**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.

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31б  Садиров Ф.Э. |  | преподаватель каф. ИУ5  Гапанюк Ю.Е. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

**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:
   1. TDD - фреймворк (не менее 3 тестов).
   2. BDD - фреймворк (не менее 3 тестов).
   3. Создание Mock-объектов (необязательное дополнительное задание).

**Текст программы**

***main.py***

import math  
  
  
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))  
 elif (t1 == 0 or t2 == 0):  
 result.append(0.0)  
 return set(result)

***test\_BDD.py***

from main import get\_roots  
from pytest\_bdd import scenarios, given, when, then, parsers  
  
scenarios("Bdd.feature")  
  
  
@given(parsers.parse("A coefficient is {A:d}"), target\_fixture="coefficientA")  
def input\_A(A):  
 return A  
  
  
@given(parsers.parse("B coefficient is {B:d}"), target\_fixture="coefficientB")  
def input\_B(B):  
 return B  
  
  
@given(parsers.parse("C coefficient is {C:d}"), target\_fixture="coefficientC")  
def input\_C(C):  
 return C  
  
  
@when(parsers.parse("Solve the equation"), target\_fixture="solvation")  
def solving\_the\_equation(coefficientA, coefficientB, coefficientC):  
 return get\_roots(coefficientA, coefficientB, coefficientC)  
  
  
@then(parsers.parse("Get number of roots {result:d} and compare"))  
def getting\_the\_result(solvation, result):  
 assert result == len(solvation)

***test\_TDD.py***

import pytest  
from main import \*  
  
  
def test\_0\_if\_zero\_root():  
 assert len(get\_roots(1, -1, 1)) == 0  
  
  
def test\_1\_if\_one\_root():  
 assert len(get\_roots(1, 0, 0)) == 1  
  
  
def test\_2\_if\_two\_roots():  
 assert len(get\_roots(1, -2, -8)) == 2  
  
  
def test\_3\_if\_three\_roots():  
 assert len(get\_roots(10, -5, 0)) == 3  
  
  
def test\_4\_if\_four\_roots():  
 assert len(get\_roots(4, -5, 1)) == 4

***Bdd.feature***

Feature: Scenario Outline  
 Testing the code  
  
 Scenario Outline: Solve the equation  
 Given A coefficient is <A>  
 And B coefficient is <B>  
 And C coefficient is <C>  
 When Solve the equation  
 Then Get number of roots <D> and compare  
  
 Examples:  
 | A | B | C | D |  
  
 | 1 | -1 | 1 | 0 |  
 | 1 | 0 | 0 | 1 |  
 | 1 | -2 | -8 | 2 |  
 | 10 | -5 | 0 | 3 |  
 | 4 | -5 | 1 | 4 |

**Анализ результатов**



