

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

**Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Базовые компоненты интернет-технологий»  
Отчёт по лабораторной работе №4**

Выполнил:  
студент группы ИУ5-31Б  
Смыслов Дмитрий  
Олегович

Подпись: \_\_\_\_\_

Дата: \_\_\_\_\_

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Юрий  
Евгеньевич

Подпись: \_\_\_\_\_

Дата: \_\_\_\_\_

Москва, 2021 г.

## Описание задания

Написать тесты для своей программы решения биквадратного уравнения. В этом случае, возможно, потребуется доработать программу решения биквадратного уравнения, чтобы она была пригодна для модульного тестирования.

В модульных тестах необходимо применить следующие технологии:

TDD - фреймворк.

BDD - фреймворк.

Создание Моск-объектов.

## Текст программы

Файл main.py

```
import sys
import math

def is_float(arg):
    try:
        float(arg)
        return True
    except ValueError:
        return False

def get_coefficient(index, prompt):
    try:
        coefficient_str = sys.argv[index]
        while not is_float(coefficient_str):
            coefficient_str = input(prompt)
    except:
        coefficient_str = input(prompt)
        while not is_float(coefficient_str):
            coefficient_str = input(prompt)
    return float(coefficient_str)

def get_quadratic_roots(a, b, c):
    result = []
    if a == 0:
        if b == 0:
            if c == 0:
                result.append('R')
            else:
                if c == 0:
                    result.append(0.0)
                else:
                    result.append(-c / b)
    else:
        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)
        root1 = (-b + sqD) / (2.0 * a)
        root2 = (-b - sqD) / (2.0 * a)
        result.append(root1)
        result.append(root2)
    return result

def get_biquadratic_roots(a, b, c):
    roots = []
    sub_roots = get_quadratic_roots(a, b, c)
    if len(sub_roots) != 0 and sub_roots[0] == 'R':
        roots.append('R')
    else:
        for sub_root in sub_roots:
            for root in get_quadratic_roots(1, 0, -sub_root):
                roots.append(root)
    return roots

def main():
    a = get_coefficient(1, 'Введите коэффициент A: ')
    b = get_coefficient(2, 'Введите коэффициент B: ')
    c = get_coefficient(3, 'Введите коэффициент C: ')

    roots = get_biquadratic_roots(a, b, c)

    len_roots = len(roots)
    if len_roots == 0:
        print('Нет корней')
    elif roots[0] == 'R':
        print('Корень - любое число')
    else:
        print('Корни', end=': ')
        for root in roots:
            print(root, end='; ')

if __name__ == "__main__":
    main()

```

В качестве TDD фреймворка использовался unittest  
 Для создания Mock-объектов использовался декоратор patch из  
 unittest.mock

Файл TDD\_tests.py

```

import unittest
from unittest.mock import patch
import main

class BiquadricRootsTest(unittest.TestCase):

    def test_no_roots(self):
        #  $x^4 + 2x^2 + 3 = 0$  - не имеет корней
        roots = main.get_biquadratic_roots(1, 2, 3) # объект на проверку
        self.assertEqual(roots, []) # проверка

    def test_one_root(self):
        #  $x^4 = 0$  - имеет единственный корень: 0
        roots = main.get_biquadratic_roots(1, 0, 0)
        self.assertEqual(roots, [0])

```

```

def test_two_roots(self):
    #  $x^4 - 2x^2 + 1 = 0$  - имеет два корня: 1, -1
    roots = main.get_biquadratic_roots(1, -2, 1)
    self.assertEqual(roots, [1, -1]) # можно list привести к set и
    # затем сравнивать assertEquals

def test_three_roots(self):
    #  $x^4 - 2x^2 = 0$  - имеет корни: 0, -2, 2
    roots = main.get_biquadratic_roots(1, -4, 0)
    self.assertEqual(roots, [-2, 0, 2])

def test_four_roots(self):
    #  $x^4 - 109x^2 + 900 = 0$  - имеет корни: 3, -3, 10, -10
    roots = main.get_biquadratic_roots(1, -109, 900)
    self.assertEqual(roots, [-10, 3, -3, 10])

def test_inf_roots(self):
    #  $0x^4 + 0x^2 + 0 = 0$  - тождество => x - любое действительное
    roots = main.get_biquadratic_roots(0, 0, 0)
    self.assertEqual(roots, ['R'])

@patch('main.get_biquadratic_roots', return_value=0)
def test_mock(self, get_biquadratic_roots):
    self.assertEqual(get_biquadratic_roots(121, -3212, 0.3), 0)

if __name__ == '__main__':
    unittest.main()

```

В качестве BDD фреймворка использовался behave  
 Файл с тестами: biquadric\_roots.feature

```

Feature: BDD tests

    Scenario: Проверка корней биквадратного уравнения
        Given  $16x^4 + -8x^2 + 1 = 0$ 
        When Решаю уравнение
        Then Ожидаю получить корни 0.5 и -0.5

```

Файл с реализацией шагов теста: steps.py

```

# -*- coding: utf-8 -*-
from behave import given, when, then
from main import *

@given("{a}x^4 + {b}x^2 + {c} = 0")
def step_impl(context, a: float, b: float, c: float):
    context.a = float(a)
    context.b = float(b)
    context.c = float(c)

@when("Решаю уравнение")
def step_impl(context):
    context.roots = get_biquadratic_roots(context.a, context.b, context.c)

@then("Ожидаю получить корни {root1} и {root2}")
def step_impl(context, root1: float, root2: float):
    result = [float(root2), float(root1)]
    assert set(context.roots) == set(result)

```

# Экранные формы с примерами выполнения программы

```
Terminal: Local x + v
(venv) PS D:\JetBrains\PyCharm 2021.2.2\PycharmProjects\LAB4> python -m TDD_tests -v
test_four_roots (__main__.BiquadricRootsTest) ... ok
test_inf_roots (__main__.BiquadricRootsTest) ... ok
test_mock (__main__.BiquadricRootsTest) ... ok
test_no_roots (__main__.BiquadricRootsTest) ... ok
test_one_root (__main__.BiquadricRootsTest) ... ok
test_three_roots (__main__.BiquadricRootsTest) ... ok
test_two_roots (__main__.BiquadricRootsTest) ... ok

-----

Ran 7 tests in 0.002s

OK
(venv) PS D:\JetBrains\PyCharm 2021.2.2\PycharmProjects\LAB4> 
```

```
Terminal: Local x + v
OK
(venv) PS D:\JetBrains\PyCharm 2021.2.2\PycharmProjects\LAB4> behave features/biquadric_roots.feature
Feature: BDD tests # features/biquadric_roots.feature:1

  Scenario: Проверка корней биквадратного уравнения # features/biquadric_roots.feature:3
    Given  $16x^4 + -8x^2 + 1 = 0$  # steps/steps.py:6
    When Решаю уравнение # steps/steps.py:13
    Then Ожидаю получить корни 0.5 и -0.5 # steps/steps.py:18

1 feature passed, 0 failed, 0 skipped
1 scenario passed, 0 failed, 0 skipped
3 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.000s
(venv) PS D:\JetBrains\PyCharm 2021.2.2\PycharmProjects\LAB4> 
```