|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 **СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

**Отчет по лабораторной работе №1**

**по дисциплине Базовые компоненты интернет технологии**

#### Тема работы: "[Основные конструкции языка Python](https://github.com/ugapanyuk/BKIT_2022/wiki/lab_python_intro)"

**Выполнил:**

Студент группы ИУ5Ц-54Б

Перевощиков Н.Д. 26.10.22 г.

(дата, подпись)

**Проверил:**

Преподаватель

Канев А.И.

(дата, подпись)

Москва, 2022

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

**СОДЕРЖАНИЕ ОТЧЕТА**

[1. Цель лабораторной работы 3](#_Toc115088672)

[2. Описание задания 3](#_Toc115088673)

[3. Текст программы 4](#_Toc115088674)

[4. Результат 6](#_Toc115088675)

## Цель лабораторной работы

Изучение основных конструкций языка Phiton.

## Описание задания

Разработать программу для решения биквадратного уравнения.

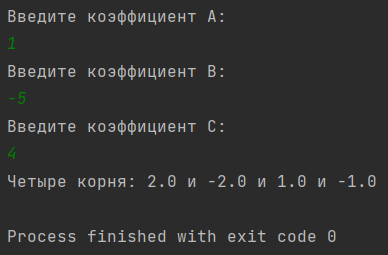
1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С могут быть заданы в виде параметров 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.

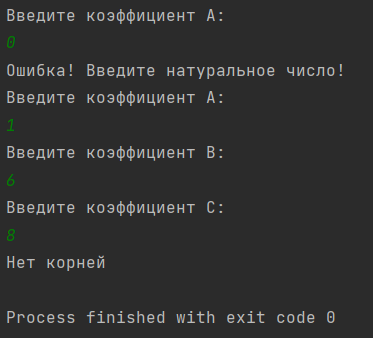
## Текст программы

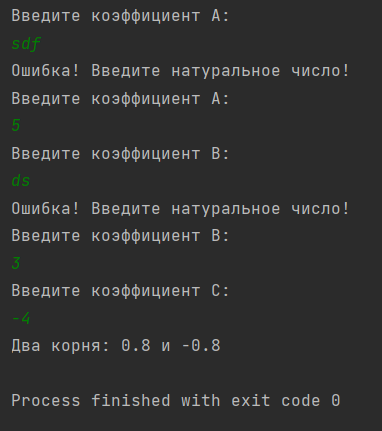
import sys  
import math  
  
def get\_coef(index, prompt):  
 *'''  
 Читаем коэффициент из командной строки или вводим с клавиатуры  
 Args:  
 index (int): Номер параметра в командной строке  
 prompt (str): Приглашение для ввода коэффицента  
 Returns:  
 float: Коэффициент квадратного уравнения  
 '''* try:  
 # Пробуем прочитать коэффициент из командной строки  
 coef\_str = sys.argv[index]  
  
 if(coef\_str[0] == '-'):  
 coef\_str = sys.argv[index].replace('-','')  
 else:  
 coef\_str = sys.argv[index]  
  
 if(coef\_str.isdigit() == True):  
 coef\_str = sys.argv[index]  
 else:  
 print('Ошибка! Введите натуральное число!')  
  
 except:  
 while True:  
 # Вводим с клавиатуры  
 print(prompt)  
 coef\_str = input()  
 # Проверка, есть ли минус числа и нулевой коэффициент?  
 if (coef\_str[0] != '0' or index == 2 or index == 3):  
 if (coef\_str[0] == '-'):  
 coef\_str\_buff = coef\_str.replace('-', '')  
 if (coef\_str\_buff.isdigit()):  
 break  
 if (coef\_str.isdigit()):  
 break  
  
 print("Ошибка! Введите натуральное число!")  
  
 # Переводим строку в действительное число  
 coef = float(coef\_str)  
 return coef  
  
  
def get\_roots(a, b, c):  
 result = []  
 D = b \* b - 4 \* a \* c  
  
 # Если дискриминат равен нулю, то корень может быть только одним  
 if D == 0.0:  
 root = -b / (2.0 \* a)  
 result.append(root)  
 if (root > 0.0):  
 root1 = math.sqrt(root)  
 result.append(root1)  
 result.append(-root1)  
  
