Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 1

по курсу «Программные технологии проектирования ПО ВС»

на тему «Основы разработки Web API»

Вариант 2

Выполнили

студенты группы 22ВВП1:

Зубриянова А.А.

Кондратьева В.И.

Приняли:

Патунин Д.В.

Деев М.В.

Пенза 2025

**Цель работы**

Освоить основы разработки Web API с использованием **ASP.NET Core**. Разработать базовую серверную часть интернет-магазина.

**Лабораторное задание**

1. Установить Visual Studio 2022.
2. Установить .NET версии 6.0.
3. Создать проект по шаблону – web api.
4. Запустить проект и продемонстрировать открывшуюся страницу в браузере.
5. Инициализировать директорию с проектом для системы контроля версий и добавить ваш проект в созданный репозиторий.
6. Установить библиотеку для взаимодействия с базой данных.
7. Реализовать класс Product, который инкапсулирует товар в интернет-магазине.
8. Разработать интерфейс IProductService, определяющий операции с товаром (добавление, удаление, изменение, поиск).
9. Разработать класс ProductService, реализующий интерфейс IProductService.
10. Создать контроллер ProductController, принимающий пользовательские запросы и возвращающий результат выполнения операции в формате json.

**Ход работы**

Докачали необходимые компоненты для Visual Studio 2022.

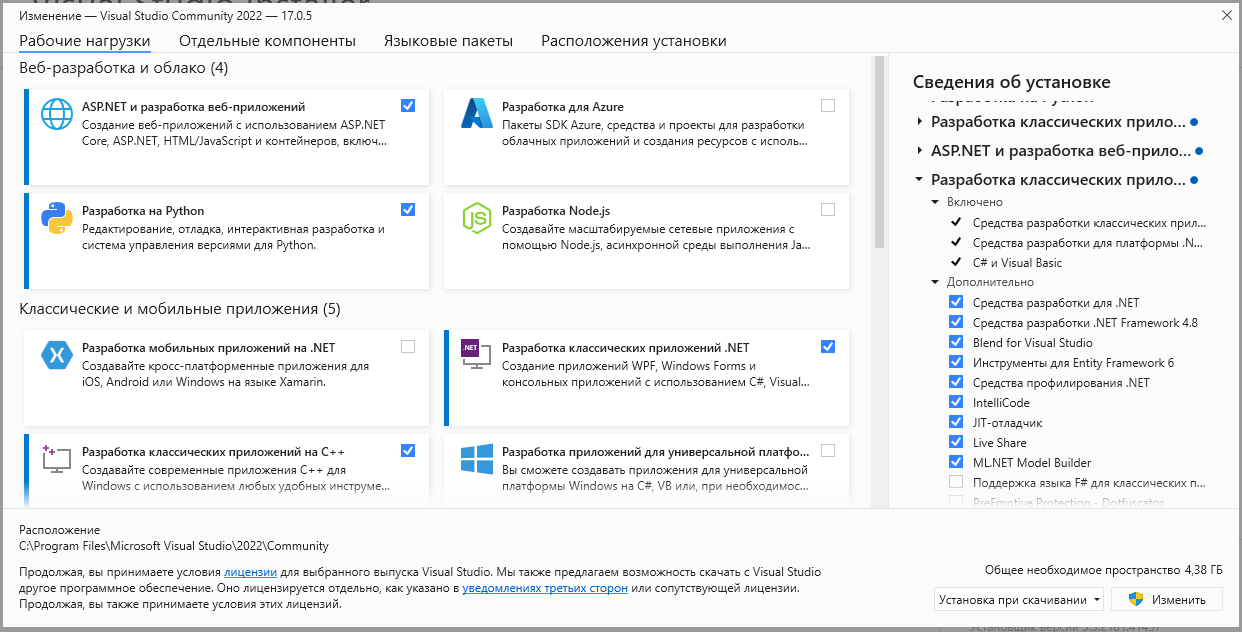


Рисунок 1 – Разработка классических приложений .Net на языке С#

Создали новый репозиторий, на который будем грузить лабораторные работы.

