



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**  
**Московский государственный технический университет**  
**им. Н.Э. Баумана**  
**(МГТУ им. Н.Э. Баумана)**

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

Отчёт по лабораторной работе № 1

По курсу: «Базовые компоненты интернет-технологий»

Выполнил:

Никулин Данила Дмитриевич  
студент группы ИУ5-31Б.

Проверил:

\_\_\_\_\_

Дата: \_\_\_\_ . \_\_\_\_ . 2022г.

Подпись: \_\_\_\_\_.

г. Москва 2022 г.

## Задание:

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  $A$ ,  $B$ ,  $C$ , вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
- 3) Коэффициенты  $A$ ,  $B$ ,  $C$  могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2.
- 4) Если коэффициент  $A$ ,  $B$ ,  $C$  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.

## Описание алгоритма:

В функции *get\_coef* получаем коэффициент биквадратного уравнения. Пробуем прочесть значение аргумента программы из командной строки. При неудаче вводим его с клавиатуры. Полученное значение проверяем на число. Если число, то возвращаем его из функции, иначе просим повторить ввод коэффициента, пока пользователь не введет корректные данные.

В функции *get\_roots* заполняем массив корнями биквадратного уравнения. Считаем дискриминант и получаем коэффициенты как в квадратном уравнении (*quadratic1(2)*). Проверяем, что они больше или равны нулю и считаем корни от них с  $+$  и  $-$ . Заполняем массив корнями и возвращаем его.

В функции *main* вызываем выше описанные функции и выводим результат. Вызываем функцию *get\_coef* для коэффициентов  $a$ ,  $b$  и  $c$ . Вызываем функцию *get\_roots* с коэффициентами  $a$ ,  $b$  и  $c$ . Считаем длину полученного массива корней. И выводим ответ в соответствии с количеством корней.

