

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков
программирования»

Отчет по лабораторной работе №1

Выполнил:

Студент ИУ5-35Б

Аннакулиева Д. Д.

Подпись и дата:

Проверил:

Преподаватель

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2023 г.

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 ([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.

Код

1_1:

```
import math
```

```
import sys
```

```
def input_num(index, prompt):
```

```
    try:
```

```
        coef_str = sys.argv[index]
```

```
    except:
```

```
        print(prompt)
```

```
        coef_str = input()
```

```
    while True:
```

```
        try:
```

```
            coef = float(coef_str)
```

```
            break
```

```
        except:
```

```
            print('Ошибка')
```

```
            coef_str = input()
```

```
    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 = input_num(1, 'Введите A')
    b = input_num(2, 'Введите B')
    c = input_num(3, 'Введите C')
    roots = get_roots(a, b, c)
    if len(roots) == 0:
        print('Нет действительных 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()
```

1_2

```
import sys
import math
```

```
class Equation:
```

```
    def __init__(self):
        self.A = 0.0
        self.B = 0.0
        self.C = 0.0
        self.solutions = []
```

```
    def get_koeff(self, index):
```

```
        if index == 1:
            letter = 'A'
        elif index == 2:
            letter = 'B'
        else:
            letter = 'C'
```

```
        try:
```

```
            koeff_str = sys.argv[index]
```

```
        except:
```

```
            while True:
```

```
                print('Введите коэффициент {}: '.format(letter))
```

```
                koeff_str = input()
```

```
                try:
```

```
                    koeff = float(koeff_str)
```

```
                    return koeff
```

```
                except ValueError:
```

```
                    print('Ошибка\n')
```

```
    def get_koeffs(self):
```

```
        self.A = self.get_koeff(1)
```

```
        self.B = self.get_koeff(2)
```

```
        self.C = self.get_koeff(3)
```

```
    def get_roots(self):
```

```
        a = self.A
```

```
        b = self.B
```

```
        c = self.C
```

```
        result = []
```

```
D = b*b - 4 * a * c
```

```
if D == 0.0:
```

```
    root = math.sqrt(-b / (2.0 * a))
```

```
    if (root == 0.0):
```

```
        result.append(abs(root))
```

```
    else:
```

```
        result.append(root)
```

```
        result.append(-root)
```

```
elif D > 0.0:
```

```
    sqD = math.sqrt(D)
```

```
    r1 = (-b + sqD) / (2.0 * a)
```

```
    r2 = (-b - sqD) / (2.0 * a)
```

```
    if r1 == 0.0:
```

```
        result.append(r1)
```

```
    if r2 == 0.0 and r1 != 0.0:
```

```
        result.append(r2)
```

```
    if r1 > 0.0:
```

```
        root1 = math.sqrt(r1)
```

```
        result.append(root1)
```

```
        result.append(-root1)
```

```
    if r2 > 0.0:
```

```
        root2 = math.sqrt(r2)
```

```
        result.append(root2)
```

```
        result.append(-root2)
```

```
return result
```

```
def main():
```

```
    eq = Equation()
```

```
    eq.get_koeffs()
```

```
    roots = eq.get_roots()
```

```
    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()
```