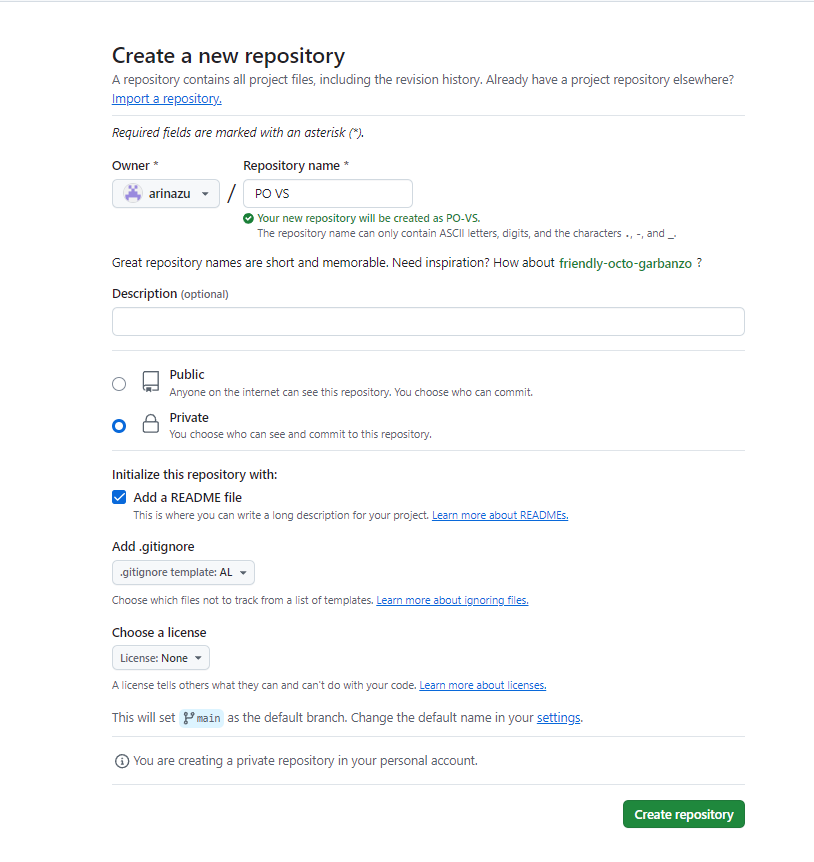


Рисунок 2 – Создание репозитория на GitHub

Выполнили установку .Net и выполнили команды, чтобы увидеть список всех установленных версий . Net SDK и список всех установленных версий . Net Runtime.

