

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 № 5

Дисциплина: Функциональное программирование

Тема: Основы F# для функционального программирования

Выполнил(а): студент(ка) группы 221-3710

Сычугова П.А.

(Фамилия И.О.)

Дата, подпись \_\_\_\_\_  
(Дата) (Подпись)

Проверил: \_\_\_\_\_  
(Фамилия И.О., степень, звание) (Оценка)

Дата, подпись \_\_\_\_\_  
(Дата) (Подпись)

Замечания: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Москва

2025

**Цель:** Освоить основы языка F# и его функциональные конструкции.

**Задание:**

1. Разработайте набор чистых функций для выполнения общих математических операций:

- Функция, которая принимает два числа и возвращает их сумму.
- Функция, которая принимает два числа и возвращает их разность.
- Функция, которая принимает два числа и возвращает их произведение.
- Функция, которая принимает два числа и возвращает результат деления.

2. Напишите рекурсивную функцию для вычисления факториала числа.

**Требования:**

- Все функции должны быть чистыми и использовать неизменяемые данные.
- Используйте каррирование для создания специализированных функций из общих.
- Код должен быть хорошо оформлен и легко читаем.

## Ход работы

1. Чистые функции для выполнения общих математических операций

```
module Operations =  
    let sum a b = a + b  
    let difference a b = a - b  
    let product a b = a * b  
    let division a b = a / b
```

2. Каррирование

```
let incrementByOne = sum 1  
let increaseByFive = sum 5  
let multiplyByThree = product 3  
let divideByTwo x = division x 2
```

3. Рекурсивная функция для вычисления факториала числа

```
let rec computeFactorial n =
```

```
match n with
| n when n <= 1 -> 1
| _ -> n * computeFactorial (n - 1)
```

### Пример данных:

open Operations

[<EntryPoint>]

let main \_ =

printfn "Сумма: 8 + 4 = %d" (sum 8 4)

printfn "Разность: 8 - 4 = %d" (difference 8 4)

printfn "Произведение: 8 \* 4 = %d" (product 8 4)

printfn "Частное: 8 / 4 = %d" (division 8 4)

printfn "Факториал 5 = %d" (computeFactorial 5)

printfn "Увеличение на 1: %d" (incrementByOne 9)

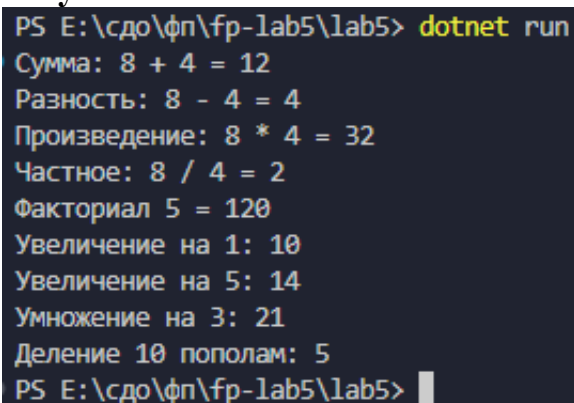
printfn "Увеличение на 5: %d" (increaseByFive 9)

printfn "Умножение на 3: %d" (multiplyByThree 7)

printfn "Деление 10 пополам: %d" (divideByTwo 10)

0

### Результат:



```
PS E:\сдо\фп\fp-lab5\lab5> dotnet run
Сумма: 8 + 4 = 12
Разность: 8 - 4 = 4
Произведение: 8 * 4 = 32
Частное: 8 / 4 = 2
Факториал 5 = 120
Увеличение на 1: 10
Увеличение на 5: 14
Умножение на 3: 21
Деление 10 пополам: 5
PS E:\сдо\фп\fp-lab5\lab5>
```