**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

# Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №1  
«Основные конструкции языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б: |  | преподаватель каф. ИУ5 |
| Балабанов Алексей Олегович |  | Гапонюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 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

'''

try:

# Пробуем прочитать коэффициент из командной строки

coef\_str = sys.argv[index]

except:

# Вводим с клавиатуры

coef\_str = "Flag"

while(isinstance(coef\_str,str)==True):

try:

coef\_str=float(coef\_str)

except:

print(prompt)

coef\_str = input()

# Переводим строку в действительное число

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

if D == 0.0:

if (-b / (2.0\*a))>0:

root01 = math.sqrt(-b / (2.0\*a))

root02 = -math.sqrt(-b / (2.0\*a))

result.append(root01)

result.append(root02)

if (-b / (2.0\*a))==0:

root00=0

result.append(root00)

elif D > 0.0:

sqD = math.sqrt(D)

if ((-b + sqD) / (2.0\*a))>0:

root1 = math.sqrt((-b + sqD) / (2.0\*a))

root3 = -math.sqrt((-b + sqD) / (2.0\*a))

result.append(root1)

result.append(root3)

elif ((-b + sqD) / (2.0\*a))==0:

root00=0

result.append(root00)

if ((-b - sqD) / (2.0\*a))>0:

root2 = math.sqrt((-b - sqD) / (2.0\*a))

root4 = -math.sqrt((-b - sqD) / (2.0\*a))

result.append(root2)

result.append(root4)

elif ((-b - sqD) / (2.0\*a))==0:

root00=0

result.append(root00)

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:')

if a==0:

if b == 0:

if c==0:

print("Бесконечное число корней")

else:

print("Корней нет")

else:

if (-c/b)>0:

print("Два корня: " + str(math.sqrt(-c/b)) +" и "+ str(-math.sqrt(-c/b)))

elif abs(-c/b)==0:

print("Два одинаковых корня: 0")

else:

print("Корней нет")

else:

# 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('Корни: {}'.format(roots[0]))

elif len\_roots == 2:

print('Два корня: {} и {}'.format(roots[0], roots[1]))

elif len\_roots == 3:

print('Три корня: {} и {} и {}'.format(roots[0], roots[1], roots[2]))

elif len\_roots == 4:

print('Четыре корня: {} и {} и {} и {}'.format(roots[0], roots[1], roots[2], roots[3]))

# Если сценарий запущен из командной строки

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

main()

# Пример запуска

# qr.py 1 0 -4

**Тестирование:**

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