МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”

A logo with a gold star and blue ribbon

Description automatically generated

Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по лабораторной работе №1  
“Основные конструкции языка Python.”

**Выполнил:**  
Студент группы ИУ5-34Б  
Верзаков Н.В.  
**Преподаватель:**  
Нардид А.Н.

Москва 2025

Задание лабораторной работы[​](https://iu5edu.ru/wiki/cpp2/docs/labs/lab6/#%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5)

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_2022/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.
5. Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.

Текст программы

biquadratic\_oop.py

import sys

import math

class SquareRoots:

def **\_\_init\_\_**(self):

self.coef\_A = 0.0

self.coef\_B = 0.0

self.coef\_C = 0.0

self.num\_roots = 0

self.roots\_list = []

def **get\_coef**(self, index, prompt):

try:

coef\_str = sys.argv[index]

except:

**print**(prompt)

coef\_str = **input**()

coef = float(coef\_str)

return coef

def **get\_coefs**(self):

self.coef\_A = self.**get\_coef**(1, 'Введите коэффициент А:')

self.coef\_B = self.**get\_coef**(2, 'Введите коэффициент B:')

self.coef\_C = self.**get\_coef**(3, 'Введите коэффициент C:')

def **calculate\_roots**(self):

a = self.coef\_A

b = self.coef\_B

c = self.coef\_C

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

roots = []

if D == 0.0:

root1 = -b / (2.0\*a)

if root1 > 0:

r = math.**sqrt**(root1)

roots.**append**(r)

roots.**append**(-r)

elif root1 == 0:

roots.**append**(0)

elif D > 0.0:

sqD = math.**sqrt**(D)

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

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

if root1 > 0:

r = math.**sqrt**(root1)

roots.**append**(r)

roots.**append**(-r)

elif root1 == 0:

roots.**append**(0)

if root2 > 0:

r = math.**sqrt**(root2)

roots.**append**(r)

roots.**append**(-r)

elif root2 == 0:

roots.**append**(0)

self.roots\_list = roots

self.num\_roots = **len**(roots)

def **print\_roots**(self):

if self.num\_roots != **len**(self.roots\_list):

**print**(('Ошибка. Уравнение содержит {} действительных корней, ' +\

'но было вычислено {} корней.').**format**(self.num\_roots, **len**(self.roots\_list)))

else:

if self.num\_roots == 0:

**print**('Нет корней')

elif self.num\_roots == 1:

**print**('Один корень: {}'.**format**(self.roots\_list[0]))

elif self.num\_roots == 2:

**print**('Два корня: {} и {}'.**format**(self.roots\_list[0], \

self.roots\_list[1]))

elif self.num\_roots == 3:

**print**('Три корня: {}, {}, {}'.**format**(self.roots\_list[0], \

self.roots\_list[1], self.roots\_list[2]))

elif self.num\_roots == 4:

**print**('Четыре корня: {}, {}, {}, {}'.**format**(self.roots\_list[0], \

self.roots\_list[1], self.roots\_list[2], self.roots\_list[3]))

def **main**():

r = SquareRoots()

r.**get\_coefs**()

r.**calculate\_roots**()

r.**print\_roots**()

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

**main**()

biquadratic\_proc.py

import sys

import math

def **get\_coef**(index, prompt):

try:

coef\_str = sys.argv[index]

except:

**print**(prompt)

coef\_str = **input**()

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:

result.**append**(root)

result.**append**(-root)

elif root == 0:

result.**append**(0)

elif D > 0.0:

sqD = math.**sqrt**(D)

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

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

if root1 > 0:

r = math.**sqrt**(root1)

result.**append**(r)

result.**append**(-r)

elif root1 == 0:

result.**append**(0)

if root2 > 0:

r = math.**sqrt**(root2)

result.**append**(r)

result.**append**(-r)

elif root2 == 0:

result.**append**(0)

return result

def **main**():

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

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

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

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

if not roots:

**print**('Нет действительных корней')

elif **len**(roots) == 1:

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

elif **len**(roots) == 2:

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

else:

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

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

**main**()

Экранные формы с примерами выполнения программы

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.