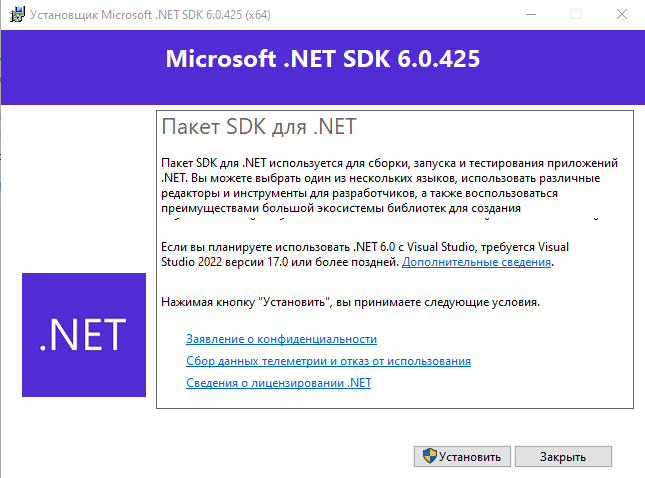


Рисунок 3 – Установка .Net

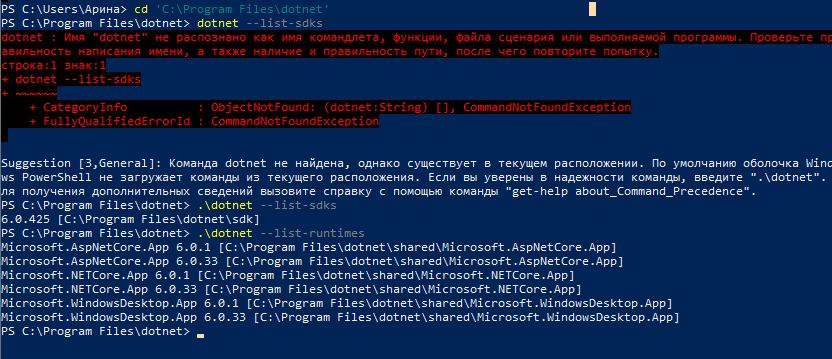


Рисунок 4 – Просмотр информации об установленных версиях .Net

1. dotnet new sln --name Magazine

Создаём файл решения. Файл решение – текстовое описание проектов, из которых состоит «общий проект».

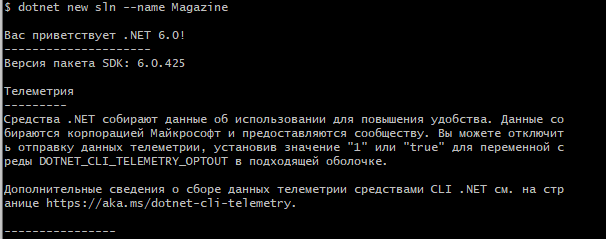


Рисунок 5 – Создание файла решения Magazine.sln

1. dotnet new classlib --name Magazine.Core

Создаём проект по шаблону «библиотека классов», в котором будут располагаться основные модели (классы), отвечающие за логику работы приложения.

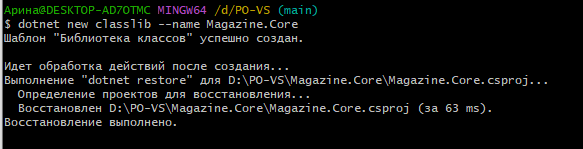


Рисунок 6 – Создание проекта Magazine.Core

1. dotnet new webapi --name Magazine.WebApi

Создаём проект по шаблону «webapi», который будет являться серверной частью приложения, то есть обрабатывать пользовательские запросы.

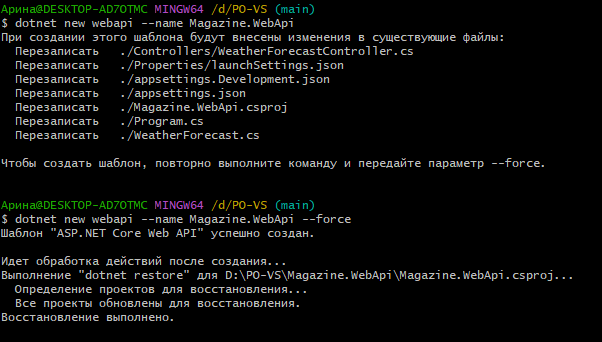


Рисунок 7 – Создание проекта Magazine.WebApi

1. dotnet sln .\Magazine.sln add .\Magazine.Core\Magazine.Core.csproj

Добавили ссылку на проект библиотеки классов.

Аналогично добавили в решение ссылку на проект webapi.

Также выполнили dotnet add .\Magazine.WebApi\Magazine.WebApi.csproj reference .\Magazine.Core\Magazine.Core.csproj.

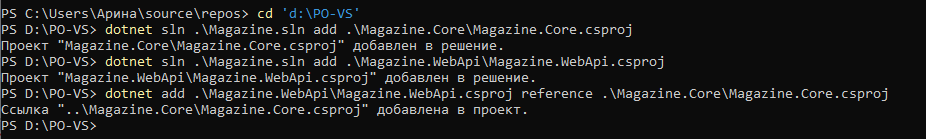


Рисунок 8 – Добавление ссылок на проекты

Запустили проект:

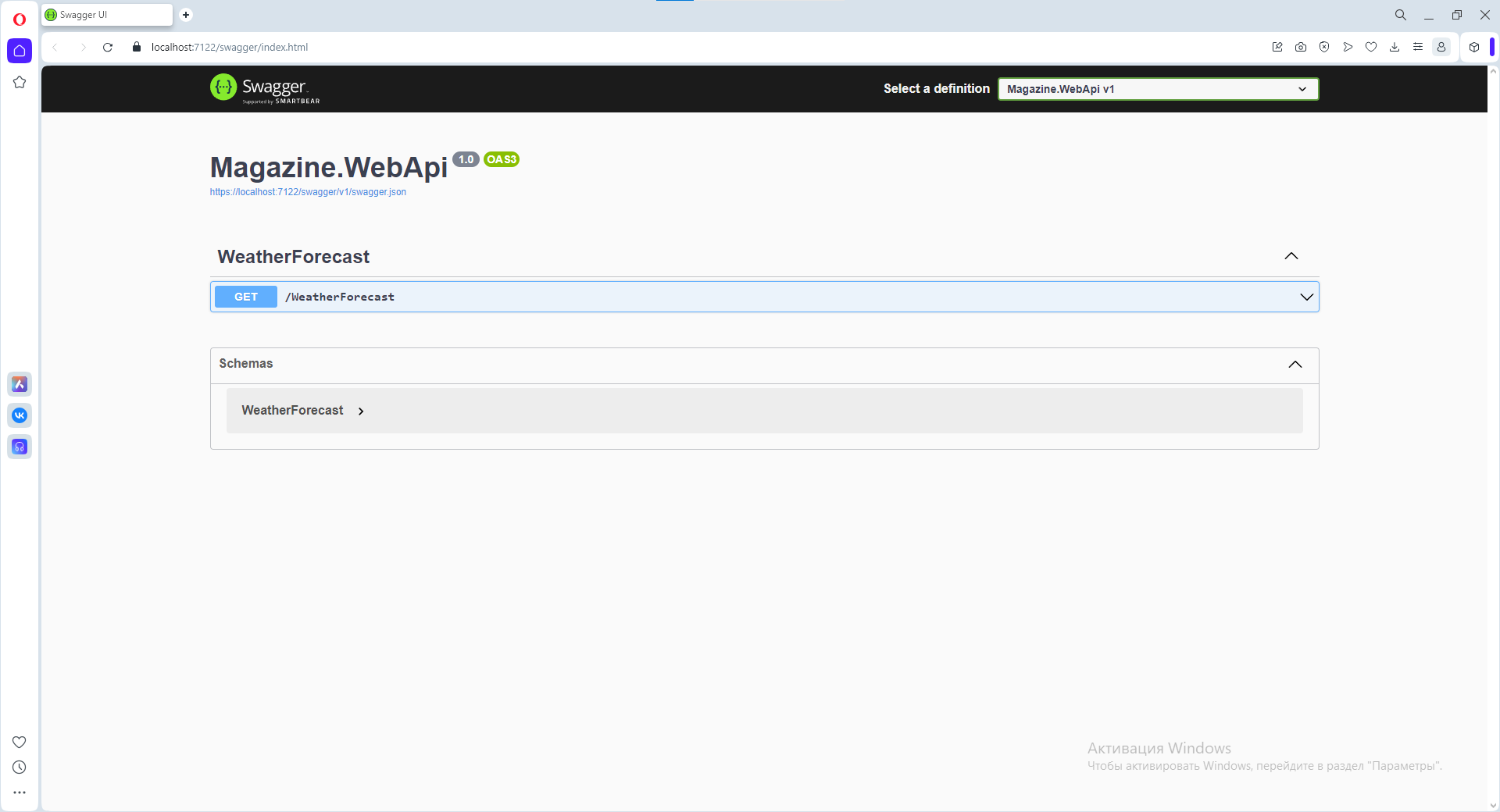
