Автор**: ФИО, Биография, Книги
17. **Турист и Путешествие**
    * **Турист**: ФИО, Email, Телефон
    * **Путешествие**: Направление, Дата начала, Дата окончания, Турист
18. **Сайт и Комментарий**
    * **Сайт**: Название, URL, Дата создания
    * **Комментарий**: Текст, Дата, Сайт
19. **Группа и Участник**
    * **Группа**: Название, Описание, Дата создания
    * **Участник**: ФИО, Роль, Группа
20. **Магазин и Отзыв**
    * **Магазин**: Название, Адрес, Телефон
    * **Отзыв**: Оценка, Текст, Магазин

**Пример разработки приложения**

**Тема:** Интернет-магазин книг (сущности Book и Author)

**Шаг 1. Анализ предметной области**

**Сущности и их свойства**

1. Author**(Автор)**
   * Id (int, первичный ключ)
   * Name (string, обязательное)
   * Country (string, опциональное)
2. Book**(Книга)**
   * Id (int, первичный ключ)
   * Title (string, обязательное)
   * Year (int, год издания)
   * AuthorId (int, внешний ключ) → связь "многие к одному" с Author

**CRUD-операции**

| **Сущность** | **Create** | **Read (все/по Id)** | **Update** | **Delete** |
| --- | --- | --- | --- | --- |
| Author | + | + | + | + |
| Book | + | + | + | + |

**Шаг 2. Проектирование слоёв приложения**

**Схема архитектуры (рисунок 1)**

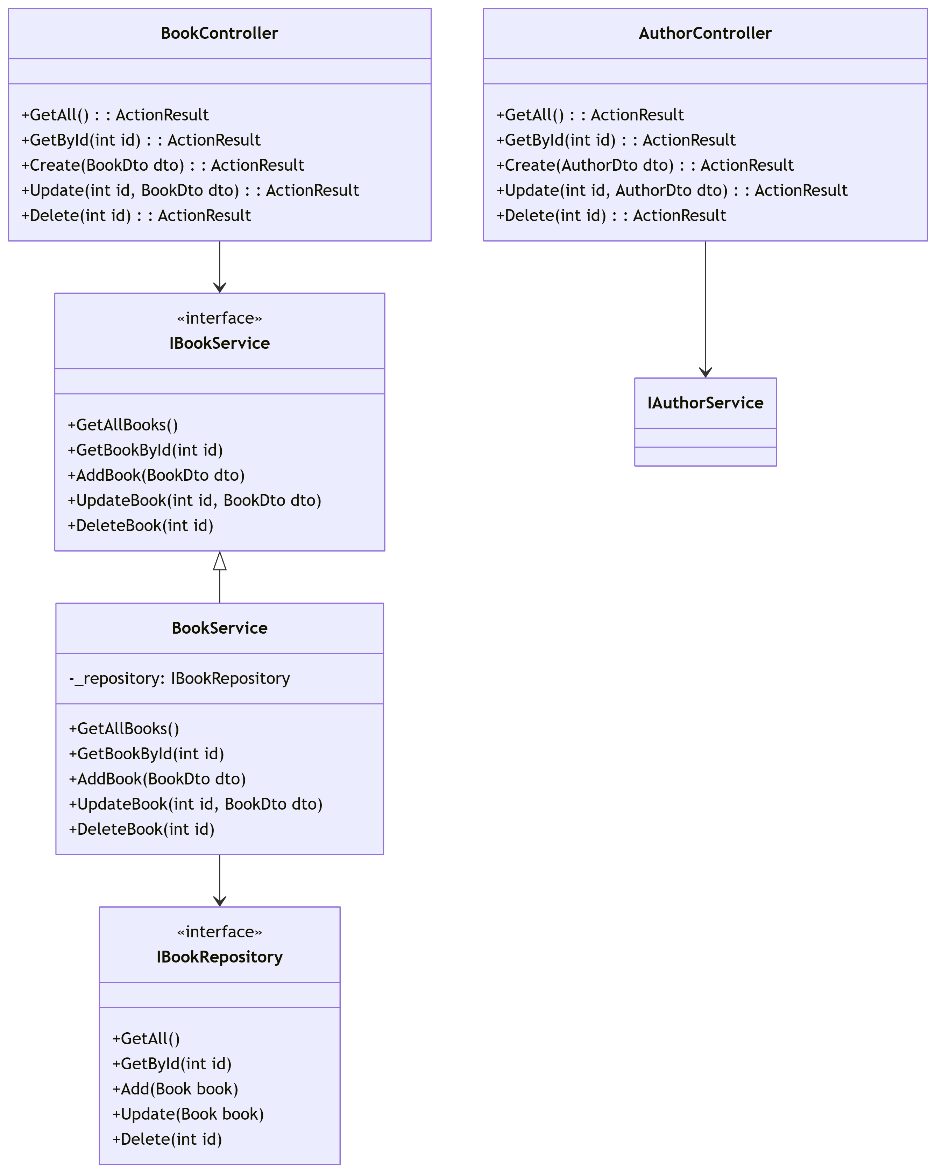
****

Рисунок 1 – Схема архитектуры программы

**Пояснения:**

1. **Контроллеры**
   * Принимают HTTP-запросы и возвращают ответы.
   * Не содержат бизнес-логики (только валидацию входных данных).
2. **Сервисы**
   * IBookService и IAuthorService — интерфейсы для абстракции бизнес-логики.
   * BookService — реализация: проверка условий (например, "нельзя удалить автора, если у него есть книги").
3. **Репозитории**
   * IBookRepository — интерфейс для работы с БД через ORM (например, Entity Framework Core).

