Московский государственный технический университет имени Н.Э.Баумана

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет РТ Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по лабораторной работе № 1 по курсу**

**Базовые компоненты интернет-технологий**

#### **"Основные конструкции языка Python"**

|  |  |  |
| --- | --- | --- |
| Исполнитель |  |  |
| студент группы РТ5-31б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Титов А.Д. |
|  |  | “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 г. |
|  |  |  |
| Проверил |  |  |
| Доцент кафедры ИУ5 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гапанюк Ю.Е. |
|  |  | “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 г. |

Москва -  2021

**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.

**Текст программы**

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

'''

try:

# Пробуем прочитать коэффициент из командной строки

coef\_str = sys.argv[index]

coef = float(coef\_str)

except:

# Вводим с клавиатуры

coef\_str = input(prompt)

try:

# Переводим строку в действительное число

coef = float(coef\_str)

except:

# В случае невозможности перевода делаем рекурсивный алгоритм

return get\_coef(index, prompt)

return coef

def get\_roots(a, b, c):

'''

Вычисление корней квадратного уравнения

Args:

a (float): коэффициент А

b (float): коэффициент B

c (float): коэффициент C

Returns:

list[float]: Список корней

'''

result = []

D = b\*b - 4\*a\*c

if D == 0.0:

root = -b / (2.0\*a)

result.append(root)

elif D > 0.0:

sqD = math.sqrt(D)

root1 = (-b + sqD) / (2.0\*a)

root2 = (-b - sqD) / (2.0\*a)

result.append(root1)

result.append(root2)

return result

def get\_roots\_b(roots):

'''

Вычисление корней для биквадратного уравнения

Args:

list [float]: массив корней квадратного уравнения

Returns:

list [float]: массив корней биквадратного уравнения

'''

result = []

for root in roots:

if root == 0:

result.append(root)

elif root > 0:

sqRoot = math.sqrt(root)

result.append(sqRoot)

result.append(-sqRoot)

return result

def main():

'''

Основная функция

'''

a = get\_coef(1, 'Введите коэффициент А: ')

b = get\_coef(2, 'Введите коэффициент B: ')

c = get\_coef(3, 'Введите коэффициент C: ')

# Вычисление корней

roots = get\_roots(a,b,c)

# Вычисление корней для биквадратного уравнения

roots = get\_roots\_b(roots)

# 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]))

# Если сценарий запущен из командной строки

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Примеры запуска

# qr.py 1 11 10 - нет корней

# qr.py 1 16 0 - {0}

# qr.py 1 -2 -8 - {2, -2}

# qr.py 1 -4 0 - {2, -2, 0}

# qr.py 1 -10 9 - {3, -3, 1, -1}

**Примеры выполнения программы**