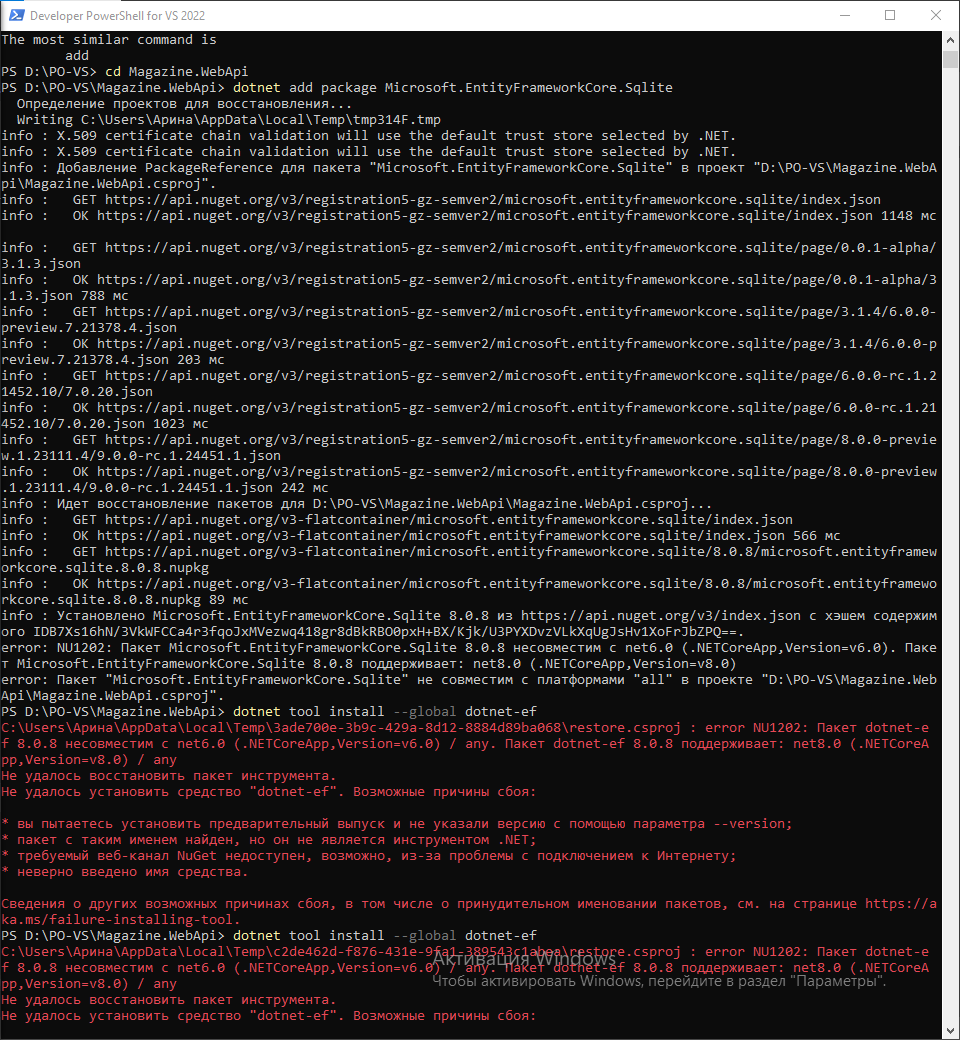
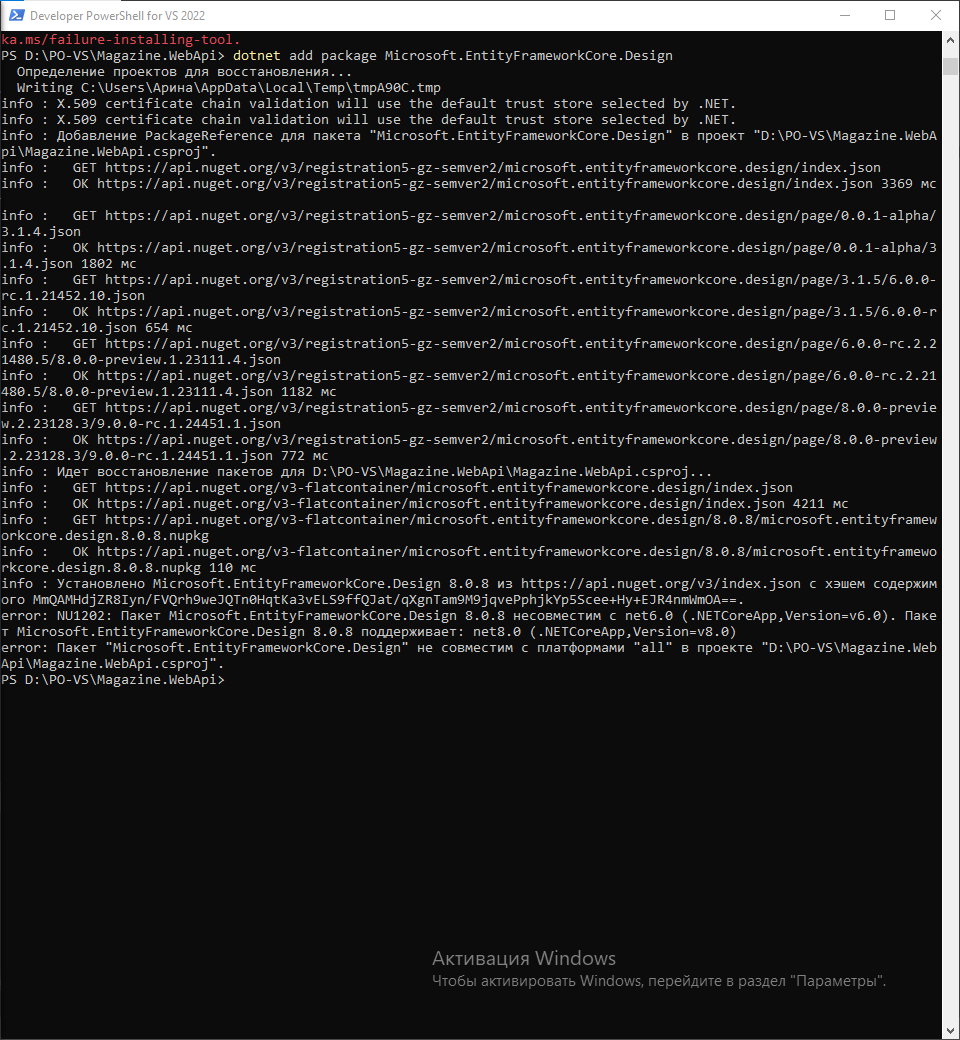


