Московский Государственный Технический Университет им. Н.Э. Баумана

Отчет по лабораторной работе № 5 по курсу Базовые компоненты интернет-технологий

" Модульное тестирование в Python."

Исполнитель:

Фруктин А.Е. РТ5-31Б

Проверил:

Гапанюк Юрий Евгеньевич

Москва, 2022

**Задание**

Цель лабораторной работы: изучение возможностей модульного тестирования в языке Python.

Задание:

Выбрать любой фрагмент кода из предыдущих лабораторных работ (выбранная работа: 3-4 лабораторная).

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

Разработать модульные тесты с помощью TDD и BDD фреймворков.

**Код**

**test\_field.py:**

from features.steps.lab3\_4.field import field

def test\_field():

    students = [

        {'name': 'Иван', 'age': 19, 'scores': [4, 5, 5, 5, 2, 3]},

        {'name': 'Мария', 'age': 17, 'scores': [3, 2, 3, 3, 3, 4]},

        {'name': 'Петр', 'age': 21, 'scores': [5, 5, 4, 4, 5, 5]},

        {'name': 'Илья', 'age': 18, 'scores': [3, 3, 3, 3, 2, 2]},

    ]

    assert field(students, 'name') == ['Иван', 'Мария', 'Петр', 'Илья']

**test\_unique.py:**

from features.steps.lab3\_4.unique import Unique

def test\_sort\_ignore\_case\_true():

    data = ['a', 'A', 'B', 'a', 'b']

    assert list(Unique(data, ignore\_case=True)) == ['a', 'b']

def test\_sort\_ignore\_case\_false():

    data = ['a', 'A', 'B', 'a', 'b']

    assert list(Unique(data, ignore\_case=False)) == ['a', 'A', 'B', 'b']

**first.feature:**

Feature: show behave

    Scenario: Test

        Given a list of people

        When I get the names

        Then I should get a list of names

        Given a list of str data

        When I call Unique with key ignore\_case=True

        Then I get list of unique str data without big letters

        Given a list of str data2

        When I call Unique with key ignore\_case=False

        Then I get list of unique str data without all letters

**bdd\_st.py:**

from lab3\_4.field import field

from lab3\_4.unique import Unique

from behave import given, then, when

@given('a list of people')

def step\_field(context):

    context.list\_of\_people = [

        {'name': 'Иван', 'age': 19, 'scores': [4, 5, 5, 5, 2, 3]},

        {'name': 'Мария', 'age': 17, 'scores': [3, 2, 3, 3, 3, 4]},

        {'name': 'Петр', 'age': 21, 'scores': [5, 5, 4, 4, 5, 5]},

        {'name': 'Илья', 'age': 18, 'scores': [3, 3, 3, 3, 2, 2]},

    ]

@when('I get the names')

def step\_field(context):

    context.names = field(context.list\_of\_people, 'name')

@then('I should get a list of names')

def step\_field(context):

    assert context.names == ['Иван', 'Мария', 'Петр', 'Илья']

@given('a list of str data')

def step\_field(context):

    context.data = ['a', 'A', 'B', 'a', 'b']

@when('I call Unique with key ignore\_case=True')

def step\_field(context):

    context.unique\_data = list(Unique(context.data, ignore\_case=True))

@then('I get list of unique str data without big letters')

def step\_field(context):

    assert context.unique\_data == ['a', 'b']

@given('a list of str data2')

def step\_field(context):

    context.data = ['a', 'A', 'B', 'a', 'b']

@when('I call Unique with key ignore\_case=False')

def step\_field(context):

    context.unique\_data = list(Unique(context.data, ignore\_case=False))

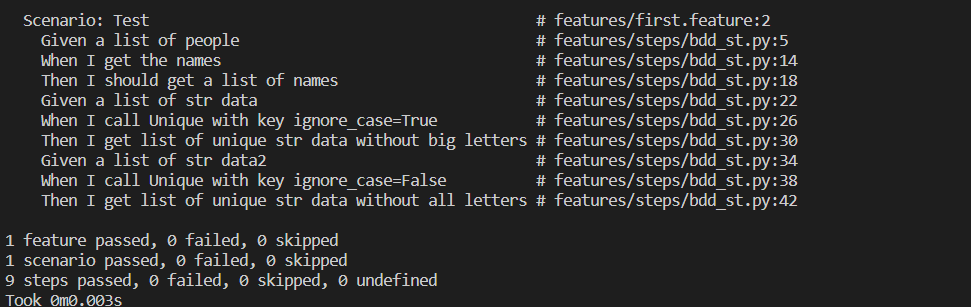
@then('I get list of unique str data without all letters')

def step\_field(context):

    assert context.unique\_data == ['a', 'A', 'B', 'b']

**Результаты выполнения программы**

Bdd:



Tdd:

