

**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 №4
«Модульное тестирование в Python»**

Выполнил:
Студент группы ИУ5-33Б
Лупарев Сергей
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 Lev_2.py(модифицированный код для нахождения расстояния Левенштейна)

```
def lev(str1, str2):
    L = len(str1) + 1
    L2 = len(str2) + 1
    mass = [0]*L2
    mass[0] = range(L)
    for i in range(1, L2):
        mass[i] = [0]*L
        mass[i][0] = i
    for i in range(1, L2):
        for j in range(1, L):
            mass[i][j] = min(mass[i][j-1] + 1,
                             mass[i-1][j] + 1,
                             mass[i-1][j-1] + (str1[j-1] != str2[i-1]))
    return mass[L2-1][L-1]
```

Файл test1.py

```
import unittest
from Lev_2 import lev

class TestLevenshteinDistance(unittest.TestCase):
    def test_equal(self):
        self.assertEqual(lev("ГОРОД", "ГОРОД"), 0)

    def test_diff(self):
        self.assertEqual(lev("ГОРОД", "ГОРЫ"), 2)

    def test_empty(self):
        self.assertEqual(lev("", "nkvd"), 4)

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

Файл test2.py

```
from Lev_2 import lev

import pytest

def test_equal():
    assert lev("ДОРОГА", "ДОРОГА") == 0

def test_diff():
    assert lev("АЧХ", "УПЧХИ") == 3

def test_empty():
    assert lev("", "gspd") == 4

if __name__ == '__main__':
    pytest.main()
```

Примеры выполнения программы

При корректном тестировании

```
===== RESTART: D:\CBeer\Labs_PICKUP\lab4\test1.py =====
...
-----
Ran 3 tests in 0.006s

OK
>>> |
```

```
===== test session starts =====
platform win32 -- Python 3.7.2, pytest-7.4.3, pluggy-1.2.0
rootdir: D:\CBeer\test_2_3\test_2_3
plugins: web3-5.28.0
collected 3 items

test_2_3.py ... [100%]

===== 3 passed in 0.14s =====
Press any key to continue . . .
```

При возникновении ошибки

```
===== test session starts =====
platform win32 -- Python 3.7.2, pytest-7.4.3, pluggy-1.2.0
rootdir: D:\CBeer\test_2_3\test_2_3
plugins: web3-5.28.0
collected 3 items

test_2_3.py ..F [100%]

===== FAILURES =====
_____ test_one_empty_string _____

    def test_one_empty_string():
>         assert lev("", "gspd") == 3
E         AssertionError: assert 4 == 3
E         + where 4 = lev('', 'gspd')

test_2_3.py:25: AssertionError

===== short test summary info =====
FAILED test_2_3.py::test_one_empty_string - AssertionError: assert 4 == 3
===== 1 failed, 2 passed in 0.14s =====
Press any key to continue . . .
```