Рисунок 9 – Запуск программы

Выполнили установку необходимой библиотеки через терминал и проверили список добавленных пакетов:





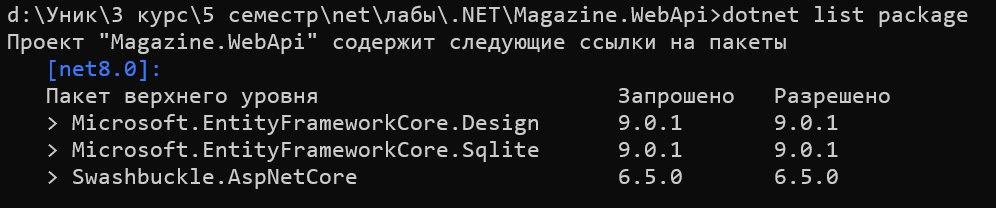


Рисунок 10 – Установка библиотеки

В проекте Magazine.Core создали папку Models, в ней создали класс Product со следующими полями:

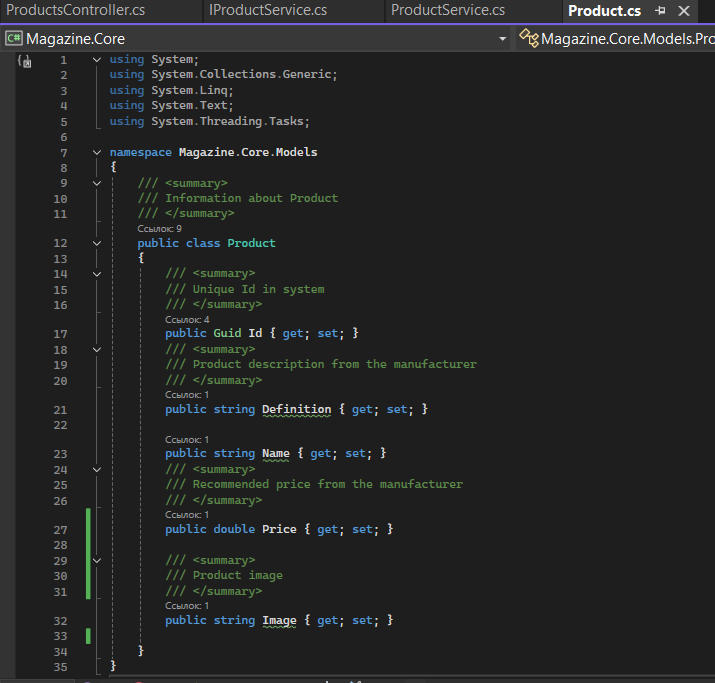


Рисунок 11 – Класс Product

В проекте Magazine.Core создайте папку Services. В ней создайте интерфейс IProductService, который описывает функционал:

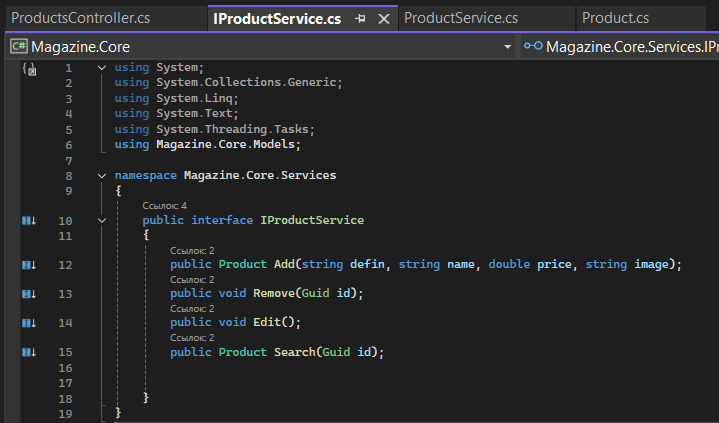


Рисунок 12 – Интерфейс IProductService

В проекте Magazine.WebApi добавить класс ProductService реализующий интерфейс IProductService.

