

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

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

Отчет по лабораторной работе №1
по курсу «Разработка интернет-приложений»

Выполнила:
студент группы ИУ5-54Б
Подопригорова Н. С.

Проверил:
Доцент кафедры ИУ5
Гапанюк Ю. Э.

Москва, 2020 г.

1. Описание задания

- 1.1. Разработать программу для решения биквадратного уравнения.
- 1.2. Программа должна быть разработана в виде консольного приложения на языке Python.
- 1.3. Программа осуществляет ввод с клавиатуры коэффициентов A , B , C , вычисляет дискриминант и корни уравнения (в зависимости от дискриминанта).
- 1.4. Если коэффициент A , B , C введен некорректно, то необходимо проигнорировать некорректное значение и ввести коэффициент повторно.
- 1.5. Первой строкой программа выводит ФИО разработчика и номер группы.
- 1.6. ДОПОЛНИТЕЛЬНОЕ ТРЕБОВАНИЕ. Коэффициенты A , B , C задаются в виде параметров командной строки. Если они не указаны, то вводятся с клавиатуры в соответствии с пунктом 2. Проверка из пункта 3 в этом случае производится для параметров командной строки без повторного ввода с клавиатуры.

2. Текст программы

```
import argparse
import math

class Equation():
    pass

class LinEquation(Equation):
    def __init__(self, a, b):
        self.a, self.b = a, b

    def calc_root(self):
        return -(self.b/self.a)

    def solution(self):
        if self.a == self.b == 0:
            print("Корней бесконечно много")
        elif self.a == 0:
            return []
        else:
            return [self.calc_root()]

class QuEquation(Equation):
    def __init__(self, a, b, c):
        self.a, self.b, self.c = a, b, c
        self.discr = -1

    def calc_discr(self):
```

```
self.dscr = self.b ** 2 - 4 * self.a * self.c
return self.dscr
```

```
def calc_root(self, d):
    return (-self.b + d) / (2 * self.a)
```

```
def solution(self):
    if self.a == 0:
        return LinEquation(self.b, self.c).solution()
```

```
    self.calc_dscr()
    if self.dscr < 0:
        return []
    elif self.dscr == 0:
        return [self.calc_root(self.dscr)]
    else:
        return [self.calc_root(math.sqrt(self.dscr)), self.calc_root(-
math.sqrt(self.dscr))]
```

```
class BiQuEquation(QuEquation):
```

```
    def __init__(self, a, b, c):
        super().__init__(a, b, c)
```

```
    def calc_dscr(self):
        if self.a == 0:
            return QuEquation(self.b, 0, self.c).calc_dscr()
        return super().calc_dscr()
```

```
    def calc_bi_root(self, qu_root):
        if qu_root < 0:
            return []
        elif qu_root == 0:
            return [qu_root]
        elif qu_root > 0:
            root = math.sqrt(qu_root)
            return [root, -root]
```

```
    def solution(self):
        if self.a == 0:
            return QuEquation(self.b, 0, self.c).solution()
        qu_roots = super().solution()
        answer = []
        for qu_root in qu_roots:
            answer += self.calc_bi_root(qu_root)
        return answer
```

```

def parse():
    parser = argparse.ArgumentParser()
    parser.add_argument('--a', help = 'first coefficient')
    parser.add_argument('--b', help = 'second coefficient')
    parser.add_argument('--c', help = 'third coefficient')
    return parser.parse_args()

def put_coef(args):
    print('Введите 3 коэффициента биквадратного уравнения')
    args.a = input('a = ')
    args.b = input('b = ')
    args.c = input('c = ')

def main():
    print('Nonpenguin IU5-54')
    args = parse()

    if not (args.a and args.b and args.c):
        put_coef(args)
    while True:
        try:
            args.a = float(args.a)
            args.b = float(args.b)
            args.c = float(args.c)
            break
        except:
            print('Не удаётся привести коэффициенты уравнения к типу float, попробуйте снова')
            put_coef(args)

    eq = BiQuEquation(args.a, args.b, args.c)
    if not (args.a == 0 and args.b == 0):
        print('Дискриминант: {d}'.format(d=eq.calc_discr()))
    answer = eq.solution()
    if answer == []:
        print('Действительных корней нет')
    elif answer is None:
        pass
    else:
        print('Корни: ', answer)

if __name__ == '__main__':
    main()

```

3. Экранные формы с примерами выполнения программы

```
(env) ~/my/pythonchic/rip/lab1 ➤ python3 solution.py --a 1 --b -40 --c 4
Подопригорова Н. ИУ5-54
Дискриминант: 1584.0
Корни: [6.3166247903554, -6.3166247903554, 0.3166247903554002, -0.3166247903554002]
(env) ~/my/pythonchic/rip/lab1 ➤ python3 solution.py --a 1 --b 5 --c -36
Подопригорова Н. ИУ5-54
Дискриминант: 169.0
Корни: [2.0, -2.0]
(env) ~/my/pythonchic/rip/lab1 ➤ python3 solution.py --a 0 --b 0 --c 8
Подопригорова Н. ИУ5-54
Действительных корней нет
(env) ~/my/pythonchic/rip/lab1 ➤ python3 solution.py --a 2 --b 50 --c 4
Подопригорова Н. ИУ5-54
Дискриминант: 2468.0
Действительных корней нет
(env) ~/my/pythonchic/rip/lab1 ➤ python3 solution.py --a 0 --b 1 --c -36
Подопригорова Н. ИУ5-54
Дискриминант: 144.0
Корни: [6.0, -6.0]
```