



**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:

Никулин Данила Дмитриевич  
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).

## Приложение 1. Текст программы:

### test\_TDD.py

```
import pytest
from main 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}

```

## test\_BDD.py

```

from main 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):
    print("=====")
    print(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.feature

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
-5	-4	4	5
-7	-2	2	7
-8	-3	3	8

$$\begin{array}{ccc|c} 1 & 12 & 36 & 0 \\ 6 & 60 & 54 & 0 \\ 3 & 31 & 56 & 0 \end{array}$$

$$\begin{array}{ccc|c} 1 & 1 & 0 & 1 \\ 5 & 15 & 0 & 1 \\ 30 & 18 & 0 & 1 \end{array}$$

$$\begin{array}{ccc|c} 3 & -5 & -28 & 2 \\ 3 & -14 & -117 & 2 \\ 11 & -86 & -117 & 2 \end{array}$$

$$\begin{array}{ccc|c} 1 & -9 & 0 & 3 \\ 3 & -75 & 0 & 3 \\ 7 & -112 & 0 & 3 \end{array}$$

$$\begin{array}{ccc|c} 7 & -287 & 2800 & 4 \\ 13 & -689 & 2548 & 4 \\ 1 & -73 & 576 & 4 \end{array}$$

## 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()
    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)
    # y = x^2 проверяем игрек 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:

```
(myvenv) nikulin_danila@ubuntu:~/github/IU5_BKIT2022/lab5$ pytest
===== test session starts =====
platform linux -- Python 3.10.6, pytest-7.2.0, pluggy-1.0.0
rootdir: /home/nikulin_danila/github/IU5_BKIT2022/lab5
plugins: bdd-6.1.1
collected 20 items

test_BDD.py F..... [ 75%]
test_TDD.py FFFFFF [100%]

===== FAILURES =====
test_solve_the_equation_with_correct_value[1-12-36-0]
-----
fixturefunc = <function t_then at 0x7f121a804700>, request = <FixtureRequest for <Function test_solve_the_equation_with_correct_value[1-12-36-0]>>, kwargs = {'equ': [-6.0], 'zero': 0}

    def call_fixture_func(
        fixturefunc: "FixtureFunc[FixtureValue]", request: FixtureRequest, kwargs
    ) -> FixtureValue:
        if is_generator(fixturefunc):
            fixturefunc = cast(
                Callable[..., Generator[FixtureValue, None, None]], fixturefunc
            )
            generator = fixturefunc(**kwargs)
            try:
                fixture_result = next(generator)
            except StopIteration:
                raise ValueError(f"{request.fixturename} did not yield a value") from None
            finalizer = functools.partial(_teardown_yield_fixture, fixturefunc, generator)
            request.addfinalizer(finalizer)
        else:
            fixturefunc = cast(Callable[..., FixtureValue], fixturefunc)
            fixture_result = fixturefunc(**kwargs)

myvenv/lib/python3.10/site-packages/_pytest/fixtures.py:907:
-----
equ = [-6.0], zero = 0

    @then(parsers.parse("I get {zero:d} roots"))
    def t_then(equ, zero):
>         assert len(equ) == zero
E         assert 1 == 0
E         + where 1 = len([-6.0])

----- Captured stdout call -----
=====
12

def tests_get_roots_zero():
    temp = get_roots(1, 12, 36)
>     assert len(temp) == 0
E     assert 1 == 0
E     + where 1 = len([-6.0])

test_TDD.py:6: AssertionError
----- tests_get_roots_one -----

def tests_get_roots_one():
    temp = get_roots(1, 1, 0)
>     assert temp == {}
E     assert [0.0] == {}
E     Use -v to get more diff

test_TDD.py:14: AssertionError
----- tests_get_roots_two -----

def tests_get_roots_two():
    temp = get_roots(3, -5, -28)
>     assert temp == {2, -2}
E     assert [2.0, -2.0] == {-2, 2}
E     Use -v to get more diff

test_TDD.py:22: AssertionError
----- tests_get_roots_three -----

def tests_get_roots_three():
    temp = get_roots(1, -9, 0)
>     assert temp == {-3, 0, 3}
E     assert [3.0, -3.0, 0.0] == {-3, 0, 3}
E     Use -v to get more diff

test_TDD.py:30: AssertionError
----- tests_get_roots_four -----

def tests_get_roots_four():
    temp = get_roots(7, -287, 2800)
>     assert temp == {-5, -4, 4, 5}
E     assert [5.0, -5.0, 4.0, -4.0] == {-5, -4, 4, 5}
test_TDD.py:38: AssertionError
===== short test summary info =====
FAILED test_BDD.py::test_solve_the_equation_with_correct_value[1-12-36-0] - assert 1 == 0
FAILED test_TDD.py::tests_get_roots_zero - assert 1 == 0
FAILED test_TDD.py::tests_get_roots_one - assert [0.0] == {}
FAILED test_TDD.py::tests_get_roots_two - assert [2.0, -2.0] == {-2, 2}
FAILED test_TDD.py::tests_get_roots_three - assert [3.0, -3.0, 0.0] == {-3, 0, 3}
FAILED test_TDD.py::tests_get_roots_four - assert [5.0, -5.0, 4.0, -4.0] == {-5, -4, 4, 5}
===== 6 failed, 14 passed in 0.14s =====
```