

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет “Радиотехнический”
Кафедра “Системы обработки информации и управления”**

Курс «Парадигмы и конструкции языков программирования»

**Отчет по лабораторной работе №4
Вариант №18**

Выполнил:
студент группы РТ5-31Б:
Филатов И. В.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2025 г.

Текст программы

Program.fs

```

let tryParseDouble (input: string) =
    match Double.TryParse(input) with
    | (true, num) -> Some num
    | _ -> None

let read_coef_cons coefficientName isA =
    let rec readLoop () =
        printf $"\\nВведите коэффициент {coefficientName}: "
        let input = Console.ReadLine()
        match tryParseDouble input with
        | Some num ->
            if isA && num = 0.0 then
                printfn "Коэффициент A не может быть равен 0 для биквадратного уравнения."
                printfn "Пожалуйста, введите ненулевое значение для A."
                readLoop ()
            else
                num
        | None ->
            printf "Некорректный ввод. Пожалуйста, введите действительное число."
            readLoop ()
        readLoop ()

let get_coef_cons () =
    printfn "\\nВведите коэффициенты биквадратного уравнения Ax^4 + Bx^2 + C = 0"
    let a = read_coef_cons "A" true
    let b = read_coef_cons "B" false
    let c = read_coef_cons "C" false
    (a, b, c)

let solve_biquadratic a b c =
    let discriminant = b * b - 4.0 * a * c

    if discriminant < 0.0 then
        []
    else
        let t1 = (-b + sqrt discriminant) / (2.0 * a)
        let t2 = (-b - sqrt discriminant) / (2.0 * a)

        let mutable roots = []

        if t1 >= 0.0 then
            let root1 = sqrt t1
            let root2 = -sqrt t1
            roots <- root1 :: root2 :: roots

        if t2 >= 0.0 then
            let root3 = sqrt t2
            let root4 = -sqrt t2
            roots <- root3 :: root4 :: roots
        roots |> List.distinct

```

```

let printColor color text =
    let currentColor = Console.ForegroundColor
    Console.ForegroundColor <- color
    printfn "%s" text
    Console.ForegroundColor <- currentColor

let parseCommandLineArgs (args: string[]) =
    if args.Length < 3 then
        None
    else
        match tryParseDouble args.[0], tryParseDouble args.[1], tryParseDouble args.[2] with
        | Some a, Some b, Some c ->
            if a = 0.0 then
                printColor ConsoleColor.Yellow "Ошибка: коэффициент A = 0, уравнение не
является биквадратным."
                printColor ConsoleColor.Yellow "Необходимо ввести коэффициенты с
клавиатуры."
                None
            else
                Some (a, b, c)
        | _ -> None

[<EntryPoint>]
let main argv =
    printfn "Лабораторная работа #1"
    let a, b, c =
        match parseCommandLineArgs argv with
        | Some (a, b, c) ->
            printfn "A = %f, B = %f, C = %f" a b c
            (a, b, c)
        | None ->
            get_coef_cons ()

    printfn "\nРешение биквадратного уравнения: %fx^4 + %fx^2 + %f = 0" a b c
    let roots = solve_biquadratic a b c
    if List.isEmpty roots then
        printColor ConsoleColor.Red "Действительных корней нет."
    else
        printColor ConsoleColor.Green "Найденные корни:"
        roots |> List.iter (fun root ->
            printColor ConsoleColor.Green (sprintf "x = %f" root))

```

Результаты

Лабораторная работа #1

Введите коэффициенты биквадратного уравнения Ax^4 + Bx^2 + C = 0

Введите коэффициент A: 1

Введите коэффициент B: -4

Введите коэффициент C: 1

Решение биквадратного уравнения: 1.000000x^4 + -4.000000x^2 + 1.000000 = 0

Найденные корни:

x = 0.517638

x = -0.517638

x = 1.931852

x = -1.931852