## Приложение 1. Текст программы:

```
import sys
import math

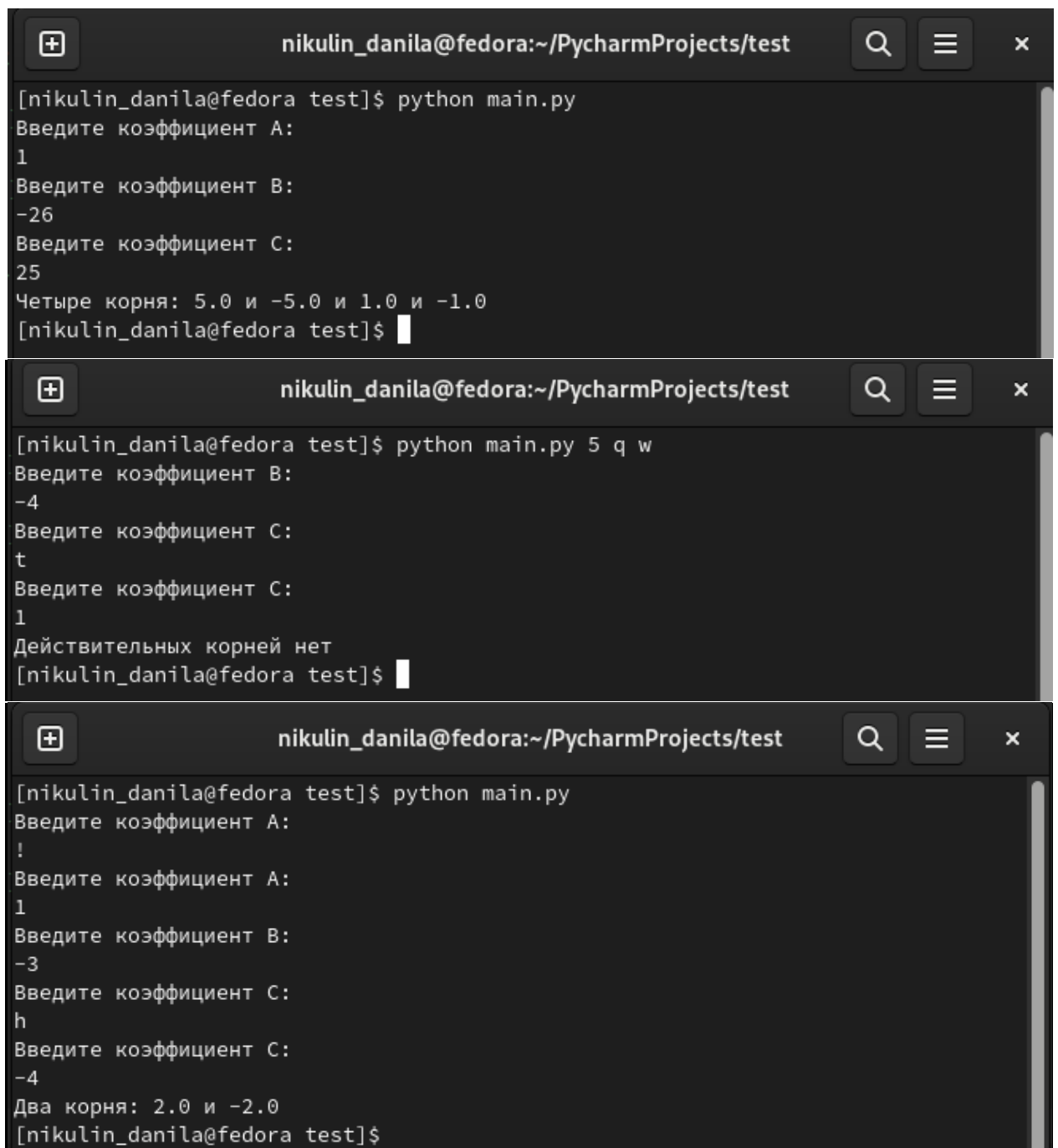
def get_coef(index, prompt):
    try:
        # Пробуем прочитать коэффициент из командной строки
        coef_str = sys.argv[index]
    except:
        # Вводим с клавиатуры
        print(prompt)
        coef_str = input()
    flag = False
    # Проверка на число
    while (flag == False):
        try:
            # Пробуем перевести строку в действительное число
            coef = float(coef_str)
        except:
            # При ошибке просим повторить ввод коэффициента
            print(prompt)
            coef_str = input()
        else:
            flag = True
    return coef

def get_roots(a, b, c):
    result = []
    D = b * b - 4 * a * c
    if D == 0.0:
        root = -b / (2.0 * a)
        result.append(root)
    elif D > 0.0:
        sqD = math.sqrt(D)
        quadratic1 = (-b + sqD) / (2.0 * a)
        quadratic2 = (-b - sqD) / (2.0 * a)
        # quadratic1(2) = x^2 (проверяем его на положительность)
        if (quadratic1 >= 0):
            root1 = math.sqrt(quadratic1)
            root2 = -root1
            result.append(root1)
            if (root1 != root2):
                result.append(root2)
        if (quadratic2 >= 0):
            root3 = math.sqrt(quadratic2)
            root4 = -root3
            result.append(root3)
            if (root3 != root4):
                result.append(root4)
    return result
```

```
def main():
    a = get_coef(1, 'Введите коэффициент A:')
    b = get_coef(2, 'Введите коэффициент B:')
    c = get_coef(3, 'Введите коэффициент C:')
    roots = get_roots(a, b, c)
    len_roots = len(roots)
    if len_roots == 0:
        print('Действительных корней нет')
    elif len_roots == 1:
        print(f'Один корень: {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]))
    else:
        print('Четыре корня: {} и {} и {} и {}'.format(roots[0], roots[1],
roots[2], roots[3]))

if __name__ == "__main__":
    main()
```

## Приложение 2. Результаты тестирования:



```
nikulin_danila@fedora:~/PycharmProjects/test
[nikulin_danila@fedora test]$ python main.py
Введите коэффициент A:
1
Введите коэффициент B:
-26
Введите коэффициент C:
25
Четыре корня: 5.0 и -5.0 и 1.0 и -1.0
[nikulin_danila@fedora test]$

nikulin_danila@fedora:~/PycharmProjects/test
[nikulin_danila@fedora test]$ python main.py 5 q w
Введите коэффициент B:
-4
Введите коэффициент C:
t
Введите коэффициент C:
1
Действительных корней нет
[nikulin_danila@fedora test]$

nikulin_danila@fedora:~/PycharmProjects/test
[nikulin_danila@fedora test]$ python main.py
Введите коэффициент A:
!
Введите коэффициент A:
1
Введите коэффициент B:
-3
Введите коэффициент C:
h
Введите коэффициент C:
-4
Два корня: 2.0 и -2.0
[nikulin_danila@fedora test]$
```