**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

«Основные конструкции языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнила: |  | Проверил: |
| Студентка группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Коренева София |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

**Описание задания**

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.
6. Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме С++).

**Текст программы**

import sys  
from math import sqrt  
  
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)  
 coef\_str = input()  
 # Переводим строку в действительное число  
 while True:  
 try:  
 float(coef\_str)  
 break  
 except:  
 print('Ошибка, введите число')  
 coef\_str = input()  
 coef = float(coef\_str)  
 return coef  
  
  
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  
 print(f"Дискриминант уравнения - {D}")  
 if D == 0.0:  
 root = -b / (2.0 \* a)  
 if root > 0:  
 result.append(sqrt(root))  
 result.append(-sqrt(root))  
 elif root == 0:  
 result.append(0)  
 elif D > 0.0:  
 sqD = sqrt(D)  
 root1 = (-b + sqD) / (2.0 \* a)  
 root2 = (-b - sqD) / (2.0 \* a)  
 if root1 > 0:  
 result.append(sqrt(root1))  
 result.append(-sqrt(root1))  
 elif root1 == 0:  
 result.append(0)  
 if root2 > 0:  
 result.append(sqrt(root2))  
 result.append(-sqrt(root2))  
 elif root2 == 0:  
 result.append(0)  
 result = sorted(result)  
 return result  
  
def main():  
 *'''  
 Основная функция  
 '''* 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("Данное уравнение не имеет действительных корней")  
 else:  
 print("Корни уравнения:")  
 for i in range(len\_roots):  
 print(f"{i + 1} корень = {roots[i]}")  
  
# Если сценарий запущен из командной строки  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Примеры выполнения программы**



