**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

# «Основные конструкции языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Борисова К.А. |  | Гапанюк Ю.Е. |

Москва, 2024 г.

**Постановка задачи:**

Разработать программу для решения биквадратного уравнения.

Программа должна быть разработана в виде консольного приложения на языке Python.

Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта). Коэффициенты А, В, С могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры. Описание работы с параметрами командной строки. Если коэффициент А, В, С 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.

Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.

**Текст программы:**

1lab1.py:

*import* math

def is\_float(*string*):

*try*:

        float(*string*)

*return* True

*except*:

*return* False

def get\_coef(*prompt*):

    coef\_str = 'test'

*while* is\_float(coef\_str) == False:

        coef\_str = input(*prompt*)

    coef = float(coef\_str)

*return* coef

def solving(*a*, *b*, *c*):

    D = *b*\*\*2 - 4\**a*\**c*

*if* D > 0:

        x1 = (-*b* + math.sqrt(D)) / (2\**a*)

        x2 = (-*b* - math.sqrt(D)) / (2\**a*)

*return* x1, x2

*elif* D == 0:

        x = -*b* / (2\**a*)

*return* x

*else*:

*return* "No real roots"

*while*(True):

    a = get\_coef("Введите коэффициент a: ")

    b = get\_coef("Введите коэффициент b: ")

    c = get\_coef("Введите коэффициент c: ")

    res = solving(a, b, c)

    print("Результаты:")

*if* isinstance(res, tuple):

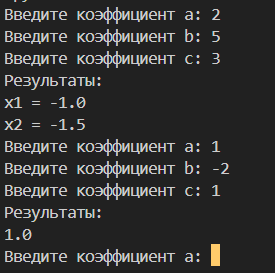
        print("x1 =", res[0])

        print("x2 =", res[1])

*else*:

        print(res)

**Анализ результатов:**



2lab1.py:

*import* math

def is\_float(*string*):

*try*:

        float(*string*)

*return* True

*except*:

*return* False

def get\_coef(*prompt*):

    coef\_str = 'test'

*while* is\_float(coef\_str) == False:

        coef\_str = input(*prompt*)

    coef = float(coef\_str)

*return* coef

class Solver:

    def \_\_init\_\_(*self*, *a*, *b*, *c*):

*self*.a = *a*

*self*.b = *b*

*self*.c = *c*

    def solve(*self*):

        D = *self*.b \*\* 2 - 4 \* *self*.a \* *self*.c

*if* D < 0:

*return* []

*elif* D == 0:

            x = -*self*.b/2\**self*.a

*return* [x]

*else*:

            x1 = (-*self*.b + math.sqrt(D)) / (2 \* *self*.a)

            x2 = (-*self*.b - math.sqrt(D)) / (2 \* *self*.a)

*return* [x1, x2]

*while*(True):

    a = get\_coef("Введите коэффициент a: ")

    b = get\_coef("Введите коэффициент b: ")

    c = get\_coef("Введите коэффициент c: ")

    solver = Solver(a, b, c)

    solutions = solver.solve()

    print("Результаты:")

*if* not solutions:

        print("Уравнение не имеет действительных корней")

*elif* len(solutions) == 1:

        print(f"Уравнение имеет один корень: {solutions[0]}")

*elif* len(solutions) == 2:

        print(f"Уравнение имеет два корня: {solutions[0]} и {solutions[1]}")

*elif* len(solutions) == 4:

        print(f"Уравнение имеет четыре корня: {solutions[0]}, {solutions[1]}, {solutions[2]} и {solutions[3]}")

    print("")

**Анализ результатов:**

