

**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:
Студентка группы ИУ5-33Б
Коренева София
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.
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

```
Ran 3 tests in 0.002s
```

```
OK
```

```
Process finished with exit code 0
```

```
===== test session starts =====  
collecting ... collected 3 items  
  
test2.py::test_empty PASSED [ 33%]  
test2.py::test_equal PASSED [ 66%]  
test2.py::test_differ PASSED [100%]  
  
===== 3 passed in 0.01s =====  
  
Process finished with exit code 0
```

При возникновении ошибки

```
Ran 3 tests in 0.004s  
  
FAILED (errors=3)  
  
Process finished with exit code 1
```

```
===== 3 failed in 0.06s =====  
  
Process finished with exit code 1
```
