

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»
Отчёт по лабораторной работе №5.

Выполнила:

студентка группы ИУ5-31Б
Слепченкова Светлана
Дмитриевна

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Дата: 09.12.2022

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

main.py

```
import sys
import math
#Сделать вывод о том что некорректный ввод
def get_coef(index, prompt):
    try:
        # Пробуем прочесть коэффициент из командной
        строки
        coef_str = sys.argv[index]
    except:
        # Вводим с клавиатуры
        print(prompt)
        coef_str = input()
    # Переводим строку в действительное число
    flag = True
    while flag:
        try:
            float(coef_str)
            flag = False
        except:
            coef_str = input()
    coef = float(coef_str)
    return coef
```

```

def get_roots(a, b, c):
    result = []
    D = b * b - 4 * a * c
    if D == 0.0:
        t = -b / (2.0 * a)
        result.append(math.sqrt(t))
        result.append(-math.sqrt(t))
    elif D > 0.0:
        sqD = math.sqrt(D)
        t1 = ((-b + sqD) / (2.0 * a))
        t2 = ((-b - sqD) / (2.0 * a))
        if (t1 >= 0):
            result.append(-math.sqrt(t1))
            result.append(math.sqrt(t1))
        if (t2 >= 0):
            result.append(-math.sqrt(t2))
            result.append(math.sqrt(t2))
    result = list(set(result))
    return sorted(result)

def main():
    a = get_coef(1, 'Введите коэффициент А:')
    b = get_coef(2, 'Введите коэффициент В:')
    c = get_coef(3, 'Введите коэффициент С:')
    roots = get_roots(a, b, c)
    # Вывод корней
    roots = sorted(roots)
    len_roots = len(roots)
    if len_roots == 0:
        print('Нет корней')
    elif len_roots == 1:
        print('Один корень: {}'.format(roots[0] * (-1) *
int(roots[0] == -0.0)))
    elif len_roots == 2:

```

```

        print('Два корня: {}, {}'.format(roots[0],
roots[1]))
    elif len_roots == 3:
        print('Три корня: {}, {}, {}'.format(roots[0],
roots[1], roots[2]))
    elif len_roots == 4:
        print('Четыре корня: {}, {}, {},
{}'.format(roots[0], roots[1], roots[2], roots[3]))

# Если сценарий запущен из командной строки
if __name__ == "__main__":
    main()

```

BDD_test_get_roots.py

```

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

scenarios("test_equation.feature")

@given(parsers.parse("The A coff {A:d}"), target_fixture
= "cofA")
def t_root_input_A(A):
    return A

@given(parsers.parse("The B coff {B:d}"), target_fixture
= "cofB")
def t_root_input_B(B):
    return B

@given(parsers.parse("The C coff {C:d}"), target_fixture
= "cofC")
def t_root_input_C(C):
    return C

```

```

@when(parsers.parse("Solve the equation"),
target_fixture = "res1")
def t_root_solve(cofA, cofB, cofC):
    return get_roots(cofA, cofB, cofC)

@then(parsers.parse("Get count {result:d} roots"))
def t_tehn(res1, result):
    assert result == len(res1)

```

Test_get_roots.py

```

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

scenarios("test_equation.feature")

@given(parsers.parse("The A coff {A:d}"), target_fixture
= "cofA")
def t_root_input_A(A):
    return A

@given(parsers.parse("The B coff {B:d}"), target_fixture
= "cofB")
def t_root_input_B(B):
    return B

@given(parsers.parse("The C coff {C:d}"), target_fixture
= "cofC")
def t_root_input_C(C):
    return C

@when(parsers.parse("Solve the equation"),
target_fixture = "res1")

```

```
def t_root_solve(cofA, cofB, cofC):
    return get_roots(cofA, cofB, cofC)

@then(parsers.parse("Get count {result:d} roots"))
def t_tehn(res1, result):
    assert result == len(res1)
```

test_equation.feature

Feature Scenario Outline

This function solve biquatratric equsion

Scenario Outline: Solve the equation

Given The A coff <A>

And The B coff

And The C coff <C>

When Solve the equation

Then Get count <D> roots

Examples:

| A | B | C | D |
|---|-----|-----|---|
| 4 | 0 | 3 | 0 |
| 1 | 1 | 0 | 1 |
| 3 | -5 | -28 | 2 |
| 1 | -9 | 0 | 3 |
| 1 | -73 | 576 | 4 |

Анализ результатов

```
C:\Users\Света\Desktop\iu5_bkit\lab_5>pytest BDD_test_get_roots.py
===== test session starts =====
platform win32 -- Python 3.11.0, pytest-7.2.0, pluggy-1.0.0
rootdir: C:\Users\Света\Desktop\iu5_bkit\lab_5
plugins: anyio-3.6.2, bdd-6.1.1
collected 5 items

BDD_test_get_roots.py ..... [100%]

===== 5 passed in 0.04s =====

C:\Users\Света\Desktop\iu5_bkit\lab_5>pytest test_get_roots.py
===== test session starts =====
platform win32 -- Python 3.11.0, pytest-7.2.0, pluggy-1.0.0
rootdir: C:\Users\Света\Desktop\iu5_bkit\lab_5
plugins: anyio-3.6.2, bdd-6.1.1
collected 5 items

test_get_roots.py ..... [100%]

===== 5 passed in 0.04s =====
```