Московский государственный технический университет им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1
«Основные конструкции языка Python»

Выполнил:
Студент группы ИУ5-31Б
Попов А.С.
Подпись и дата:

Проверил:
Преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2025 г.

Задание

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

biquadratic_oop.py

```
import sys
import math

class BiquadraticSolver:
    def __init__(self, a: float, b: float, c: float):
        self.a = a
        self.b = b
        self.c = c

    def discriminant(self):
        return self.b*self.b - 4*self.a*self.c

    def solve(self):
        d = self.discriminant()
        print(f"Дискриминант: {d}")
        if d < 0:
            print("Действительных корней нет.")
            return []

        if d > 1e16:
            print("Дискриминант слишком большой.")
            return []

        sqrt_d = math.sqrt(d)
        x1 = (-self.b + sqrt_d) / (2*self.a)
        x2 = (-self.b - sqrt_d) / (2*self.a)

        roots = [x1]
        if x1 != x2:
            roots = [x1, x2]

        return roots

def get_coefficient(name: str, args: [str], index: int):
    while True:
        try:
            if len(args) > index:
                return float(args[index])

            return float(input(f"Введите коэффициент {name}: "))
        except ValueError:
            print(f"Некорректное значение для {name}, попробуйте
снова.")
            if len(args) > index:
```

```
                args[index] = None

def main():
    args = sys.argv
    a = get_coefficient("A", args, 1)
    if a == 0:
        print("Коэффициент A в квадратном уравнении не может быть
равен 0.")
        return

    b = get_coefficient("B", args, 2)
    c = get_coefficient("C", args, 3)

    solver = BiquadraticSolver(a, b, c)
    roots = solver.solve()
    if roots:
        print("Действительные корни:", roots)
        return

    print("Корней нет.")

if __name__ == "__main__":
    main()
```

biquadratic_procedural.py

```
import sys
import math

def get_coefficient(name, args, index):
    while True:
        try:
            if len(args) > index:
                return float(args[index])

            return float(input(f"Введите коэффициент {name}: "))
        except ValueError:
            print(f"Некорректное значение для {name}, попробуйте
снова.")
            if len(args) > index:
                args[index] = None

def solve_biquadratic(a, b, c):
    d = b*b - 4*a*c
    print(f"Дискриминант: {d}")
    if d < 0:
        print("Действительных корней нет.")
        return []
```

```
    if d > 1e16:
        print("Дискриминант слишком большой.")
        return []

    sqrt_d = math.sqrt(d)
    x1 = (-b + sqrt_d) / (2*a)
    x2 = (-b - sqrt_d) / (2*a)

    roots = [x1]
    if x1 != x2:
        roots = [x1, x2]

    return roots

def main():
    args = sys.argv
    a = get_coefficient("A", args, 1)
    if a == 0:
        print("Коэффициент A в квадратном уравнении не может быть
равен 0.")
        return

    b = get_coefficient("B", args, 2)
    c = get_coefficient("C", args, 3)

    roots = solve_biquadratic(a, b, c)
    if roots:
        print("Действительные корни:", roots)
        return

    print("Корней нет.")

if __name__ == "__main__":
    main()
```

Скриншоты работы программы

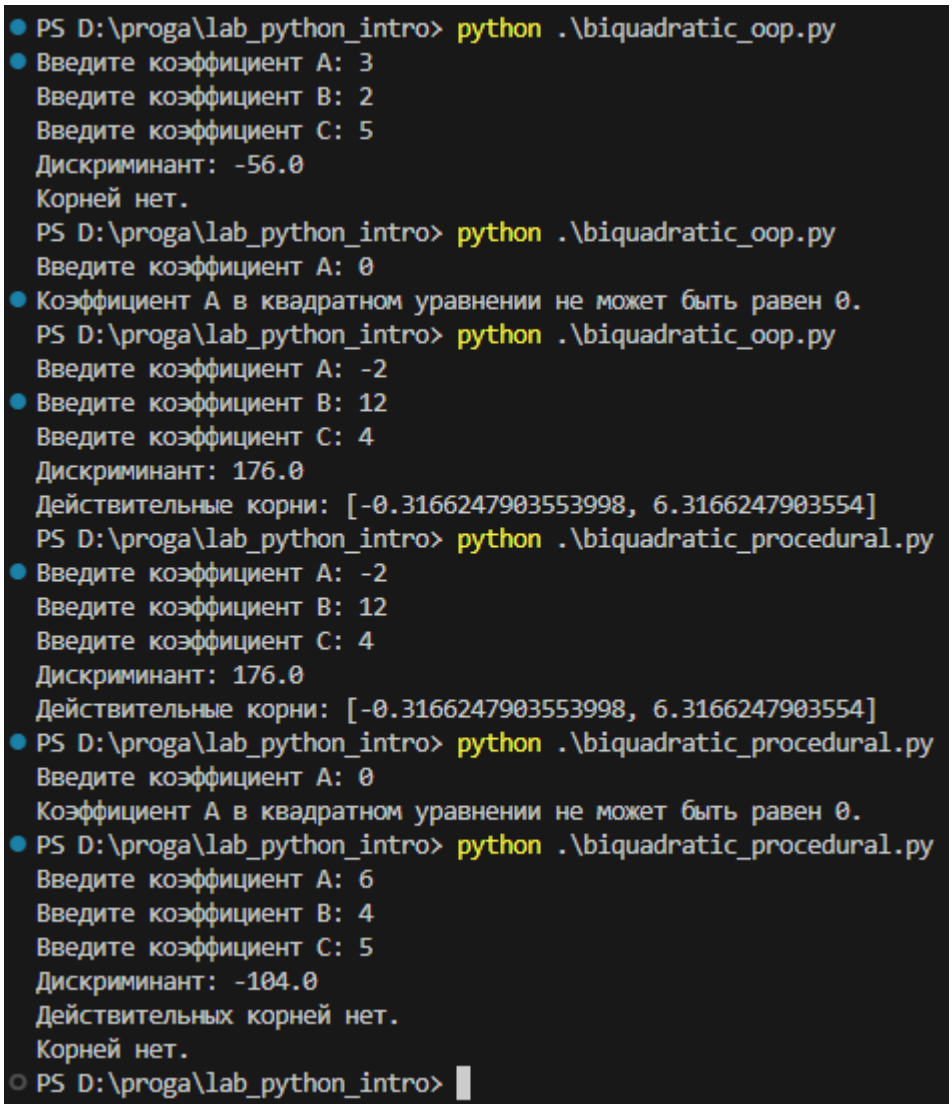

Рис. 1 – Работа приложения