

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)

**Отчет по лабораторной работе № 1 по курсу
Базовые компоненты интернет-технологий
«Основные конструкции языка Python»**

ПРЕПОДАВАТЕЛЬ:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

(подпись)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-34Б

Верин Д.С.

(подпись)

" " _____ 2021 г.

1. 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.

2. Текст программы.

Файл qr.py:

```
import sys
import math

def is_real(string):
    try:
        float(string)
        return True
    except (ValueError, TypeError):
        return False

def check(index, prompt, coef_str):
    while True:
        if not is_real(coef_str):
            print(f"{index} коэффициент введен неверно, невозможно "
                  f"преобразовать в вещественное число\n{prompt} ", end = '')
            coef_str = input()
        else:
            break
    return coef_str

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 коэффициента
    Returns:
        float: Коэффициент квадратного уравнения
    """
    try:
        # Пробуем прочитать коэффициент из командной строки
        coef_str = check(index, prompt, sys.argv[index])
    except:
        # Вводим с клавиатуры
        print(prompt, end = ' ')
        coef_str = check(index, prompt, input())
    # Переводим строку в действительное число
    coef = float(coef_str)
    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
    ...
if a == 0 and b == 0 and c == 0:
    result = [1, 2, 3, 4, 5]
elif a > 0 and b > 0 and c > 0 or a < 0 and b < 0 and c < 0:
    result = []
elif a == 0 or b == 0:
    if c > 0:
        result = []
    elif c == 0:
        result = [0,]
    else:
        if a == 0:
            result = [-math.sqrt(-c/b), math.sqrt(-c/b)]
        else:
            result = [-math.sqrt(math.sqrt(-c/a)), math.sqrt(math.sqrt(-
c/a))]]
elif c == 0:
    if a > 0 and b > 0:
        result = [0,]
    else:
        result = [-math.sqrt(abs(b/a)), 0, math.sqrt(abs(b/a))]
else:
    D = b * b - 4 * a * c
    if D < 0:
        result = []
    elif D == 0:
        result = [-math.sqrt(-b / 2 / a), math.sqrt(-b / 2 / a)]
    else:
        D = math.sqrt(D)
        s1 = (-b + D) / 2 / a
        s2 = (-b - D) / 2 / a
        if s1 < 0 and s2 < 0:
            result = []
        elif (s1 > 0 and s2 < 0) or (s1 < 0 and s2 > 0):
            if s1 > 0:
                result = [-math.sqrt(s1), math.sqrt(s1)]
            else:
                result = [-math.sqrt(s2), math.sqrt(s2)]
        else:
            result = [-math.sqrt(s1), -math.sqrt(s2), math.sqrt(s2),
math.sqrt(s1)]
    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(f"Один корень: {roots[0]}")
elif len_roots == 2:
    print(f"Два корня: {roots[0]} и {roots[1]}")
elif len_roots == 3:
    print(f"Три корня: {roots[0]}, {roots[1]} и {roots[2]}")
elif len_roots == 4:
    print(f"Четыре корня: {roots[0]}, {roots[1]}, {roots[2]} и {roots[3]}")
else:
    print("Бесконечное множество корней")

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

```

Пример выполнения программы

```
D:\python_projects\lab1>python qr.py 0 0 0
Бесконечное множество корней

D:\python_projects\lab1>python qr.py 0 7 0
Один корень: 0

D:\python_projects\lab1>python qr.py 0 1 -25
Два корня: -5.0 и 5.0

D:\python_projects\lab1>python qr.py 10 -15 0
Три корня: -1.224744871391589, 0 и 1.224744871391589

D:\python_projects\lab1>python qr.py 1 -5 4
Четыре корня: -2.0, -1.0, 1.0 и 2.0

D:\python_projects\lab1>python qr.py a -5 -36
1 коэффициент введен неверно, невозможно преобразовать в вещественное число
Введите коэффициент A: 1
Два корня: -3.0 и 3.0

D:\python_projects\lab1>python qr.py a -5 c
1 коэффициент введен неверно, невозможно преобразовать в вещественное число
Введите коэффициент A: 1
3 коэффициент введен неверно, невозможно преобразовать в вещественное число
Введите коэффициент C: 6
Четыре корня: -1.7320508075688772, -1.4142135623730951, 1.4142135623730951 и 1.7320508075688772

D:\python_projects\lab1>python qr.py
Введите коэффициент A: 0.2
Введите коэффициент B: -1
Введите коэффициент C: -7.2
Два корня: -3.0 и 3.0
```