|  |  |  |
| --- | --- | --- |
| Защищено:  Нардид А.Н.    "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |  | Демонстрация:  Нардид А.Н.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

**Отчет по лабораторной работе № 1 по курсу**

**Парадигмы и конструкции языков программирования**

**ГУИМЦ**

#### Тема работы: "Основные конструкции языка Python."

3

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы ИУ5Ц-53Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Пронин В.К. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

Москва, МГТУ - 2024 г.

# Тема и задание для выполнения лабораторной работы.

Тема работы: "Изучение основных конструкций языка Python.". Необходимо написать программу, которая решает квадратное уравнение вида:

# Листинг программы

import cmath

def get\_coefficient(name, min\_value=-1000, max\_value=1000, non\_zero=False):

    while True:

        try:

            value = float(input(f"Введите коэффициент {name}: "))

            if non\_zero and value == 0:

                print(f"Коэффициент {name} не должен быть равен нулю. Попробуйте снова.")

                continue

            if not (min\_value <= value <= max\_value):

                print(f"Коэффициент {name} должен быть в диапазоне от {min\_value} до {max\_value}. Попробуйте снова.")

                continue

            return value

        except ValueError:

            print(f"Некорректное значение для коэффициента {name}. Попробуйте снова.")

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

    discriminant = b \*\* 2 - 4 \* a \* c

    if discriminant >= 0:

        root1 = (-b + cmath.sqrt(discriminant)) / (2 \* a)

        root2 = (-b - cmath.sqrt(discriminant)) / (2 \* a)

        return root1.real, root2.real if discriminant > 0 else (root1.real, None)

    else:

        root1 = (-b + cmath.sqrt(discriminant)) / (2 \* a)

        root2 = (-b - cmath.sqrt(discriminant)) / (2 \* a)

        return root1, root2   # сложные корни

def format\_complex(num):

    # Форматируем комплексное число в виде a + bi или a - bi

    real\_part = round(num.real, 2)

    imaginary\_part = round(num.imag, 2)

    if imaginary\_part == 0:

        return f"{real\_part}"

    elif imaginary\_part > 0:

        return f"{real\_part} + {imaginary\_part}i"

    else:

        return f"{real\_part} - {abs(imaginary\_part)}i"

def main():

    print("Решение квадратного уравнения Ax^2 + Bx + C = 0")

    a = get\_coefficient("A", non\_zero=True)

    b = get\_coefficient("B")

    c = get\_coefficient("C")

    roots = solve\_quadratic(a, b, c)

    if isinstance(roots[0], complex) or roots[1] is None:

        root1 = format\_complex(roots[0])

        root2 = format\_complex(roots[1])

        print(f"Уравнение имеет комплексные корни: x1 = {root1}, x2 = {root2}")

    else:

        print(f"Корни уравнения: x1 = {roots[0]}, x2 = {roots[1]}")

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

    main()

# Результаты работы программы

C:\Users\Вячеслав\python\П\_и\_К\_я\_п\Лаб.1>python LR1.py

Решение квадратного уравнения Ax^2 + Bx + C = 0

Введите коэффициент A: 1

Введите коэффициент B: -3

Введите коэффициент C: 2

Корни уравнения: x1 = 2.0, x2 = 1.0

C:\Users\Вячеслав\python\П\_и\_К\_я\_п\Лаб.1>python LR1.py

Решение квадратного уравнения Ax^2 + Bx + C = 0

Введите коэффициент A: abc

Некорректное значение для коэффициента A. Попробуйте снова.

Введите коэффициент A: 1001

Коэффициент A должен быть в диапазоне от -1000 до 1000. Попробуйте снова.

Введите коэффициент A: 1

Введите коэффициент B: -2

Введите коэффициент C: 1

Корни уравнения: x1 = 1.0, x2 = (1.0, None)

C:\Users\Вячеслав\python\П\_и\_К\_я\_п\Лаб.1>python LR1.py

Решение квадратного уравнения Ax^2 + Bx + C = 0

Введите коэффициент A: 1

Введите коэффициент B: 2

Введите коэффициент C: 5

Уравнение имеет комплексные корни: x1 = -1.0 + 2.0i, x2 = -1.0 - 2.0i