|  |  |
| --- | --- |
| Изображение выглядит как текст, керамические изделия, фарфор  Автоматически созданное описание | **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 «ГУИМЦ»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Базовые компоненты ИТ»

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

«Основные конструкции языка Python.»

Студент: Аброчнов Е.С., группа ИУ5Ц-53Б

Преподаватель: Гапанюк Ю.Е.

2021 г.

**1. 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.

**2. Листинг программы**

import sys

import math

def get\_coef(index, prompt):

try:

# Пробуем прочитать коэффициент из командной строки

coef\_str = sys.argv[index]

except:

# Вводим с клавиатуры

buf = False

while (buf != True):

print(prompt)

coef\_str = input()

try:

float(coef\_str)

buf = True

except ValueError:

buf = False

coef = float(coef\_str)

return coef

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

result = []

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

if D == 0.0:

root = -b / (2.0 \* a)

if root >= 0.0:

Root1 = -math.sqrt(root)

Root2 = math.sqrt(root)

result.append(Root1)

result.append(Root2)

elif root < 0.0:

return result

elif D < 0.0:

return result

elif D > 0.0:

sqD = math.sqrt(D)

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

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

if root1 >= 0.0:

if math.sqrt(root1) == 0.0:

Root1 = math.sqrt(root1)

result.append(Root1)

elif math.sqrt(root1) != 0.0:

Root1 = -math.sqrt(root1)

Root2 = math.sqrt(root1)

result.append(Root1)

result.append(Root2)

if root2 >= 0.0:

if math.sqrt(root2) == 0.0:

Root5 = math.sqrt(root2)

result.append(Root5)

elif math.sqrt(root2) != 0.0:

Root3 = -math.sqrt(root2)

Root4 = math.sqrt(root2)

result.append(Root3)

result.append(Root4)

return result

def main():

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

while a == 0.0:

print('a в биквадратном уравнении не может равняться нулю')

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)

# 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:

if (roots[0] == 0.0) or (roots[0] == -0.0):

print('Один корень: 0.0')

elif roots[0] != 0.0:

print('Два корня: {} и {}'.format(roots[0], roots[1]))

elif len\_roots == 3:

print('Три корня: {}, {}, {}'.format(roots[0], roots[1], roots[2]))

elif len\_roots == 4:

print('Четыре корня: {}, {}, {}, {}'.format(roots[0], roots[1], roots[2], roots[3]))

# Если сценарий запущен из командной строки

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

main()

**3. Экранные формы с примерами выполнения программы**

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание