|  |  |
| --- | --- |
| Защищено: Гапанюк Ю.Е.  " " **2025** г. | Демонстрация:  Константинов А.А.  " " **2025** г. |

Отчет по лабораторной работе № 1 по курсу Парадигмы и конструкции языков программирования

**Тема работы: " Основные конструкции языка Python"**

5

(количество листов)

|  |  |
| --- | --- |
|  |  |
| студент группы ИУ5Ц-53Б |  |
|  | (подпись) |
| Константинов А.А. | " " **2025** г. |

Москва, МГТУ - **2025**

# **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 А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С 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 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.
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 С++).

# **Листинг программы**

Часть 1

Процедурная версия

import sys

import math

def input\_coefficient(name, args, index):

    """Получает коэффициент (из командной строки или с клавиатуры) с проверкой корректности."""

    while True:

        try:

            value = args[index] if len(args) > index else input(f"Введите коэффициент {name}: ")

            return float(value)

        except ValueError:

            print(f"Ошибка: коэффициент {name} должен быть числом. Повторите ввод.")

def solve\_biquadratic(a, b, c):

    """Решает биквадратное уравнение A\*x^4 + B\*x^2 + C = 0"""

    print(f"\nРешаем уравнение: ({a})x⁴ + ({b})x² + ({c}) = 0")

    if a == 0:

        print("Ошибка: коэффициент A не может быть равен 0 (это не биквадратное уравнение).")

        return []

    D = b\*\*2 - 4\*a\*c

    print(f"Дискриминант D = {D:.4f}")

    if D < 0:

        print("Действительных корней нет (дискриминант отрицателен).")

        return []

    y1 = (-b + math.sqrt(D)) / (2\*a)

    y2 = (-b - math.sqrt(D)) / (2\*a)

    print(f"Промежуточные значения: y₁ = {y1:.4f}, y₂ = {y2:.4f}")

    roots = []

    for y in [y1, y2]:

        if y > 0:

            roots.extend([math.sqrt(y), -math.sqrt(y)])

        elif y == 0:

            roots.append(0.0)

    if roots:

        roots = sorted(roots)

        print("Действительные корни уравнения:", ", ".join(f"{r:.4f}" for r in roots))

    else:

        print("Действительных корней нет (y < 0).")

    return roots

def main():

    args = sys.argv[1:]

    a = input\_coefficient("A", args, 0)

    b = input\_coefficient("B", args, 1)

    c = input\_coefficient("C", args, 2)

    solve\_biquadratic(a, b, c)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Часть 2

Объектно-ориентированная версия

# Объектно-ориентированная версия решения биквадратного уравнения

import sys

import math

class BiquadraticEquation:

    """Класс для решения биквадратного уравнения A\*x^4 + B\*x^2 + C = 0"""

    def \_\_init\_\_(self, a, b, c):

        self.a = a

        self.b = b

        self.c = c

    def discriminant(self):

        """Вычисляет дискриминант D = B² - 4AC"""

        return self.b\*\*2 - 4\*self.a\*self.c

    def solve(self):

        """Решает уравнение и возвращает действительные корни"""

        print(f"\nРешаем уравнение: ({self.a})x⁴ + ({self.b})x² + ({self.c}) = 0")

        if self.a == 0:

            print("Ошибка: коэффициент A не может быть равен 0 (это не биквадратное уравнение).")

            return []

        D = self.discriminant()

        print(f"Дискриминант D = {D:.4f}")

        if D < 0:

            print("Действительных корней нет (дискриминант отрицателен).")

            return []

        y1 = (-self.b + math.sqrt(D)) / (2\*self.a)

        y2 = (-self.b - math.sqrt(D)) / (2\*self.a)

        print(f"Промежуточные значения: y₁ = {y1:.4f}, y₂ = {y2:.4f}")

        roots = []

        for y in [y1, y2]:

            if y > 0:

                roots.extend([math.sqrt(y), -math.sqrt(y)])

            elif y == 0:

                roots.append(0.0)

        if roots:

            roots = sorted(roots)

            print("Действительные корни уравнения:", ", ".join(f"{r:.4f}" for r in roots))

        else:

            print("Действительных корней нет (y < 0).")

        return roots

def get\_coefficient(name, args, index):

    """Получает коэффициент из командной строки или вводом с клавиатуры"""

    while True:

        try:

            value = args[index] if len(args) > index else input(f"Введите коэффициент {name}: ")

            return float(value)

        except ValueError:

            print(f"Ошибка: коэффициент {name} должен быть числом. Повторите ввод.")

def main():

    args = sys.argv[1:]

    a = get\_coefficient("A", args, 0)

    b = get\_coefficient("B", args, 1)

    c = get\_coefficient("C", args, 2)

    equation = BiquadraticEquation(a, b, c)

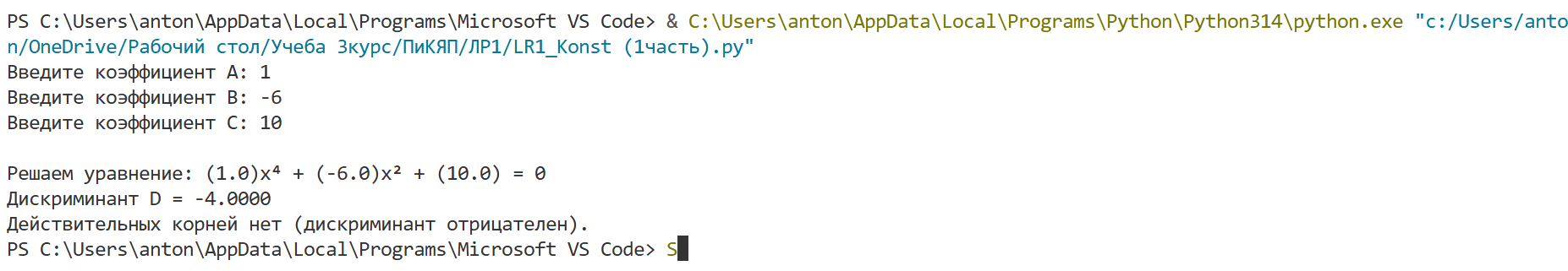
    equation.solve()

if \_\_name\_\_ == "\_\_main\_\_":

    main()

# **Результаты работы программы**

Часть 1 (Процедурная версия)



Часть 2 (Объектно-ориентированная версия)

