

Защищено:
Гапанюк Ю.Е.

Демонстрация:
Фень Н.Т.

"__" _____ 2023 г.

"__" _____ 2023 г.

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

Тема работы: " Основные конструкции языка Python. "

4

(количество листов)

Вариант № 1

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-52Б

Фень Н.Т.

(подпись)

"__" _____ 2023 г.

СОДЕРЖАНИЕ

1. Описание задания	3
2. Текст программы	3
3. Экранные формы с примерами выполнения программы.....	4

1. Описание задания

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. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. 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.

Дополнительное задание 2 (*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

2. Текст программы

```
import sys
import cmath

# Функция для ввода коэффициентов и проверки на ошибки
def get_coefficient(prompt, coef_name):
    while True:
        try:
            coef = float(input(prompt))
            return coef
        except ValueError:
            print(f"Некорректное значение для коэффициента {coef_name}. Повторите ввод.")

# Функция для вычисления корней уравнения
def solve_biquadratic(a, b, c):
    discriminant = b ** 2 - 4 * a * c

    if discriminant > 0:
        root1 = cmath.sqrt((-b + cmath.sqrt(discriminant)) / (2 * a))
        root2 = -root1
        root3 = cmath.sqrt((-b - cmath.sqrt(discriminant)) / (2 * a))
        root4 = -root3
        return root1, root2, root3, root4
    elif discriminant == 0:
        root = cmath.sqrt(-b / (2 * a))
        return root, -root
    else:
        return ()

# Функция поиска действительных корней
def findTrueRoots(roots):
    trueRoots = []
    for root in roots:
        # Проверка на действительность и на дубликатность ( 0 и -0)
```

```

        if (root.imag == 0) and (str(root.real) != "-0.0"):
            trueRoots.append(root.real)
    return trueRoots

def printTrueRoots(trueRoots):
    if len(trueRoots) == 4:
        print(
            f"Уравнение имеет четыре действительных корня: {trueRoots[0]}, {trueRoots[1]}, {trueRoots[2]}, {trueRoots[3]}"
        )
    elif len(trueRoots) == 3:
        print(f"Уравнение имеет три действительных корня: {trueRoots[0]}, {trueRoots[1]}, {trueRoots[2]}")
    elif len(trueRoots) == 2:
        print(f"Уравнение имеет два действительных корня: {trueRoots[0]}, {trueRoots[1]}")
    elif len(trueRoots) == 1:
        print(f"Уравнение имеет один действительный корень: {trueRoots[0]}")
    else:
        print("Уравнение не имеет действительных корней.")

def main():
    # Проверка ввода из консоли и получение аргументов
    if len(sys.argv) == 4:
        try:
            a = float(sys.argv[1])
            b = float(sys.argv[2])
            c = float(sys.argv[3])
        except ValueError:
            print("Некорректные коэффициенты в командной строке. Пожалуйста, введите их с клавиатуры.")
            a = get_coefficient("Введите коэффициент A: ", "A")
            b = get_coefficient("Введите коэффициент B: ", "B")
            c = get_coefficient("Введите коэффициент C: ", "C")
    else:
        a = get_coefficient("Введите коэффициент A: ", "A")
        b = get_coefficient("Введите коэффициент B: ", "B")
        c = get_coefficient("Введите коэффициент C: ", "C")
    # Получение корней
    roots = solve_biquadratic(a, b, c)
    # Вывод ответа
    trueRoots = findTrueRoots(roots)
    printTrueRoots(trueRoots)

if __name__ == "__main__":
    main()

```

3. Экранные формы с примерами выполнения программы

```

D:\Ucheba\proga5sem\lab1>python lab1.py 1 -7 10
Уравнение имеет четыре действительных корня: 2.23606797749979, -2.23606797749979, 1.4142135623730951, -1.4142135623730951

D:\Ucheba\proga5sem\lab1>python lab1.py 1 -1 0
Уравнение имеет три действительных корня: 1.0, -1.0, 0.0

D:\Ucheba\proga5sem\lab1>python lab1.py 1 0 -4
Уравнение имеет два действительных корня: 1.4142135623730951, -1.4142135623730951

D:\Ucheba\proga5sem\lab1>python lab1.py 1 1 0
Уравнение имеет один действительный корень: 0.0

D:\Ucheba\proga5sem\lab1>python lab1.py 1 0 4
Уравнение не имеет действительных корней.

```

```
D:\Ucheba\proga5sem\lab1>python lab1.py
Введите коэффициент A: 1
Введите коэффициент B: 0
Введите коэффициент C: -4
Уравнение имеет два действительных корня: 1.4142135623730951, -1.4142135623730951
```