

**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»**

**Лабораторная работа №5
«Модульное тестирование в Python»
по предмету
«Базовые компоненты интернет-технологий»**

**Выполнил:
студент группы № ИУ5-31Б
Изибаев Андрей**

**Проверил:
Преподаватель кафедры ИУ-5
Гапанюк Юрий**

2022 г.

Постановка задачи

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

Задание

В качестве объекта для тестирования была выбрана программа, находящая корни биквадратного уравнения

Текст программы

main.py

```
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()
# 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_TDD.py

```

import pytest
from main import get_roots

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}

```

test_BDD.py

```

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

scenarios("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

```

equation.feature

Feature: Scenario Outline

This app solve biquadratic equation

Scenario Outline: Solve the equation with correct value

Given The A coefficient <A>

And The B coefficient

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

Анализ результатов

```
C:\Users\ASUS\PycharmProjects\pythonProject3(lab5)\venv\Scripts\python.exe "C:/Program Files/Jet
Testing started at 22:35 ...
Launching pytest with arguments C:/Users/ASUS/PycharmProjects/pythonProject3(lab5)/test_TDD.py -

===== test session starts =====
collecting ... collected 3 items

test_TDD.py::tests_get_roots_one PASSED [ 33%]
test_TDD.py::tests_get_roots_two PASSED [ 66%]
test_TDD.py::tests_get_roots_three PASSED [100%]

===== 3 passed in 0.03s =====

Process finished with exit code 0
```

```
C:\Users\ASUS\PycharmProjects\pythonProject3(lab5)\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition 2022.1
Testing started at 22:36 ...
Launching pytest with arguments C:/Users/ASUS/PycharmProjects/pythonProject3(lab5)/test_BDD.py --no-header --no-summary -q in C:/Users\

===== test session starts =====
collecting ... collected 9 items

test_BDD.py::test_solve_the_equation_with_correct_value[1-12-36-0] <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [ 11%]
test_BDD.py::test_solve_the_equation_with_correct_value[6-60-54-0] <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [ 22%]
test_BDD.py::test_solve_the_equation_with_correct_value[3-31-56-0] <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [ 33%]
test_BDD.py::test_solve_the_equation_with_correct_value[1-1-0-1] <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [ 44%]
test_BDD.py::test_solve_the_equation_with_correct_value[5-15-0-1] <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [ 55%]
test_BDD.py::test_solve_the_equation_with_correct_value[30-18-0-1] <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [ 66%]
test_BDD.py::test_solve_the_equation_with_correct_value[3--5--28-2] <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [ 77%]
test_BDD.py::test_solve_the_equation_with_correct_value[3--14--117-2] <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [ 88%]
test_BDD.py::test_solve_the_equation_with_correct_value[11--86--117-2] <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [100%]

===== 9 passed in 0.06s =====

Process finished with exit code 0
```