

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

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

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №1
«Основные конструкции языка Python»

Выполнил:
студент группы ИУ5-32Б:
Бекетов Роман Александрович
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2022 г.

Задание

Разработать программу для решения биквадратного уравнения.

Программа должна быть разработана в виде консольного приложения на языке Python.

Программа осуществляет ввод с клавиатуры коэффициентов A , B , C , вычисляет дискриминант и действительные корни уравнения (в зависимости от дискриминанта).

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.

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.

Листинг

```
import sys
import math

class BisquareRoots:

    def __init__(self):
        """
        Конструктор класса
        """
        # Объявление коэффициентов
        self.coef_A = 0.0
        self.coef_B = 0.0
        self.coef_C = 0.0
        # Количество корней
        self.num_roots = 0
        # Список корней
        self.roots_list = []

    def get_coef(self, index, prompt):
        """
        Читаем коэффициент из командной строки или вводим с клавиатуры

        Args:
            index (int): Номер параметра в командной строке
            prompt (str): Приглашение для ввода коэффициента

        Returns:
            float: Коэффициент квадратного уравнения
        """
```

```

while True:
    # Пробуем прочитать коэффициент из командной строки
    try:
        coef = float(sys.argv[index])
        if coef == 0.0 and index == 1:
            raise UserWarning
        break
    except Exception as e:
        # Вводим с клавиатуры
        try:
            if (e.__class__ == ValueError):
                print('Некорректный ввод.')
            if (e.__class__ == UserWarning):
                print('Введённое значение параметра A некорректно');
            print(prompt)
            coef = float(input())
            if (index == 1 and coef == 0.0):
                raise UserWarning
            break
        except ValueError:
            print('Некорректный ввод.')
        except UserWarning:
            print('Введённое значение параметра A некорректно');
        except:
            print('Что-то пошло не так...')
    return coef

def get_coefs(self):
    """
    Чтение трех коэффициентов
    """
    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):
    """
    Вычисление корней квадратного уравнения
    """
    a = self.coef_A
    b = self.coef_B
    c = self.coef_C
    # Вычисление дискриминанта и корней
    D = b*b - 4*a*c
    if D == 0.0:
        root = -b / (2.0*a)
        if root > 0.0:
            self.num_roots = 2
            self.roots_list.append(math.sqrt(root))
            self.roots_list.append(-math.sqrt(root))
        elif root == 0:

```

```

        self.num_roots = 1
        self.roots_list.append(root)
elif D > 0.0:
    sqD = math.sqrt(D)
    root1 = (-b + sqD) / (2.0*a)
    root2 = (-b - sqD) / (2.0*a)
    if root1 > 0.0:
        self.num_roots += 2
        self.roots_list.append(math.sqrt(root1))
        self.roots_list.append(-math.sqrt(root1))
    elif root1 == 0.0:
        self.num_roots += 1
        self.roots_list.append(root1)
    if root2 > 0.0:
        self.num_roots += 2
        self.roots_list.append(math.sqrt(root2))
        self.roots_list.append(-math.sqrt(root2))
    elif root2 == 0.0:
        self.num_roots += 1
        self.roots_list.append(root2)

def print_roots(self):
    # Проверка отсутствия ошибок при вычислении корней
    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:
        match self.num_roots:
            case 0:
                print('Нет корней')
            case 1:
                print('Один корень: {}'.format(self.roots_list[0]))
            case 2:
                print('Два корня: {} и {}'.format(self.roots_list[0],
self.roots_list[1]))
            case 3:
                print('Три корня: {}, {} и {}'.format(self.roots_list[0],
self.roots_list[1], self.roots_list[2]))
            case 4:
                print('Четыре корня: {}, {}, {} и {}'.format(self.roots_list[0],
self.roots_list[1], self.roots_list[2], self.roots_list[3]))
            case _:
                print('Ошибка выполнения.')

def main():
    """
    Основная функция
    """
    # Создание объекта класса

```

```

r = BisquareRoots()
# Последовательный вызов необходимых методов
r.get_coefs()
r.calculate_roots()
r.print_roots()

# Если сценарий запущен из командной строки
if __name__ == "__main__":
    main()

```

Тесты

```

romanbeketov@MacBook-Air-Roman biquadratic % python3 qr.py 1 -13 36
Четыре корня: 3.0, -3.0, 2.0 и -2.0

```

```

romanbeketov@MacBook-Air-Roman biquadratic % python3 qr.py 1 error 36
Некорректный ввод.
Введите коэффициент B:
-13
Четыре корня: 3.0, -3.0, 2.0 и -2.0

```

```

romanbeketov@MacBook-Air-Roman biquadratic % python3 qr.py
Введите коэффициент A:
dwe
Некорректный ввод.
Введите коэффициент A:
=1
Некорректный ввод.
Введите коэффициент A:
1
Введите коэффициент B:
-2
Введите коэффициент C:
1
Два корня: 1.0 и -1.0

```

```

romanbeketov@MacBook-Air-Roman biquadratic % python3 qr.py
Введите коэффициент A:
1
Введите коэффициент B:
-2
Введите коэффициент C:
1
Два корня: 1.0 и -1.0

```

```
romanbeketov@MacBook-Air-Roman biquadratic % python3 qr.py 53 3 53
Нет корней
```