*Государственное образовательное учреждение высшего профессионального образования*



***«***

***Московский***

***государственный***

***технический***

***университет***

***имени***

***Н***

***.***

***Э***

***.***

***Баумана***

***»***

***(***

***МГТУ***

***им***

***.***

***Н***

***.***

***Э***

***.***

***Баумана***

***)***

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Системы обработки информации и управления» (ИУ-5)

**ЛАБОРАТОРНАЯ РАБОТА**

**№ 1**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Основные конструкции языка Python\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

Группа ИУ5-35Б

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **16.12.2024** /**Д.Е. Мушкарин /**

(Подпись, дата) (И.О.Фамилия)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** /**Ю. Е. Гапанюк/**

(Подпись, дата) (И.О.Фамилия)

2024

**Задание:**

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.
5. Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме С++).

**Текст программы (Python):**

**def.py**

import math

from typing import List

import sys

def SolveQuadratic(Equation: List) -> List:

a, b,c = map(float,Equation)

D = math.pow(b,2) - 4 \* a \* c

if (D < 0):

solutions = ()

if (D == 0):

solutions = -b / (2\*a)

if (D > 0):

sq = math.sqrt(D)

solutions = (-b + sq) / (2\*a), (-b - sq) / (2\*a)

return solutions

def initialization():

coef = []

arg = sys.argv

if (len(arg) > 1):

coef = arg[1:4]

else:

for i in ["A","B","C"]:

while (True):

try:

coef.append(float(input(f"Введите коэфициент {i}: ")))

break

except ValueError: print("Неверный коэфициент. Попробуйте снова.\n")

except: return -1

return coef

def main(\*args, \*\*kwargs):

coef = initialization()

quadratic\_solutions = SolveQuadratic(coef)

solutions = []

for i in quadratic\_solutions:

solutions.extend(SolveQuadratic([1,0,-i]))

print(\*solutions)

return 0

if (\_\_name\_\_ == "\_\_main\_\_"): main()

**Дополнительное задание 1**

**class.py**

import math

from typing import List

import sys

class Quadratic:

def \_\_init\_\_(self, \*args, \*\*kwargs) -> List[float]:

self.coef = []

if (len(args) == 4):

self.coef = args[1:4]

else:

for i in ["A","B","C"]:

while (True):

try:

self.coef.append(float(input(f"Введите коэфициент {i}: ")))

break

except ValueError: print("Неверный коэфициент. Попробуйте снова.\n")

def get(self):

return self.coef[0],self.coef[1],self.coef[2]

def solve(self) -> List:

a, b,c = map(float,self.get())

D = math.pow(b,2) - 4 \* a \* c

if (D < 0):

solutions = ()

if (D == 0):

solutions = -b / (2\*a)

if (D > 0):

sq = math.sqrt(D)

solutions = (-b + sq) / (2\*a), (-b - sq) / (2\*a)

return solutions

def main(\*args, \*\*kwargs):

initial = Quadratic(sys.argv).solve()

solutions = []

for i in initial:

solutions.extend(Quadratic("",1,0,-i).solve())

print(\*solutions)

return 0

if (\_\_name\_\_ == "\_\_main\_\_"): main()

**Вывод:**

