



**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 № 5

По курсу: «Базовые компоненты интернет-технологий»

Выполнил:

Строганов Георгий Константинович  
студент группы ИУ5-31Б.

Проверил:

\_\_\_\_\_

Дата: \_\_\_\_ . \_\_\_\_ . 2022г.

Подпись: \_\_\_\_\_.

г. Москва 2022 г.

## Задание:

1. Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3-4.
2. Модифицируйте код таким образом, чтобы он был пригоден для модульного тестирования.
3. Разработайте модульные тесты. В модульных тестах необходимо применить следующие технологии:
  - TDD - фреймворк (не менее 3 тестов).
  - BDD - фреймворк (не менее 3 тестов).
  - Создание Mock-объектов (необязательное дополнительное задание).

## Текст программы:

### Lab\_1.py

```
import sys
import math

def get_coef(index, prompt):
    try:
        # Попробуем прочитать коэффициент из командной строки
        coef_str = sys.argv[index]
    except:
        # Вводим с клавиатуры
        print(prompt)
        coef_str = input()
    # Обрабатываем неправильный ввод
    while True:
        try:
            coef = float(coef_str)
        except:
            print("Введены неправильные данные.", prompt)
            coef_str = input()
        else:
            return coef

def get_roots(a, b, c):
    result = []
    D = b*b - 4*a*c
    if D == 0.0:
        root = -b / (2.0*a)
        if root >= 0:
            result.append(math.sqrt(root))
            result.append(-math.sqrt(root))
    elif D > 0.0:
        sqD = math.sqrt(D)
        root1 = (-b + sqD) / (2.0*a)
        if root1 >= 0:
            if root1 == 0:
                result.append(root1)
            else:
                result.append(math.sqrt(root1))
                result.append(-math.sqrt(root1))
        root2 = (-b - sqD) / (2.0*a)
        if root2 >= 0:
            if root2 == 0.0:
```

```

        result.append(root2)
    else:
        result.append(math.sqrt(root2))
        result.append(-math.sqrt(root2))
    result = set(result)
    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:')
    # Вычисление корней
    roots = get_roots(a, b, c)
    # Вывод корней
    len_roots = len(roots)
    if len_roots == 0:
        print('Нет корней', end = " ")
        return
    elif len_roots == 1:
        print('Один корень:', end = " ")
    elif len_roots == 2:
        print('Два корня:', end = " ")
    elif len_roots == 3:
        print('Три корня:', end = " ")
    else:
        print('Четыре корня:')
    print(*roots, sep = ", ")
# Если сценарий запущен из командной строки
if __name__ == "__main__":
    main()

```

## test\_BDD.py

```

from Lab_1 import get_roots
from pytest_bdd import scenarios, given, when, then, parsers

scenarios("test_equation.feature")

@given(parsers.parse("The A coefficient {A:d}"), target_fixture = "coefA")
def t_root_input_1(A):
    return A

@given(parsers.parse('The B coefficient {B:d}'), target_fixture = "coefB")
def t_root_input_2(B):
    return B

@given(parsers.parse('The C coefficient {C:d}'), target_fixture = "coefC")
def t_root_input_3(C):
    return C

@when(parsers.parse('Solve the equation'), target_fixture = "equ")
def t_root_solve(coefA, coefB, coefC):
    return get_roots(coefA, coefB, coefC)

@then(parsers.parse("I get {zero:d} roots"))
def t_then(equ, zero):
    assert len(equ) == zero

```

## test\_equation.future

Feature: Scenario Outline

This app solve biquatratric equation

Scenario Outline: Solve the equation with correct value

Given The A coefficient <A>

And The B coefficient <B>

And The C coefficient <C>

When Solve the equation

Then I get <D> roots

Examples:

A	B	C	D
1	12	36	0
6	60	54	0
3	31	56	0
1	1	0	1
5	15	0	1
30	18	0	1
3	-5	-28	2
3	-14	-117	2
11	-86	-117	2
1	-9	0	3
3	-75	0	3
7	-112	0	3
7	-287	2800	4
13	-689	2548	4
1	-73	576	4

## test\_TDD.py

```
import pytest
from Lab_1 import get_roots

def tests_get_roots_zero():
    temp = get_roots(1, 12, 36)
    assert len(temp) == 0
    temp = get_roots(6, 60, 54)
    assert len(temp) == 0
    temp = get_roots(3, 31, 56)
    assert len(temp) == 0

def tests_get_roots_one():
    temp = get_roots(1, 1, 0)
    assert temp == {0}
    temp = get_roots(5, 15, 0)
    assert temp == {0}
    temp = get_roots(30, 18, 0)
    assert temp == {0}

def tests_get_roots_two():
    temp = get_roots(3, -5, -28)
    assert temp == {2, -2}
    temp = get_roots(3, -14, -117)
    assert temp == {3, -3}
    temp = get_roots(11, -86, -117)
```

```

    assert temp == {3, -3}

def tests_get_roots_three():
    temp = get_roots(1, -9, 0)
    assert temp == {-3, 0, 3}
    temp = get_roots(3, -75, 0)
    assert temp == {-5, 0, 5}
    temp = get_roots(7, -112, 0)
    assert temp == {-4, 0, 4}

def tests_get_roots_four():
    temp = get_roots(7, -287, 2800)
    assert temp == {-5, -4, 4, 5}
    temp = get_roots(13, -689, 2548)
    assert temp == {-7, -2, 2, 7}
    temp = get_roots(1, -73, 576)
    assert temp == {-8, -3, 3, 8}

```

## Результаты тестирования:

```

D:\pp\BKIT_3sem\BKIT_3sem\Lab_5>pytest test_BDD.py
===== test session starts =====
platform win32 -- Python 3.10.5, pytest-7.1.3, pluggy-1.0.0
rootdir: D:\pp\BKIT_3sem\BKIT_3sem\Lab_5
plugins: bdd-6.0.1
collected 15 items

test_BDD.py ..... [100%]

===== 15 passed in 0.13s =====

D:\pp\BKIT_3sem\BKIT_3sem\Lab_5>pytest test_TDD.py
===== test session starts =====
platform win32 -- Python 3.10.5, pytest-7.1.3, pluggy-1.0.0
rootdir: D:\pp\BKIT_3sem\BKIT_3sem\Lab_5
plugins: bdd-6.0.1
collected 5 items

test_TDD.py ..... [100%]

===== 5 passed in 0.05s =====

D:\pp\BKIT_3sem\BKIT_3sem\Lab_5>

```