

**Университет им. Н.Э. Баумана**

**Факультет Радиотехнический**

**Кафедра РТ5**

**Курс «Парадигмы и конструкции языков программирования»**

**Отчет по лабораторной работе №2  
«Функциональные возможности языка Python»**

Выполнил: Кудрявцев Р. В.  
Студент группы: РТ5-31Б

Проверяющий: Гапанюк Ю.Е.  
Доцент кафедры ИУ5

Москва, 2023 г.

## 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  $A$ ,  $B$ ,  $C$ , 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

```
import sys

class BiquadraticEquation:
    #конструктор класса с коэффициентами по умолчанию
    def __init__(self, a = 0.0, b = 0.0, c = 0.0):
        self.a = a
        self.b = b
```

```

self.c = cfh
self.roots = set()
self.num_roots = 0
#функция для установки коэффициентов
def coef_setter(self):
    #пробуем считать из системных атрибутов коэффициенты
    try:
        self.a = float(sys.argv[1])
        self.b = float(sys.argv[2])
        self.c = float(sys.argv[3])
    #считать из системных атрибутов не получилось, значит вводим с клавиатуры
    except:
        while True:
            try:
                print('Введите коэффициент A: ')
                self.a = float(input())
                break
            except:
                pass
        while True:
            try:
                print('Введите коэффициент B: ')
                self.b = float(input())
                break
            except:
                pass
        while True:
            try:
                print('Введите коэффициент C: ')
                self.c = float(input())
                break
            except:
                pass
#функция вычисления корней
def calculate_roots(self):
    self.roots = set()
    self.num_roots = 0
    a = self.a
    b = self.b
    c = self.c
    d = b**2 - 4*a*c
    if d == 0:
        t = -1*b/(2*a)
        if t == 0:
            self.roots.add(0)
        elif t > 0:
            for j in {(-1)**i*(t**0.5) for i in range(0, 2)}:
                self.roots.add(j)
    elif d > 0:
        t1 = (-1*b+d**0.5)/(2*a)
        t2 = (-1*b-d**0.5)/(2*a)

```

```

        if t1 == 0:
            self.roots.add(0)
        elif t1 > 0:
            for j in {(-1)**i*(t1**0.5) for i in range(0, 2)}:
                self.roots.add(j)
        if t2 == 0:
            self.roots.add(0)
        elif t2 > 0:
            for j in {(-1)**i*(t2**0.5) for i in range(0, 2)}:
                self.roots.add(j)
        self.num_roots = len(self.roots)
        if self.num_roots == 0:
            self.solution_exception()
#функция вывода коэффициентов
def coef_print(self):
    print('Коэффициент A: {}, Коэффициент B: {}, Коэффициент C:
{}').format(self.a, self.b, self.c))
#функция вывода корней
def roots_print(self):
    print('Решения уравнения: ' + ' ,'.join(map(str, self.roots)))
#функция вывода ошибки при вычислении корней
def solution_exception(self):
    print('Нет действительных корней')

def main():
    equation = BiquadraticEquation()
    equation.coef_setter()
    equation.calculate_roots()
    equation.roots_print()

if __name__ == '__main__':
    main()

```

## Экранные формы с примерами выполнения программы

Введите коэффициент A:

4

Введите коэффициент B:

-4

Введите коэффициент C:

1

Решения уравнения: 0.7071067811865476 , -0.7071067811865476

PS D:\учеба\парадигмы программирования\лабы\лаб1>

PS D:\учеба\парадигмы программирования\лабы\лаб1> py lab01.py 1 -10 9

Решения уравнения: 1.0 , 3.0 , -3.0 , -1.0

—