**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

****

**Московский политехнический университет**

***Факультет Информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**Лабораторная работа № 3**

**Дисциплина: Backend-разработка**

**Тема:** Создание консольного приложения с внедренными зависимостями на основе ASP.NET Core

**Выполнил: студент группы 231-333**

**Фабрикант Тимур Романович**

(Фамилия И.О)

**Дата, подпись**

(Дата) (Подпись)

**Проверил: Полубояринова А.С.**

(Фамилия И.О., степень, звание) (Оценка)

**Замечание:**

**Москва**

**2025**

**Цель**: Освоить создание консольного приложения, использующего внедрение зависимостей в ASP.NET Core.

**Задачи:**

1. Создать новый проект консольного приложения ASP.NET Core.
2. Настроить внедрение зависимостей для работы с различными сервисами или компонентами приложения.
3. Реализовать основные функции приложения, используя внедренные зависимости.
4. Протестировать работу приложения, обеспечив его функциональность в консольном режиме.

**Ход выполнения работы:**

Написание интерфейсов для сервисов

Листинг 1. IMessageService

public interface IMessageService

{

void GetMessage(string message);

}

Листинг 2. ICalculatorService

public interface ICalculatorService

{

int Add(int a, int b);

}

Написание классов, реализующие интерфейсы

Листинг 3. MessageService

public class MessageService : IMessageService

{

public void GetMessage(string message)

{

Console.WriteLine(message);

}

}

Листинг 4. CalculatorService

public class CalculatorService : ICalculatorService

{

public int Add(int a, int b)

{

return a + b;

}

}

Внедрение зависимостей для работы с различными сервисами

Листинг 5. Добавление сервисов

public class Program

{

public static void Main(string[] args)

{

var serviceProvider = new ServiceCollection()

.AddSingleton<ICalculatorService, CalculatorService>() // Добавление сервиса CalculatorService

.AddTransient<IMessageService, MessageService>() // Добавление сервиса MessageService

.BuildServiceProvider();

// Тест сервиса CalculatorService

var calculatorService = serviceProvider.GetService<ICalculatorService>();

Console.WriteLine("Первый сервис: " + calculatorService?.GetType().Name);

Console.WriteLine("Введите два числа:");

int a = Convert.ToInt32(Console.ReadLine());

int b = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Результат: " + calculatorService?.Add(a, b));

Console.WriteLine();

// Тест сервиса MessageService

var messageService = serviceProvider.GetService<IMessageService>();

Console.WriteLine("Второй сервис: " + messageService?.GetType().Name);

messageService?.GetMessage("Hello World");

Console.WriteLine();

}

}

Выполнение кода

