**Министерство науки и высшего образования Российской Федерации** **Федеральное государственное бюджетное образовательное учреждение высшего образования** **«Московский государственный технический университет** **имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий»

Лабораторная работа №1

«Основные конструкции языка Python»

Выполнил:

студент группы ИУ5-32Б

Долинский А.А.

Проверил:

Канев А.И.

2021 г.

**Цель лабораторной работы:**

Изучение основных конструкций языка Python.

**Постановка задачи:**

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_2021/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 кода:**

import sys  
import math  
  
  
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  
 '''* r,k,etalon =0,0, "-0123456789."  
 try:  
 # Пробуем прочитать коэффициент из командной строки  
 coef\_str = sys.argv[index]  
 except:  
 # Вводим с клавиатуры  
 print(prompt)  
 coef\_str = input()  
 while True:  
 for i in coef\_str:  
 if i in etalon:  
 if i == "-" or i == ".":  
 r+=1  
 else:  
 k+=1  
 else:  
 break  
 if k != 0 and k+r == len(coef\_str):  
 coef = float(coef\_str)  
 break  
 else:  
 coef\_str = input()  
 r, k = 0,0  
 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  
 if a == b == c == 0:  
 result = "12345"  
 elif a == b == 0:  
 result = []  
 elif a == 0:  
 if c/b <= 0:  
 if math.sqrt(abs(c/b)) == -math.sqrt(abs(c/b)):  
 result.append(math.sqrt(abs(c/b)))  
 else:  
 result.append(math.sqrt(abs(c/b)))  
 result.append(-math.sqrt(abs(c/b)))  
   
 else:  
 if D == 0.0:  
 if (-b / (2.0 \* a)) >= 0:  
 if math.sqrt(-b / (2.0 \* a)) == -math.sqrt(-b / (2.0 \* a)):  
 result.append(math.sqrt(-b / (2.0 \* a)))  
 else:  
 result.append(math.sqrt(-b / (2.0 \* a)))  
 result.append(-math.sqrt(-b / (2.0 \* a)))   
 elif D > 0.0:  
 sqD = math.sqrt(D)  
 root1 = (-b + sqD) / (2.0 \* a)  
 root2 = (-b - sqD) / (2.0 \* a)  
 if root1 == root2:  
 result.append(math.sqrt(root1))  
 result.append(-math.sqrt(root1))  
 else:  
 if root1 >= 0:  
 if math.sqrt(root1) == -math.sqrt(root1):  
 result.append(math.sqrt(root1))  
 else:  
 result.append(math.sqrt(root1))  
 result.append(-math.sqrt(root1))  
  
 if root2 >= 0:  
 if math.sqrt(root2) == -math.sqrt(root2):  
 result.append(math.sqrt(root2))  
 else:  
 result.append(math.sqrt(root2))  
 result.append(-math.sqrt(root2))  
 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('Нет корней')  
 elif len\_roots == 1:  
 print('Один корень: {}'.format(roots[0]))  
 elif len\_roots == 2:  
 print('Два корня: {} и {}'.format(roots[0], roots[1]))  
 elif len\_roots == 3:  
 print('Три корня: {}, {} и {}'.format(roots[0], roots[1], roots[2]))  
 elif len\_roots == 4:  
 print('Четыре корня: {}, {}, {} и {}'.format(roots[0], roots[1], roots[2], roots[3]))  
 elif len\_roots == 5:  
 print("Бесконечное количество корней")  
  
  
# Если сценарий запущен из командной строки  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Тестирование:**

|  |  |
| --- | --- |
| Ввод/ожидание: | Вывод: |
| 0 0 0 -­­­ Бесконечное число корней | Бесконечное число корней |
| 0 0 1 – Нет корней | Нет корней |
| 1 0 -4 - Два корня:  1.4142135623730951 и  -1.4142135623730951 | Два корня:  1.4142135623730951 и  -1.4142135623730951 |
| 1 -4 0 - Три корня: 2.0, -2.0 и 0 | Три корня: 2.0, -2.0 и 0 |
| 1 1 1 - Нет корней | Нет корней |
| 0 1 -4 - Два корня: 2.0 и -2.0 | Два корня: 2.0 и -2.0 |

**Вывод:**

Благодаря данной лабораторной работе я познакомился с базовым синтаксисом языка Python , разобрался с базовыми функциями , узнал и применил конструкцию try – except .