

**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  
«Изучение основных конструкций языка Python»

Выполнил:  
студент группы ИУ5-33Б  
Нагапетян Валерий

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю. Е.

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.

# Текст программы

```
import sys
import math

class SquareRoots:
    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 ,биквадратного уравнения
        """

        while True:
            try:
                # Пробуем прочитать коэффициент из командной строки
                coef_str = sys.argv[index]

            except Exception:
                # Вводим с клавиатуры
                print(prompt)
                coef_str = input()
            try:
```

```

        # Переводим строку в действительное число
        coef = float(coef_str)
        return coef
    except ValueError:
        print("Ошибка! Коэффициенты уравнение должны быть числового
типа")

def get_coefs(self):
    """Чтение трех коэффициентов"""

    self.coef_A = self.get_coef(1, "Введите коэффициент A:")
    while self.coef_A == 0:
        self.coef_A = self.get_coef(
            1,
            "Коэффициент а не может быть равен нулю! Иначе уравнение будет
квадратным\nВведите 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 биквадратного уравнения"""

    a = self.coef_A
    b = self.coef_B
    c = self.coef_C
    # Вычисление дискриминанта и корней
    bi_root_list = []
    D = b * b - 4 * a * c

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

```

```
self.processing_roots(bi_root_list)
```

```
def processing_roots(self, roots):
```

```
    for bi_root in roots:
```

```
        if bi_root > 0:
```

```
            self.roots_list.append(math.sqrt(bi_root))
```

```
            self.roots_list.append(-math.sqrt(bi_root))
```

```
            self.num_roots += 2
```

```
        elif bi_root == 0:
```

```
            self.roots_list.append(0)
```

```
            self.num_roots += 1
```

```
def print_roots(self):
```

```
    # Проверка отсутствия ошибок при вычислении корней
```

```
    assert self.num_roots == len(
```

```
        self.roots_list
```

```
    ), f'Ошибка! Уравнение содержит {self.num_roots} \
```

```
    действительных корней, но было вычислено {len(self.roots_list)} корней."
```

```
    if self.num_roots == 0:
```

```
        print("Нет корней")
```

```
    else:
```

```
        print("Корни биквадратного уравнения:")
```

```
        counter = 1
```

```
        for root in self.roots_list:
```

```
            print(f"x{counter} = {root}", end="; ")
```

```
            counter += 1
```

```
def main():
```

```
    """Основная функция"""
```

```
    # Создание объекта класса
```

```
    r = SquareRoots()
```

```
    # Последовательный вызов необходимых методов
```

```
    r.get_coefs()
```

```
    r.calculate_roots()
```

```
    r.print_roots()
```

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

# Результаты

```
(www_venv) PS C:\Users\valer\PycharmProjects\PPSH\00P and syntax> python lab1.py
Введите коэффициент A:
1
Введите коэффициент B:
-10
Введите коэффициент C:
9
Корни биквадратного уравнения:
x1 = 3.0; x2 = -3.0; x3 = 1.0; x4 = -1.0;
(www_venv) PS C:\Users\valer\PycharmProjects\PPSH\00P and syntax> python lab1.py 1 -10 9
Корни биквадратного уравнения:
x1 = 3.0; x2 = -3.0; x3 = 1.0; x4 = -1.0;
(www_venv) PS C:\Users\valer\PycharmProjects\PPSH\00P and syntax> python lab1.py 1 -10
Введите коэффициент C:
9
Корни биквадратного уравнения:
x1 = 3.0; x2 = -3.0; x3 = 1.0; x4 = -1.0;
```