**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

«Изучение основных конструкций языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Нагапетян Валерий |  | Гапанюк Ю. Е. |
|  |  |  |

Москва, 2023 г.

**Задание**

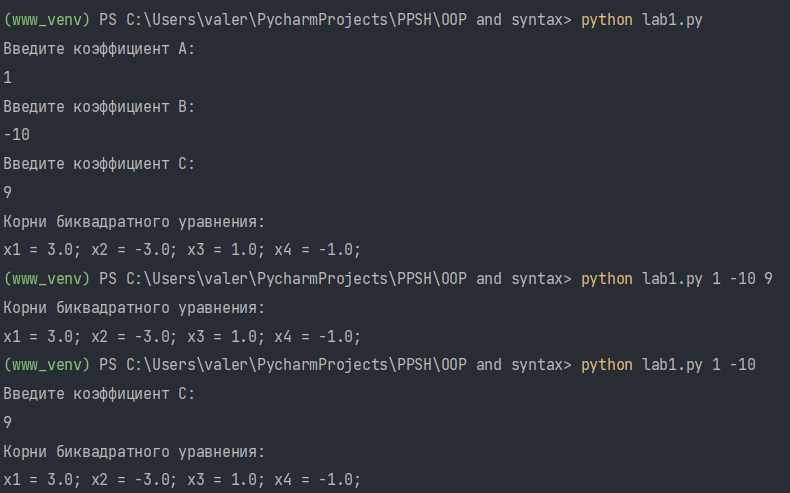
Разработать программу для решения [биквадратного уравнения.](https://ru.wikipedia.org/wiki/Уравнение_четвёртой_степени#Биквадратное_уравнение)

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.

**Текст программы**

*import* sys  
*import* math  
  
  
*class* SquareRoots:  
 *def* \_\_init\_\_(*self*):  
 *"""Конструктор класса"""  
  
 # Объявление коэффициентов  
 self*.coef\_A = 0.0  
 *self*.coef\_B = 0.0  
 *self*.coef\_C = 0.0  
  
 *# Количество корней  
 self*.num\_roots = 0  
  
 *# Список корней  
 self*.roots\_list = []  
  
 *def* get\_coef(*self*, 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 ,биквадратного уравнения  
 """  
 while True*:  
 *try*:  
 *# Пробуем прочитать коэффициент из командной строки* coef\_str = sys.argv[index]  
  
 *except* Exception:  
 *# Вводим с клавиатуры* print(prompt)  
 coef\_str = input()  
 *try*:  
 *# Переводим строку в действительное число* coef = float(coef\_str)  
 *return* coef  
 *except* ValueError:  
 print("Ошибка! Коэффициенты уравнение должны быть числового типа")  
  
 *def* get\_coefs(*self*):  
 *"""Чтение трех коэффициентов"""  
  
 self*.coef\_A = *self*.get\_coef(1, "Введите коэффициент А:")  
 *while self*.coef\_A == 0:  
 *self*.coef\_A = *self*.get\_coef(  
 1,  
 "Коэффициент а не может быть равен нулю! Иначе уравнение будет квадратным\nВведите 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*):  
 *"""Вычисление корней биквадратного уравнения"""* a = *self*.coef\_A  
 b = *self*.coef\_B  
 c = *self*.coef\_C  
 *# Вычисление дискриминанта и корней* bi\_root\_list = []  
 D = b \* b - 4 \* a \* c  
  
 *if* D == 0.0:  
 root = -b / (2.0 \* a)  
 bi\_root\_list.append(root)  
 *self*.processing\_roots(bi\_root\_list)  
 *elif* D > 0.0:  
 sqD = math.sqrt(D)  
 root1 = (-b + sqD) / (2.0 \* a)  
 root2 = (-b - sqD) / (2.0 \* a)  
 bi\_root\_list.append(root1)  
 bi\_root\_list.append(root2)  
 *self*.processing\_roots(bi\_root\_list)  
  
 *def* processing\_roots(*self*, roots):  
 *for* bi\_root *in* roots:  
 *if* bi\_root > 0:  
 *self*.roots\_list.append(math.sqrt(bi\_root))  
 *self*.roots\_list.append(-math.sqrt(bi\_root))  
 *self*.num\_roots += 2  
 *elif* bi\_root == 0:  
 *self*.roots\_list.append(0)  
 *self*.num\_roots += 1  
  
 *def* print\_roots(*self*):  
 *# Проверка отсутствия ошибок при вычислении корней  
 assert self*.num\_roots == len(  
 *self*.roots\_list  
 ), f"Ошибка! Уравнение содержит {*self*.num\_roots} \  
 действительных корней, но было вычислено {len(*self*.roots\_list)} корней."  
  
 *if self*.num\_roots == 0:  
 print("Нет корней")  
 *else*:  
 print("Корни биквадратного уравнения:")  
 counter = 1  
 *for* root *in self*.roots\_list:  
 print(f"x{counter} = {root}", end="; ")  
 counter += 1  
  
  
*def* main():  
 *"""Основная функция"""  
  
 # Создание объекта класса* r = SquareRoots()  
  
 *# Последовательный вызов необходимых методов* r.get\_coefs()  
 r.calculate\_roots()  
 r.print\_roots()  
  
  
*if* \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Результаты**

****