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 |  |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Гапанюк Ю. Е. | (подпись) |
|  |  |
| ИСПОЛНИТЕЛЬ: |  |
| студентка группы ИУ5-35Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Гурова М.Д. | (подпись) |
|  | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

Москва - 2021

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

Задание:

Разработать программу для решения [биквадратного уравнения.](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%87%D0%B5%D1%82%D0%B2%D1%91%D1%80%D1%82%D0%BE%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8#%D0%91%D0%B8%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5)

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](https://github.com/ugapanyuk/BKIT_2021/blob/main/code/lab1_code) ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки.](https://realpython.com/python-command-line-arguments/#the-command-line-interface)
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

choice =10

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]

except:

# Вводим с клавиатуры

print(prompt)

try:

choice = float(raw\_input(""))

if choice == 0 :

print "Вы ввели 0 -> ввод будет начат заново"

main()

except ValueError:

print "Вы ввели неправильный символ -> ввод будет начат заново"

main()

return choice

def get\_coef1(index, prompt):

try:

# Пробуем прочитать коэффициент из командной строки

coef\_str = sys.argv[index]

except:

# Вводим с клавиатуры

print(prompt)

try:

choice = float(raw\_input(""))

except ValueError:

print "Вы ввели неправильный символ -> ввод будет начат заново"

main()

return choice

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)

sqr=root

root1=math.sqrt(sqr)

root2=-math.sqrt(sqr)

result.append(root1)

result.append(root2)

elif (D > 0 and c!=0 ):

sqD = math.sqrt(D)

sq1 = (-b + sqD) / (2.0\*a)

sq2 = (-b - sqD) / (2.0\*a)

if sq1>0:

root1=math.sqrt(sq1)

root2=-math.sqrt(sq1)

result.append(root1)

result.append(root2)

if sq2>0:

root3=math.sqrt(sq2)

root4=-math.sqrt(sq2)

result.append(root3)

result.append(root4)

elif (b<0 and c==0):

root3=0

root1=math.sqrt(abs(a\*b))

root2=-math.sqrt(abs(a\*b))

result.append(root1)

result.append(root2)

result.append(root3)

return result

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

'''

Дискриминант

'''

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

return D

def main():

'''

Основная функция

'''

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

b = get\_coef1(2, 'Введите коэффициент B:')

c = get\_coef1(3, 'Введите коэффициент C:')

# Вычисление корней

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

# Вывод корней

len\_roots = len(roots)

d=get\_ds(a,b,c)

print('D = {}'.format(d))

if len\_roots == 0:

print('Нет корней')

main()

elif len\_roots == 1:

print('Один корень: {}'.format(roots[0]))

elif len\_roots == 2:

if (roots[0]==0.0 and roots[1]==-0.0):

print('Один корень: {}'.format(0))

else :

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()

# Пример запуска

# qr.py 1 0 -4

Пример работы программы программы: