|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ПРАКТИЧЕСКАЯ РАБОТА №2**

по дисциплине

«Тестирование и верификация программного обеспечения»

**Студент группы** ИКБО-02-20

Попов Александр Юрьевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Принял руководитель работы** преподаватель Петренко А. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Практические работы выполнены «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Зачтено «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

1. Документация проекта

Функция «format» способна анализировать строки, содержащей данные в специальном текстовом формате ("<sect> <% store 'название подсписка' is list( подсписок) %>”) и преобразует результат в виде словаря.

1. Документация чужого проекта

Функция «copy\_handler» способна обновлять индекс дубликата имени файла или добавлять при его отсутствии. Также есть поддержка имен файлов с двойными расширениями такие как «.tar.gz».

1. Тесты на чужой проект

# Тест

@pytest.mark.parametrize('name,res', [

    ('subfolder/1.txt',         'subfolder/1 (2).txt'),

    ('subfolder/1 (2).txt',     'subfolder/1 (3).txt'),

    ('fold/.FTA/1.txt',         'fold/.FTA/1 (2).txt'),

    ('fold/.FTA/2 (1).txt',     'fold/.FTA/2 (2).txt'),

    ('fold/.FTA/1 (4).txt',     'fold/.FTA/1 (5).txt'),

    ('1.txt',                   '1 (2).txt'),

    ('1 (2).txt',               '1 (3).txt'),

    ('1 (2).tar.gz',            '1 (3).tar.gz'),

    ('1 (10).tar.gz',           '1 (11).tar.gz'),

])

def test\_copy\_handler(name, res):

    # Проверка переименований копий

    assert copy\_handler(name) == res

# Функция

def copy\_handler(filename) -> str:

    """ Обновление индекса одинаковых имен """

    name, ext = os.path.splitext(filename)

    # Исправления tar.gz

    part = os.path.splitext(name)[1]

    while part != '':

        name = name[:len(name)-len(part)]

        ext = part + ext

        part = os.path.splitext(name)[1]

    # Проверка существующего индекса

    index = re.findall('\s\(\d+\)$', name)

    if index:

        num = int(re.findall('\d+', index[0])[0])

        name = name[:len(name)-len(index[0])] + f' ({str(num+1)})'

    # Не существует

    else:

        name = name + ' (2)'

    return name + ext

1. Оценки/Замечания

Функция работает правильно и ошибки не были обнаружены.

1. Выводы

Я научился создавать модульные тесты и с их помощью находить ошибки в коде.