**Шаг 3. Проектирование API**

**Эндпоинты для**Book

| **Метод** | **URL** | **Действие** | **Пример тела запроса (JSON)** |
| --- | --- | --- | --- |
| GET | /api/books | Получить все книги | – |
| GET | /api/books/1 | Получить книгу по Id | – |
| POST | /api/books | Добавить книгу | {"title": "CLR via C#", "year": 2020, "authorId": 1} |
| PUT | /api/books/1 | Обновить книгу | {"title": "CLR via C# 2nd Ed.", "year": 2023, "authorId": 1} |
| DELETE | /api/books/1 | Удалить книгу | – |

**Пример ответа для**GET /api/books/1**:**

{

"id": 1,

"title": "CLR via C#",

"year": 2020,

"authorId": 1,

"author": {

"id": 1,

"name": "Джеффри Рихтер",

"country": "США"

}

}

**Шаг 4. План тестирования**

**Сценарии:**

1. **Автор**
   * Попытка удалить автора, у которого есть книги → ошибка 400 Bad Request.
   * Создание автора без указания страны → успех (201 Created).
2. **Книга**
   * Добавление книги с несуществующим authorId → ошибка 404 Not Found.
   * Обновление года издания на отрицательное число → ошибка 400 Bad Request.

# Содержание пояснительной записки

1. Постановка задачи. Приводится теоретический материал, использованный при написании приложения.

2. Формулировка задания и вариант. Приводится задание на лабораторную работу и вариант этого задания.

3. Описание выполняемых действий. Необходимо привести описание последовательности разработки программы, реализации используемых методов, алгоритмов, блок-схем.

4. Анализ результатов. Привести анализ входных и выходных данных. Показать результаты выполнения программного кода. Предоставить скриншоты обработки тестовых примеров. Сделать выводы.

5. Листинг программы. Привести листинг разработанного программного кода, содержание файлов входных и выходных данных.

# Используемое программное обеспечение

1. Среда программирования MS Visual Studio Community 2022 (Свободно распространяемое программное обеспечение (в учебных целях));
2. Microsoft Office Standard 2007 (Open License: 42267924);
3. Open Office (Свободно распространяемое программное обеспечение).
4. Браузер (Свободно распространяемое программное обеспечение).

# Список литературы

* + - 1. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 285 c.
      2. Биллиг, В. A. Основы объектного программирования на С# (C# 3.0, Visual Studio 2008) [Электронный ресурс] : учебное пособие / В. A. Биллиг. — Электрон. текстовые данные. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 583 c. — 978-5-4487-0145-0. — Режим доступа: http://www.iprbookshop.ru/72339.html
      3. Павловская, Т. А. Программирование на языке высокого уровня C# [Электронный ресурс] / Т. А. Павловская. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 245 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73713.html
      4. Агапов, В. П. Основы программирования на языке С# [Электронный ресурс] : учебное пособие / В. П. Агапов. — Электрон. текстовые данные. — М. : Московский государственный строительный университет, ЭБС АСВ, 2012. — 128 c. — 978-5-7264-0576-6. — Режим доступа: http://www.iprbookshop.ru/16366.html
      5. Медведев, М. А. Программирование на СИ# [Электронный ресурс] : учебное пособие / М. А. Медведев, А. Н. Медведев ; под ред. А. В. Присяжный. — Электрон. текстовые данные. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2015. — 64 c. — 978-5-7996-1561-1. — Режим доступа: http://www.iprbookshop.ru/69667.html
      6. Казанский А.А. Объектно-ориентированное программирование на языке Microsoft Visual С# в среде разработки Microsoft Visual Studio 2008 и .NET Framework. 4.3 [Электронный ресурс]: учебное пособие и практикум/ Казанский А.А.— Электрон. текстовые данные.— М.: Московский государственный строительный университет, ЭБС АСВ, 2011.— 180 c
      7. Уйманова Н.А. Основы объектно-ориентированного программирования [Электронный ресурс]: практикум/ Уйманова Н.А., Таспаева М.Г.— Электрон. текстовые данные.— Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2017.— 156 c.— Режим доступа: http://www.iprbookshop.ru/78808.html.— ЭБС «IPRbooks»
      8. Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс]: учебно-методическое пособие к лабораторным работам/ Новиков П.В.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2017.— 124 c.— Режим доступа: http://www.iprbookshop.ru/64650.html.— ЭБС «IPRbooks»