

**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 № 1
«Основные конструкции языка Python»
Вариант № 5**

Выполнил:
студент группы РТ5-31Б
Деревянкина М. О.
Подпись и дата:

Проверил:
к.т.н., доц. каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2025 г.

Описание задания

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов A , B , C , вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты A , B , C могут быть заданы в виде параметров командной строки (вариант задания параметров приведен в конце файла с примером кода). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.
4. Если коэффициент A , B , C введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент — это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.
5. Дополнительное задание 1 (*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

Текст программы

```
open System

let readCoef name =
    let mutable ok = false
    let mutable v = 0.0
    while not ok do
        Console.WriteLine($"{name} = ")
        let s = Console.ReadLine()
        let okParse, x = Double.TryParse s
        if okParse then
```

```

        v <- x
        ok <- true
    else
        Console.WriteLine("Некорректный ввод, попробуйте ещё раз.")
    v
}

let printRootsFromY (y: float) defColor =
    if y < 0.0 then
        Console.ForegroundColor <- ConsoleColor.Red
        Console.WriteLine("Действительных корней нет (y < 0).")
    elif y = 0.0 then
        Console.ForegroundColor <- ConsoleColor.Green
        Console.WriteLine("x = 0")
    else
        let s = Math.Sqrt y
        Console.ForegroundColor <- ConsoleColor.Green
        Console.WriteLine($"x1 = {s}")
        Console.WriteLine($"x2 = {-s}")
        Console.ForegroundColor <- defColor

let printRootsFromYReturn (y: float) =
    if y < 0.0 then
        false
    elif y = 0.0 then
        Console.WriteLine("x = 0")
        true
    else
        let s = Math.Sqrt y
        Console.WriteLine($"x = {s}")
        Console.WriteLine($"x = {-s}")
        true

[<EntryPoint>]
let main argv =
    let def = ConsoleColor

    let a, b, c =
        if argv.Length >= 3 then
            let okA, a = Double.TryParse argv[0]
            let okB, b = Double.TryParse argv[1]
            let okC, c = Double.TryParse argv[2]
            if okA && okB && okC then
                a, b, c
            else
                Console.ForegroundColor <- ConsoleColor.Red
                Console.WriteLine("Ошибка: неверные параметры командной строки.")
                Console.ForegroundColor <- def
                Environment.Exit 1
                0.0, 0.0, 0.0
        else

```

```

    let a = readCoef "A"
    let b = readCoef "B"
    let c = readCoef "C"
    a, b, c

Console.WriteLine($"Уравнение: {a} * x^4 + {b} * x^2 + {c} = 0")

if a = 0.0 then
    // B*y + C = 0
    if b = 0.0 then
        if c = 0.0 then
            Console.ForegroundColor <- ConsoleColor.Green
            Console.WriteLine("Бесконечно много корней (тождество 0 = 0).")
        else
            Console.ForegroundColor <- ConsoleColor.Red
            Console.WriteLine("Корней нет.")
            Console.ForegroundColor <- def
        else
            let y = -c / b
            printRootsFromY y def
    else

        let d = b * b - 4.0 * a * c
        Console.WriteLine($"D = {d}")

        if d < 0.0 then
            Console.ForegroundColor <- ConsoleColor.Red
            Console.WriteLine("Корней нет (D < 0).")
            Console.ForegroundColor <- def
        elif d = 0.0 then
            let y = -b / (2.0 * a)
            printRootsFromY y def
        else
            let sq = Math.Sqrt d
            let y1 = (-b + sq) / (2.0 * a)
            let y2 = (-b - sq) / (2.0 * a)

            Console.ForegroundColor <- ConsoleColor.Green
            let mutable any = false
            if printRootsFromYReturn y1 then any <- true
            if printRootsFromYReturn y2 then any <- true

            if not any then
                Console.ForegroundColor <- ConsoleColor.Red
                Console.WriteLine("Действительных корней нет (y1 < 0 и y2 < 0).")

            Console.ForegroundColor <- def

```

Экранные формы с примерами выполнения программы

```
PS D:\1C> dotnet run
A = 2
B = -5
C = 4
Уравнение:  $2 * x^4 + -5 * x^2 + 4 = 0$ 
D = -7
Корней нет ( $D < 0$ ).
PS D:\1C> █
```