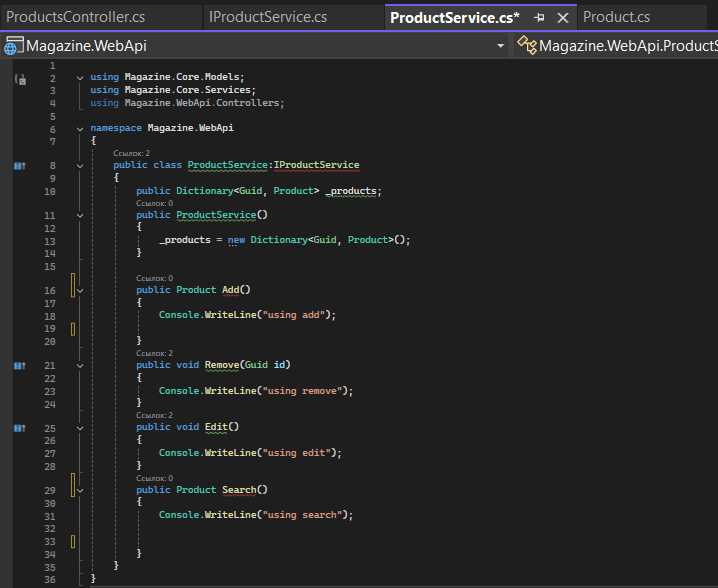


Рисунок 13 – Класс ProductService

Разработали класс ProductController с 4 методами, с помощью которых можно будет использовать функционал класс ProductService.

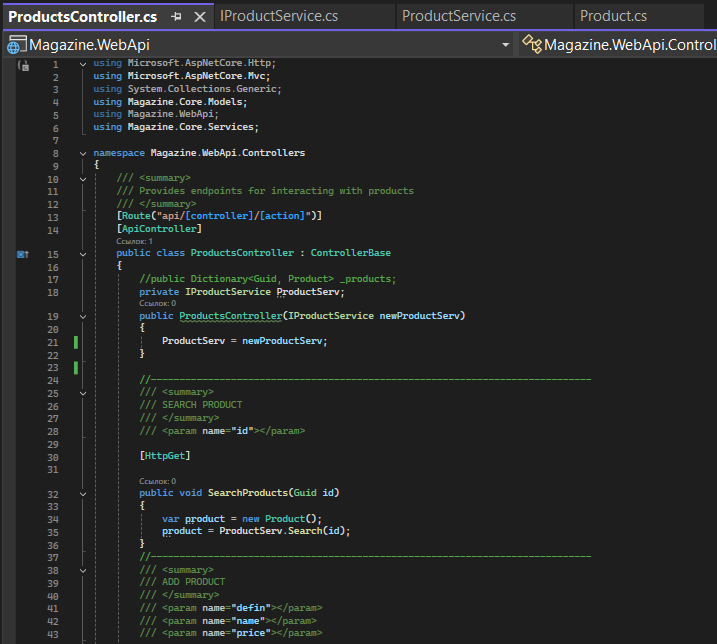


Рисунок 14 – Класс ProductController

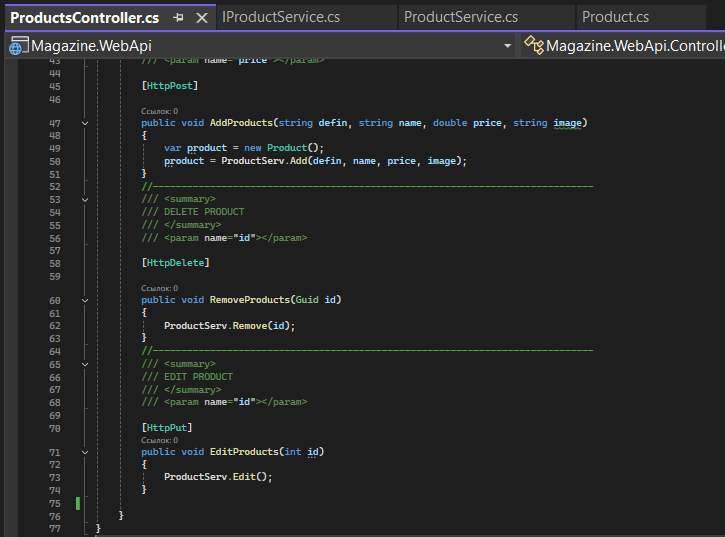


Рисунок 15 – Класс ProductController

**Вывод**

В ходе выполнения лабораторной работы была разработана базовая серверная часть интернет-магазина, освоены основы разработки Web API с использованием **ASP.NET Core**. Результаты работы программы совпали с ожидаемыми результатами, следовательно, программа работает без ошибок.