**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №5

«Модульное тестирование в Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| Студентка группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Коренева София |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

**Описание задания**

Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3–4.

Модифицируйте код таким образом, чтобы он был пригоден для модульного тестирования.

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**

Файл modify\_levenstain.py

def levenstain(word1, word2):  
 a = word1  
 b = word2  
 mat = []  
 st = []  
 for i in range(len(a) + 1):  
 for j in range(len(b) + 1):  
 st.append(0)  
 mat.append(st)  
 st = []  
  
 for i in range(len(b) + 1):  
 mat[0][i] = i  
  
 for i in range(len(a) + 1):  
 mat[i][0] = i  
  
 m = 1  
 for i in range(1, len(a) + 1):  
 for j in range(1, len(b) + 1):  
 if a[i - 1] == b[j - 1]:  
 m = 0  
 first = mat[i][j - 1] + 1  
 second = mat[i - 1][j] + 1  
 third = mat[i - 1][j - 1] + m  
 mat[i][j] = min(first, second, third)  
 m = 1  
  
 return mat[len(a-1)][len(b)]

Файл test1.py

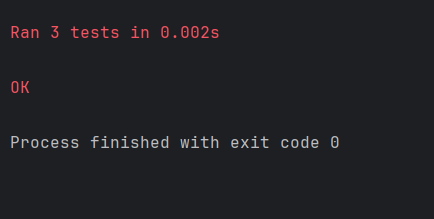
import unittest  
from modify\_levenstain import levenstain  
  
class LevenstainTest(unittest.TestCase):  
 def test\_empty(self):  
 self.assertEqual(levenstain("", "пустыня"), 7)  
 def test\_equal(self):  
 self.assertEqual(levenstain("пустыня", "пустыня"), 0)  
 def test\_differ (self):  
 self.assertEqual(levenstain("ложка","лодка"), 1)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

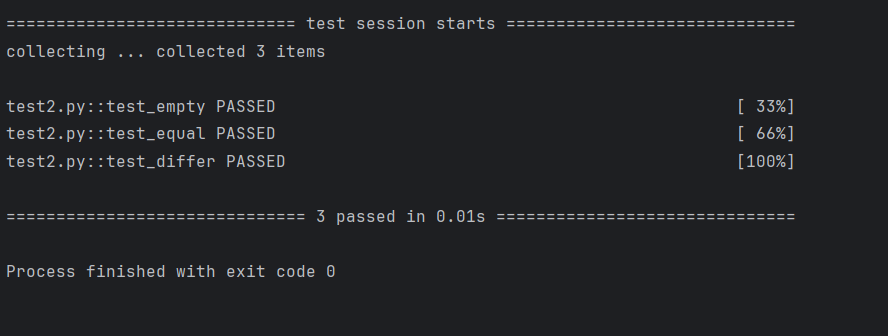
Файл test2.py

from modify\_levenstain import levenstain  
  
import pytest  
def test\_empty():  
 assert levenstain("", "пустыня") == 7  
def test\_equal():  
 assert levenstain("пустыня", "пустыня") == 0  
def test\_differ():  
 assert levenstain("ложка","лодка") == 1  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 pytest.main()

**Примеры выполнения программы**

При корректном тестировании





При возникновении ошибки

