



**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
по предмету
«Базовые компоненты интернет-технологий»

Выполнил:
студент группы ИУ5-35Б
Сулайманов Роман

Проверил:
Преподаватель кафедры ИУ-5
Гапанюк Юрий

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).

Код программы

Программа состоит из нескольких файлов.

function.py

```
from math import sqrt

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 ** 0.5))
        result.append((root ** 0.5))
    elif D > 0.0:
        sqD = sqrt(D)
        root1 = (-b + sqD) / (2.0 * a)
        root2 = (-b - sqD) / (2.0 * a)
        if root1 > 0:
            result.append(-(root1 ** 0.5))
            result.append((root1 ** 0.5))
        if root2 > 0:
            result.append(-(root2 ** 0.5))
            result.append((root2 ** 0.5))
    return result
```

testing.py

```
import unittest
from math import *
from function import get_roots
from pytest_bdd import scenario, given, when, then

# TDD
class programm_test(unittest.TestCase):
    def test1(self):
        self.assertEqual(get_roots(1, -5, 6), [-sqrt(3), sqrt(3), -sqrt(2), sqrt(2)])

    def test2(self):
        self.assertEqual(get_roots(1, -4, 4), [-sqrt(2), sqrt(2)])

    def test3(self):
        self.assertEqual(get_roots(1, 0, 10), [])

    def test4(self):
        self.assertEqual(get_roots(1, 0, 10), [1])

# BDD
@scenario("scenarios.feature", "4 roots")
def test0():
    print('Scenarios: 4 roots')

@given("Data for equation")
def test1():
    print("\nData for equation")
    print(f'Data: {[1, -5, 6]}\n')

@when('I want to solve the equation')
```

```

def test2():
    print('I want to solve the equation')

@then('I get roots')
def test3():
    print('I get roots')
    assert get_roots(1, -5, 6) == [-sqrt(3), sqrt(3), -sqrt(2), sqrt(2)]

@then('I get roots')
def test4():
    print('I get roots')
    assert get_roots(1, -5, 6) == [-sqrt(3), sqrt(3), -sqrt(2)]

def main():
    unittest.main()

if __name__ == '__main__':
    main()

```

scenarios.feature

```

Feature: Testing

    Scenario: 4 roots
        Given Data for equation

        When I want to solve the equation

        Then I get roots

```

Результаты выполнения

PyCharm IDE showing test results for `testing.py`. The test suite `test4` failed, with a 44% pass rate. The error message indicates that the expected list is empty, but the actual result is `[1]`.

```
def test4():
    print('I get roots')
    assert get_roots(1, -5, 6) == [-sqrt(3), sqrt(3), -sqrt(2), sqrt(2)]
```

Test Results:

- test1: 0 ms
- test2: 0 ms
- test3: 0 ms
- test4: 6 ms
- test0: 1 ms
- test1: 0 ms
- test2: 0 ms
- test3: 0 ms
- test4: 0 ms

Test Results Summary:

Test	Duration	Status
test1	0 ms	Passed
test2	0 ms	Passed
test3	0 ms	Passed
test4	6 ms	Failed
test0	1 ms	Passed
test1	0 ms	Passed
test2	0 ms	Passed
test3	0 ms	Passed
test4	0 ms	Passed

Test Results Summary:

Test	Duration	Status
test1	0 ms	Passed
test2	0 ms	Passed
test3	0 ms	Passed
test4	6 ms	Failed
test0	1 ms	Passed
test1	0 ms	Passed
test2	0 ms	Passed
test3	0 ms	Passed
test4	0 ms	Passed

PyCharm IDE showing test results for `testing.py`. The test suite `test4` failed, with a 100% pass rate. The error message indicates that the expected list is empty, but the actual result is `[1]`.

```
def test4():
    print('I get roots')
    assert get_roots(1, -5, 6) == [-sqrt(3), sqrt(3), -sqrt(2), sqrt(2)]
```

Test Results:

- test1: 0 ms
- test2: 0 ms
- test3: 0 ms
- test4: 6 ms
- test0: 1 ms
- test1: 0 ms
- test2: 0 ms
- test3: 0 ms
- test4: 0 ms

Test Results Summary:

Test	Duration	Status
test1	0 ms	Passed
test2	0 ms	Passed
test3	0 ms	Passed
test4	6 ms	Failed
test0	1 ms	Passed
test1	0 ms	Passed
test2	0 ms	Passed
test3	0 ms	Passed
test4	0 ms	Passed

Test Results Summary:

Test	Duration	Status
test1	0 ms	Passed
test2	0 ms	Passed
test3	0 ms	Passed
test4	6 ms	Failed
test0	1 ms	Passed
test1	0 ms	Passed
test2	0 ms	Passed
test3	0 ms	Passed
test4	0 ms	Passed