 # Если дискриминат больше нуля, то количество кореней может быть четыре  
 elif D > 0.0:  
 sqD = math.sqrt(D)  
 root1 = (-b + sqD) / (2.0 \* a)  
 root2 = (-b - sqD) / (2.0 \* a)  
  
 if (root1 == 0):  
 result.append(abs(root1))  
 elif (root2 == 0):  
 result.append(abs(root2))  
  
 if (root1 > 0.0):  
 root3 = math.sqrt(root1)  
 result.append(root3)  
 result.append(-root3)  
  
 if (root2 > 0.0):  
 root4 = math.sqrt(root2)  
 result.append(root4)  
 result.append(-root4)  
  
 return result  
  
  
def main():  
 while True:  
 try:  
 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(round(roots[0], 2)))  
 elif len\_roots == 2:  
 print('Два корня: {} и {}'.format(round(roots[0], 2), round(roots[1], 2)))  
 elif len\_roots == 3 and roots[0] == 0.0:  
 print('Три корня: {} и {} и {}'.format(round(roots[0], 2), round(roots[1], 2), round(roots[2], 2)))  
 elif len\_roots == 3:  
 print('Два корня: {} и {}'.format(round(roots[1], 2), round(roots[2], 2)))  
 elif len\_roots == 4:  
 print('Четыре корня: {} и {} и {} и {}'.format(round(roots[0], 2), round(roots[1], 2),  
 round(roots[2], 2), round(roots[3], 2)))  
 break  
 except:  
 print('Ошибка заполнения!')  
 break  
  
# Если сценарий запущен из командной строки  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

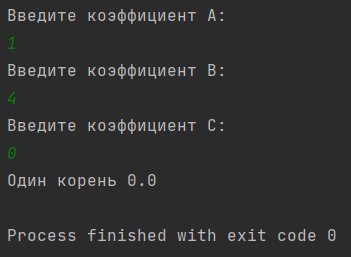
## Результат

**PyCharm**

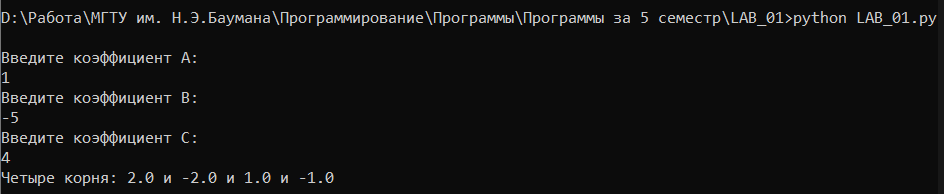
****

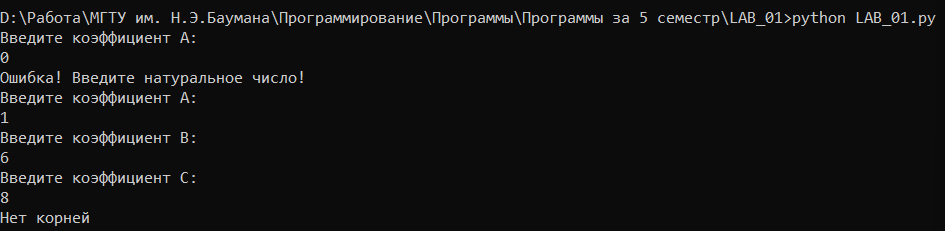
****

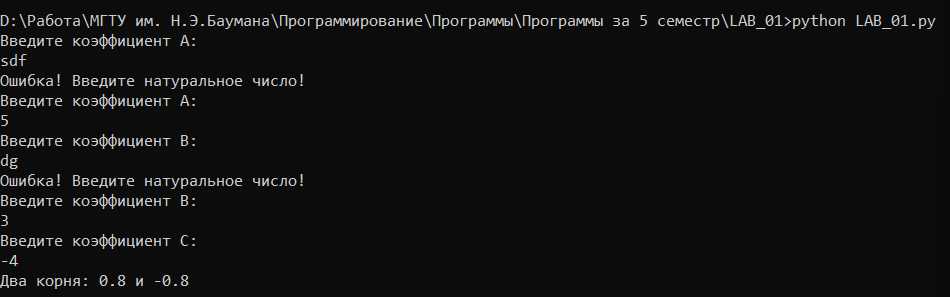
****

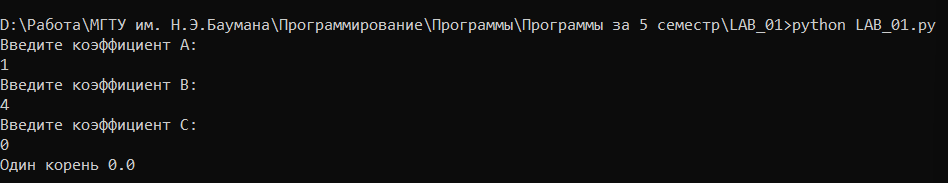
****

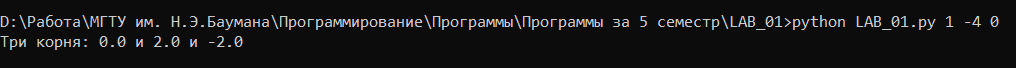
**Командная строка**

****

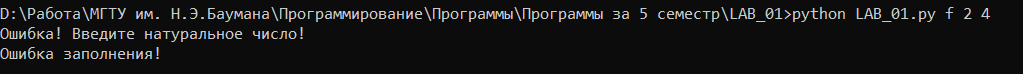
****

****

****

****

****

****