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»

Курс «Базовые компоненты интернет-технологий»

Лабораторная работа №1

«Основные конструкции языка Python»

Выполнил:
студент группы ИУ5-32Б
Долинский А.А.

Проверил:
Канев А.И.

2021 г.

Цель лабораторной работы:

Изучение основных конструкций языка Python.

Постановка задачи:

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 кода:

```
import sys
import math


def get_coef(index, prompt):
    '''
    Читаем коэффициент из командной строки или вводим с клавиатуры
    Args:
        index (int): Номер параметра в командной строке
        prompt (str): Приглашение для ввода коэффицента
    Returns:
        float: Коэффициент квадратного уравнения
    '''
    r,k,etalon =0,0, "-0123456789."
    try:
        # Пробуем прочитать коэффициент из командной строки
        coef_str = sys.argv[index]
    except:
        # Вводим с клавиатуры
        print(prompt)
        coef_str = input()
        while True:
            for i in coef_str:
                if i in etalon:
                    if i == "-" or i == ".":
                        r+=1
                    else:
                        k+=1
                else:
                    break
```

```
            if k != 0 and k+r == len(coef_str):
                coef = float(coef_str)
                break
            else:
                coef_str = input()
                r, k = 0,0
    return coef

def get_roots(a, b, c):
    '''
    Вычисление корней квадратного уравнения
    Args:
        a (float): коэффициент A
        b (float): коэффициент B
        c (float): коэффициент C
    Returns:
        list[float]: Список корней
    '''
    result = []
    D = b * b - 4 * a * c
    if a == b == c == 0:
        result = "12345"
    elif a == b == 0:
        result = []
    elif a == 0:
        if c/b <= 0:
            if math.sqrt(abs(c/b)) == -math.sqrt(abs(c/b)):
                result.append(math.sqrt(abs(c/b)))
            else:
                result.append(math.sqrt(abs(c/b)))
                result.append(-math.sqrt(abs(c/b)))

    else:
        if D == 0.0:
            if (-b / (2.0 * a)) >= 0:
                if math.sqrt(-b / (2.0 * a)) == -math.sqrt(-b / (2.0 * a)):
                    result.append(math.sqrt(-b / (2.0 * a)))
                else:
                    result.append(math.sqrt(-b / (2.0 * a)))
                    result.append(-math.sqrt(-b / (2.0 * a)))
        elif D > 0.0:
            sqD = math.sqrt(D)
            root1 = (-b + sqD) / (2.0 * a)
            root2 = (-b - sqD) / (2.0 * a)
            if root1 == root2:
                result.append(math.sqrt(root1))
                result.append(-math.sqrt(root1))
            else:
                if root1 >= 0:
                    if math.sqrt(root1) == -math.sqrt(root1):
                        result.append(math.sqrt(root1))
                    else:
                        result.append(math.sqrt(root1))
                        result.append(-math.sqrt(root1))

                if root2 >= 0:
                    if math.sqrt(root2) == -math.sqrt(root2):
                        result.append(math.sqrt(root2))
                    else:
                        result.append(math.sqrt(root2))
                        result.append(-math.sqrt(root2))
    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:')
    # Вычисление корней
    roots = get_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]))
    elif len_roots == 4:
        print('Четыре корня: {}, {}, {} и {}'.format(roots[0], roots[1],
roots[2], roots[3]))
    elif len_roots == 5:
        print("Бесконечное количество корней")


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

Тестирование:

Ввод/ожидание:	Вывод:
0 0 0 - Бесконечное число корней	Бесконечное число корней
0 0 1 – Нет корней	Нет корней
1 0 -4 - Два корня: 1.4142135623730951 и -1.4142135623730951	Два корня: 1.4142135623730951 и -1.4142135623730951
1 -4 0 - Три корня: 2.0, -2.0 и 0	Три корня: 2.0, -2.0 и 0
1 1 1 - Нет корней	Нет корней
0 1 -4 - Два корня: 2.0 и -2.0	Два корня: 2.0 и -2.0

Вывод:

Благодаря данной лабораторной работе я познакомился с базовым синтаксисом языка Python , разобрался с базовыми функциями , узнал и применил конструкцию try – except .