**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

«Основные конструкции языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил:  Каженец Д.Н.  ИУ5-31Б |  | Проверил:  Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Постановка задачи

Разработка алгоритма

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.

Код программы

import argparse

import math

def get\_coefficient(n):

while True:

try:

return float(input(n))

except ValueError:

print("Пожалуйста, введите корректное число.")

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

if a == 0:

print("Уравнение не является биквадратным.")

return []

discriminant = b \*\* 2 - 4 \* a \* c

if discriminant < 0:

return []

y1 = (-b + math.sqrt(discriminant)) / (2 \* a)

y2 = (-b - math.sqrt(discriminant)) / (2 \* a)

roots = set()

if y1 >= 0:

roots.add(math.sqrt(y1))

roots.add(-math.sqrt(y1))

if y2 >= 0 and y1 != y2:

roots.add(math.sqrt(y2))

roots.add(-math.sqrt(y2))

return list(roots)

def main():

parser = argparse.ArgumentParser()

parser.add\_argument('a', nargs='?', type=str)

parser.add\_argument('b', nargs='?', type=str)

parser.add\_argument('c', nargs='?', type=str)

args = parser.parse\_args()

a = float(args.a if args.a is not None else get\_coefficient("Введите коэффициент A: "))

b = float(args.b if args.b is not None else get\_coefficient("Введите коэффициент B: "))

c = float(args.c if args.c is not None else get\_coefficient("Введите коэффициент C: "))

if a == 0:

print("Коэффициент A не может быть равен нулю для квадратного уравнения.")

return roots = calculate\_roots(a, b, c)

if len(roots) == 4:

print(f"Уравнение имеет четыре действительных корня: {roots[0]}, {roots[1]}, {roots[2]} и {roots[3]}")

elif len(roots) == 3:

print(f"Уравнение имеет три действительных корня: {roots[0]}, {roots[1]} и {roots[2]}")

elif len(roots) == 2:

print(f"Уравнение имеет два действительных корня: {roots[0]} и {roots[1]}")

elif len(roots) == 1:

print(f"Уравнение имеет один действительный корень: {roots[0]}")

else:

print("Уравнение не имеет действительных корней.")

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

main()

**Анализ результатов**



