**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «БКИТ»

Отчет по лабораторной работе №5

«Модульное тестирование в Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35Б |  | преподаватель каф. ИУ5 |
| Большаков Георгий |  | Нардид А.Н. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

**Описание задания**

### 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).
   * Создание Mock-объектов (необязательное дополнительное задание).

**Текст программы**

**Файлы для BDD-тестирования**

**scenario.feature**

**Feature**: Testing  
**Scenario**: 2 roots  
**Given** nums  
**When** equation is solved  
**Then** roots are

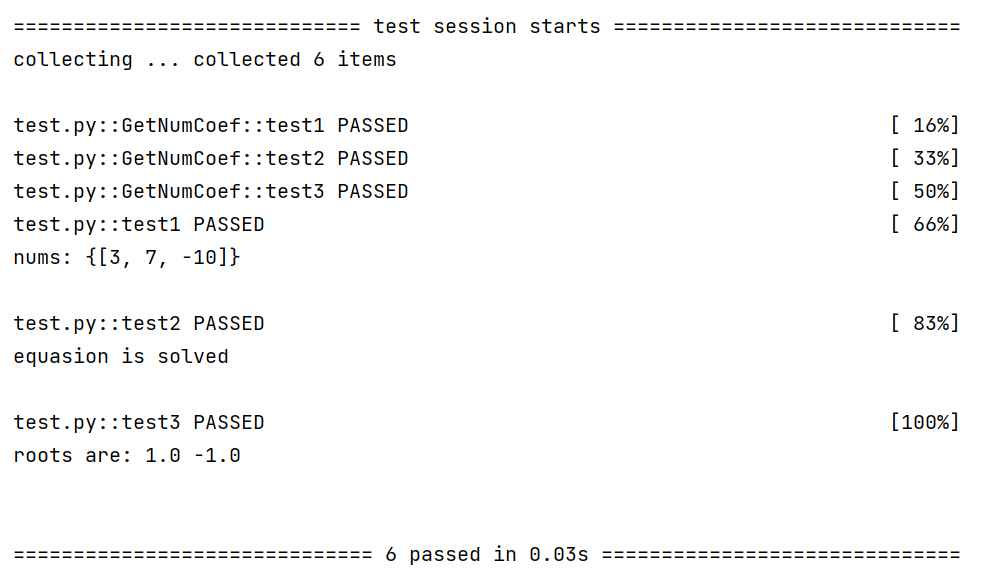
**Lab1.py**

**import** random  
**import** sys  
**import** math  
  
**def** get\_num(index, promt):  
 **try**:  
 num\_str = sys.argv[index]  
 **except**:  
 print(promt)  
 num\_str = input()  
  
 **if not** num\_str.isalpha():  
 num = float(num\_str)  
 **else**:  
 print("Коэффициент задан некорректно. Он будет задан рандомно от 1 до 10")  
 num = random.randint(1, 10)  
 print(num)  
 **return** num  
  
**def** get\_roots(a, b, c):  
 result = []  
 **if** a == 0:  
 print("коэффициент А не может быть равне нулю. он будет задан рандомно от 1 до 10")  
 a = random.randint(1, 10)  
 print(a)  
 D = b \* b - 4 \* a \* c  
 **if** D == 0.0:  
 t = -b / (2.0 \* a)  
 x1 = - math.sqrt(t)  
 x2 = math.sqrt(t)  
 result.append(x1)  
 **if** x1 != 0:  
 result.append(x2)  
 **elif** D > 0.0:  
 t1 = (- b + math.sqrt(D)) / (2.0 \* a)  
 **if** t1 > 0.0:  
 x1\_1 = math.sqrt(t1)  
 x1\_2 = - math.sqrt(t1)  
 result.append(x1\_1)  
 result.append(x1\_2)  
  
 t2 = (- b - math.sqrt(D)) / (2.0 \* a)  
 **if** t2 > 0.0:  
 x2\_1 = math.sqrt(t2)  
 x2\_2 = - math.sqrt(t2)  
 result.append(x2\_1)  
 result.append(x2\_2)  
  
 **return** result  
  
**def** main():  
 a = get\_num(1, "Введите коэффициент А:")  
 b = get\_num(2, "Введите коэффициент В:")  
 c = get\_num(3, "Введите коэффициент C:")  
  
 roots = get\_roots(a, b, c)  
 len\_roots = len(roots)  
 **if** len\_roots == 0:  
 print("Нет корней")  
 **elif** len\_roots == 1:  
 print("Один корень: {}".format(roots[0]))  
 **elif** len\_roots == 2:  
 print("Два корня: {} и {}".format(roots[0], roots[1]))  
 **elif** len\_roots == 4:  
 print("Четыре корня: {}; {}; {}; {};".format(roots[0], roots[1], roots[2], roots[3]))  
  
 # если сценарий запущен из командной строки  
**if** \_\_name\_\_ == "\_\_main\_\_":  
 main()

**test.py**

**import** unittest  
**import** pytest  
**from** pytest\_bdd **import** feature, scenario, given, when, then  
**from** main **import** get\_roots  
**from** unittest **import** TestCase  
  
  
**class** GetNumCoef(TestCase):  
 **def** test1(self):  
 self.assertEqual(get\_roots(3, 7, -10), [1.0, -1.0])  
 **def** test2(self):  
 self.assertEqual(get\_roots(1, 1, 1), [])  
 **def** test3(self):  
 self.assertEqual(get\_roots(8, -6, 1), [0.7071067811865476, -0.7071067811865476, 0.5, -0.5])  
  
  
@scenario("scenarios.feature", "2 roots")  
**def** test1():  
 print("**\n**Scenario: 2 roots")  
  
@given("nums")  
**def** test1():  
 print("**\n**nums: {[3, 7, -10]}")  
  
@when('equasion is solved')  
**def** test2():  
 print('**\n**equasion is solved')  
  
@then('roots are')  
**def** test3():  
 print('**\n**roots are:',1.0, -1.0)  
 **assert** get\_roots(3, 7, -10) == [1.0, -1.0]  
  
**def** main():  
 unittest.main()

**Экранные формы**

****