

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 № 6

Дисциплина: Функциональное программирование

Тема: Применение функционального программирования в F#

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: Применить принципы функционального программирования для разработки небольшого консольного приложения.

Задание:

- Разработайте консольное приложение "Калькулятор", которое позволяет пользователю выполнять следующие операции:
- Сложение, вычитание, умножение и деление.
- Возведение в степень.
- Вычисление квадратного корня.
- Вычисление синуса, косинуса и тангенса угла.

Требования:

- Используйте принципы функционального программирования, такие как иммутабельность данных и чистые функции.
- Используйте функции высшего порядка для обработки данных и создания новых функций.
- Консольное приложение должно быть реализовано с использованием F# и библиотек .NET.
- Интерфейс должен быть интуитивно понятным и удобным для пользователя.

Ход работы

1. Листинг

```
open System

module MathOps =

    let sum a b = a + b
    let diff a b = a - b
    let prod a b = a * b
    let divide a b = if b = 0.0 then Double.NaN else a / b
    let exponent a b = a ** b
    let root a = if a < 0.0 then Double.NaN else Math.Sqrt a
    let private degToRad x = x * Math.PI / 180.0
    let sine x = Math.Sin(degToRad x)
    let cosine x = Math.Cos(degToRad x)
    let tangent x = Math.Tan(degToRad x)
```

```
let makePower p = fun x -> exponent x p
let square = makePower 2.0
let cube = makePower 3.0
```

```
module CLI =
```

```
open MathOps
```

```
let showBanner () =
    printfn "\nWelcome to Functional Calculator \`(` ^ ∇ ^ `)"
    printfn "[1] Add      [2] Subtract"
    printfn "[3] Multiply [4] Divide"
    printfn "[5] Power    [6] Square Root"
    printfn "[7] Sine     [8] Cosine"
    printfn "[9] Tangent  [0] Quit"
    printf "Choose option (0–9): "
```

```
let inputNumber label =
    printf "%s: " label
    match Double.TryParse(Console.ReadLine()) with
    | true, v -> v
    | _ ->
        printfn "Invalid input, defaulting to 0.0"
        0.0
```

```
let execute choice =
    let displayResult value =
        if Double.IsNaN value then
            printfn "Math error or invalid input!"
        else
            printfn "=> Result: %.4f" value

    match choice with
```

```
    | 1 ->
        let x = inputNumber "Enter A"
        let y = inputNumber "Enter B"
        sum x y |> displayResult
    | 2 ->
        let x = inputNumber "Enter A"
        let y = inputNumber "Enter B"
        diff x y |> displayResult
    | 3 ->
        let x = inputNumber "Enter A"
        let y = inputNumber "Enter B"
```

```

    prod x y |> displayResult
| 4 ->
    let x = inputNumber "Enter A"
    let y = inputNumber "Enter B"
    divide x y |> displayResult
| 5 ->
    let x = inputNumber "Base"
    let y = inputNumber "Exponent"
    exponent x y |> displayResult
| 6 ->
    let x = inputNumber "Number"
    root x |> displayResult
| 7 ->
    let x = inputNumber "Angle (°)"
    sine x |> displayResult
| 8 ->
    let x = inputNumber "Angle (°)"
    cosine x |> displayResult
| 9 ->
    let x = inputNumber "Angle (°)"
    tangent x |> displayResult
| 0 ->
    printfn "Goodbye!"
| _ ->
    printfn "Invalid selection. Try again."

```

```

let rec mainLoop () =
    showBanner ()
    match Int32.TryParse(Console.ReadLine()) with
    | true, 0 -> ()
    | true, opt ->
        execute opt
        mainLoop ()
    | _ ->
        printfn "Invalid input. Please enter a digit."
        mainLoop ()

```

```

[<EntryPoint>]
let main _ =
    CLI.mainLoop ()
    0

```

Примеры работы:

1. Сложение

```

Choose option (0-9): 0
PS E:\сдо\фп\fp-lab6\lab6> dotnet run

Welcome to Functional Calculator √(^ ∇ ^)
[1] Add          [2] Subtract
[3] Multiply     [4] Divide
[5] Power        [6] Square Root
[7] Sine         [8] Cosine
[9] Tangent      [0] Quit
Choose option (0-9): 1
Enter A: 1,5
Enter B: 8
=> Result: 9.5000

```

2. Вычитание

```

Welcome to Functional Calculator √(^ ∇ ^)
[1] Add          [2] Subtract
[3] Multiply     [4] Divide
[5] Power        [6] Square Root
[7] Sine         [8] Cosine
[9] Tangent      [0] Quit
Choose option (0-9): 2
Enter A: 7
Enter B: 10000
=> Result: -9993.0000

```

3. Умножение

```

Welcome to Functional Calculator √(^ ∇ ^)
[1] Add          [2] Subtract
[3] Multiply     [4] Divide
[5] Power        [6] Square Root
[7] Sine         [8] Cosine
[9] Tangent      [0] Quit
Choose option (0-9): 3
Enter A: 4
Enter B: 10
=> Result: 40.0000

```

4. Разность

```

Welcome to Functional Calculator √(^ ∇ ^)
[1] Add          [2] Subtract
[3] Multiply     [4] Divide
[5] Power        [6] Square Root
[7] Sine         [8] Cosine
[9] Tangent      [0] Quit
Choose option (0-9): 4
Enter A: 9
Enter B: 4
=> Result: 2.2500

```

5. Возведение в степень

```

Welcome to Functional Calculator <(^ ▽ ^)
[1] Add      [2] Subtract
[3] Multiply [4] Divide
[5] Power    [6] Square Root
[7] Sine     [8] Cosine
[9] Tangent  [0] Quit
Choose option (0-9): 5
Base: 7
Exponent: 3
=> Result: 343.0000

```

6. Квадратный корень

```

Welcome to Functional Calculator <(^ ▽ ^)
[1] Add      [2] Subtract
[3] Multiply [4] Divide
[5] Power    [6] Square Root
[7] Sine     [8] Cosine
[9] Tangent  [0] Quit
Choose option (0-9): 6
Number: 7
=> Result: 2.6458

```

7. Синус

```

Welcome to Functional Calculator <(^ ▽ ^)
[1] Add      [2] Subtract
[3] Multiply [4] Divide
[5] Power    [6] Square Root
[7] Sine     [8] Cosine
[9] Tangent  [0] Quit
Choose option (0-9): 7
Angle (°): 180
=> Result: 0.0000

```

8. Косинус

```

Welcome to Functional Calculator <(^ ▽ ^)
[1] Add      [2] Subtract
[3] Multiply [4] Divide
[5] Power    [6] Square Root
[7] Sine     [8] Cosine
[9] Tangent  [0] Quit
Choose option (0-9): 8
Angle (°): 180
=> Result: -1.0000

```

9. Тангенс

```

Welcome to Functional Calculator <(^ ▽ ^)
[1] Add      [2] Subtract
[3] Multiply [4] Divide
[5] Power    [6] Square Root
[7] Sine     [8] Cosine
[9] Tangent  [0] Quit
Choose option (0-9): 9
Angle (°): 145
=> Result: -0.7002

```