



**Московский государственный технический
университет им. Н. Э. Баумана
(МГТУ им. Н. Э. Баумана)**

Факультет «Информатика и системы управления»

**Кафедра «Система обработки информации и управления»
(ИУ5)**

Курс «Парадигмы и конструкции языков программирования»

**Лабораторная работа №1
«Основные конструкции языка Python»**

Выполнил:
студент группы ИУ5-32Б:
Микаелян С. В.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

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. 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.

Код программы:

Файл lab_procedure.py

```
import sys
import math

def get_coef(index, prompt):
    try:
        coef_str = sys.argv[index]
    except:
        print(prompt)
        coef_str = input()
    coef = float(coef_str)
    return coef

def get_roots(a, b, c):
    pre_result = []
    result = []
    D = b*b - 4*a*c
    if a == 0.0:
        pre_result.append(-c/b)
    elif D == 0.0:
        pre_root = -b / (2.0*a)
        pre_result.append(pre_root)
    elif D > 0.0:
        sqD = math.sqrt(D)
        pre_root1 = (-b + sqD) / (2.0*a)
        pre_root2 = (-b - sqD) / (2.0*a)
        pre_result.append(pre_root1)
        pre_result.append(pre_root2)
    for pre_root in pre_result:
        if pre_root > 0:
            result.append(math.sqrt(pre_root))
            result.append(-math.sqrt(pre_root))
        elif pre_root == 0:
            result.append(pre_root)

    return result

def main():
    a = get_coef(1, 'Введите коэффициент А:')
    b = get_coef(2, 'Введите коэффициент В:')
    c = get_coef(3, 'Введите коэффициент С:')

    roots = get_roots(a,b,c)

    len_roots = len(roots)
    if len_roots == 0:
        print('Нет корней')
    elif len_roots == 1:
        print('Один корень: {}'.format(roots[0]))
    elif len_roots == 2:
        print('Два корня: {} и {}'.format(roots[0], roots[1]))
    elif len_roots == 3:
        print('Три корня: {}, {} и {}'.format(roots[0], roots[1], roots[2]))
    elif len_roots == 4:
        print('Четыре корня: {}, {}, {}, и {}'.format(roots[0], roots[1],
roots[2], roots[3]))

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

Результат выполнения программы:

```
Введите коэффициент A:
1
Введите коэффициент B:
0
Введите коэффициент C:
-4
Два корня: 1.4142135623730951 и -1.4142135623730951
```

Файл lab_oop.py

```
import sys
import math

class BiSquare:
    def __init__(self):
        self.coef_A = 0.0
        self.coef_B = 0.0
        self.coef_C = 0.0
        self.num_roots = 0
        self.roots_list = []

    def get_coef(self, index, prompt):
        try:
            coef_str = sys.argv[index]
        except:
            print(prompt)
            coef_str = input()
        coef = float(coef_str)
        return coef

    def get_coefs(self):
        self.coef_A = self.get_coef(1, 'Введите коэффициент A:')
        self.coef_B = self.get_coef(2, 'Введите коэффициент B:')
        self.coef_C = self.get_coef(3, 'Введите коэффициент C:')

    def calculate_roots(self):
        pre_roots = []
        a = self.coef_A
        b = self.coef_B
        c = self.coef_C
        D = b*b - 4*a*c
        if a == 0.0:
            pre_roots.append(-c/b)
        elif D == 0.0:
            pre_roots.append(-b / (2.0 * a))
        elif D > 0.0:
            sqD = math.sqrt(D)
            pre_roots.append((-b + sqD) / (2.0*a))
            pre_roots.append((-b - sqD) / (2.0*a))
        for pre_root in pre_roots:
            if pre_root == 0:
                self.num_roots += 1
                self.roots_list.append(pre_root)
            elif pre_root > 0:
                self.num_roots += 2
                self.roots_list.append(math.sqrt(pre_root))
                self.roots_list.append(-math.sqrt(pre_root))
```

```

def print_roots(self):
    if self.num_roots != len(self.roots_list):
        print(('Ошибка. Уравнение содержит {} действительных корней, ' +\
            'но было вычислено {} корней.').format(self.num_roots,
len(self.roots_list)))
    else:
        if self.num_roots == 0:
            print('Нет корней')
        elif self.num_roots == 1:
            print('Один корень: {}'.format(self.roots_list[0]))
        elif self.num_roots == 2:
            print('Два корня: {} и {}'.format(self.roots_list[0],
self.roots_list[1]))
        elif self.num_roots == 3:
            print('Три корня: {}, {} и {}'.format(self.roots_list[0],
self.roots_list[1], self.roots_list[2]))
        elif self.num_roots == 4:
            print('Четыре корня: {}, {}, {}, и {}'.format(self.roots_list[0],
self.roots_list[1], self.roots_list[2], self.roots_list[3]))

def main():
    r = BiSquare()
    r.get_coefs()
    r.calculate_roots()
    r.print_roots()

if __name__ == "__main__":
    main()

```

Результат выполнения программы:

```

Введите коэффициент A:
1
Введите коэффициент B:
0
Введите коэффициент C:
-4
Два корня: 1.4142135623730951 и -1.4142135623730951

```