

**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
«Основные конструкции языка Python»**

Выполнила:
Студентка группы ИУ5-33Б
Беспалова Виктория
Подпись и дата:

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

2023 год

Описание задания

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 А, В, С, вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С 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 А, В, С 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.
6. Дополнительное задание 2 (*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

Текст программы

```
# Copyright 2023 Victoria Besspalova <vickycat2014@gmail.com>

import sys
import math

def get_coef(index, prompt) -> float:
    """
    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
    """
    try:
        # Пробуем прочитать коэффициент из командной строки
        try:
            coef_str = sys.argv[index]
            # Переводим строку в действительное число
            coef = float(coef_str)
            return coef
        except ValueError:
            # Некорректное значение коэффициента, повторяем ввод
            pass
    except:
        # Вводим с клавиатуры
        while True:
```

```

        try:
            print(prompt)
            coef_str = input()
            # Переводим строку в действительное число
            coef = float(coef_str)
            return coef
        except ValueError:
            # Некорректное значение коэффициента, повторяем ввод
            pass

def get_quadratic_equation_roots(a, b, c) -> tuple:
    '''
    Вычисление корней квадратного уравнения

    Args:
        a (float): коэффициент A
        b (float): коэффициент B
        c (float): коэффициент C

    Returns:
        tuple(float): Кортеж корней
    '''
    result = ()
    D = b * b - 4 * a * c
    if D == 0.0:
        root = -b / (2.0 * a)
        result += (root,)
    elif D > 0.0:
        sqD = math.sqrt(D)
        root1 = (-b + sqD) / (2.0 * a)
        root2 = (-b - sqD) / (2.0 * a)
        result += (root1, root2)

    return result

def get_biquadratic_equation_roots(a, b, c) -> tuple:
    # Вычисление корней квадратного уравнения
    roots = get_quadratic_equation_roots(a, b, c)
    result = ()
    for root in roots:
        if root > 0.0:
            result += (math.sqrt(root),)
            result += (-1 * math.sqrt(root),)
        elif root == 0.0:
            result += (root,)

    return result

def main():
    '''
    Основная функция
    '''
    a = get_coef(1, 'Введите коэффициент A:')
    b = get_coef(2, 'Введите коэффициент B:')
    c = get_coef(3, 'Введите коэффициент C:')

    roots = get_biquadratic_equation_roots(a, b, c)
    # Вывод корней
    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]))
else:
    print('Четыре корня: {}, {}, {} и {}'.format(roots[0], roots[1],
roots[2], roots[3]))

# Если сценарий запущен из командной строки
if __name__ == "__main__":
    print('_____Лабораторная работа №1_____')
    main()

```

Примеры выполнения программы

```

/Users/victoria/PycharmProjects/lab01/venv/bin/python /Users/victoria/PycharmProjects/lab01/main.py
_____Лабораторная работа №1_____
Введите коэффициент A:
3
Введите коэффициент B:
-10
Введите коэффициент B:
-
Введите коэффициент B:
-17
Введите коэффициент C:
16
Четыре корня: 4.0, -4.0, 1.0 и -1.0

Process finished with exit code 0

```

```

/Users/victoria/PycharmProjects/lab01/venv/bin/python /Users/victoria/PycharmProjects/lab01/main.py
_____Лабораторная работа №1_____
Введите коэффициент A:
3
Введите коэффициент B:
3
Введите коэффициент C:
10
Два корня: 1.4142135623730951 и -1.4142135623730951

Process finished with exit code 0

```

```

victoria@MacBook-Pro-victoria ~ % cd /Users/victoria/Downloads
victoria@MacBook-Pro-victoria Downloads % python3 main.py 1 -1 4
_____Лабораторная работа №1_____
Нет корней
victoria@MacBook-Pro-victoria Downloads % python3 main.py 1 -5 6
_____Лабораторная работа №1_____
Четыре корня: 1.7320508075688772, -1.7320508075688772, 1.4142135623730951 и -1.4142135623730951
victoria@MacBook-Pro-victoria Downloads % python3 main.py -4 16 0
_____Лабораторная работа №1_____
Три корня: -0.0, 2.0 и -2.0
victoria@MacBook-Pro-victoria Downloads % python3 main.py 1 0 10
_____Лабораторная работа №1_____
Нет корней
victoria@MacBook-Pro-victoria Downloads % python3 main.py 1 0 0
_____Лабораторная работа №1_____
Один корень: -0.0
victoria@MacBook-Pro-victoria Downloads %

```