

РК2

Условия рубежного контроля №2 по курсу ПИК ЯП

Рубежный контроль представляет собой разработку тестов на языке Python.

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Код:

Test123.py

```
from operator import itemgetter
from .database import Database

def task1(data: "Database"):
    searched_char = '3'
    res_11 = [el for el in data.one_to_many if el[0][0] == searched_char]
    return res_11

def task2(data: "Database"):
    res_12_unsorted = [[el.name, 1000000] for el in data.base]
    for t in res_12_unsorted:
        t[1] = min(map(lambda i: i[1], list(filter(lambda i: i[2] == t[0],
data.one_to_many))))
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    return res_12

def task3(data: "Database"):
    res_13 = sorted(data.many_to_many, key=itemgetter(1), reverse=True)
    return res_13
```

dataclasses.py

```
class Table():
    def __init__(self, id, name, base_id, kol):
        self.id = id
        self.name = name
        self.base_id = base_id
        self.kol = kol

class Base():
    def __init__(self, id, name):
        self.id = id
        self.name = name

class TableBase():
    def __init__(self, base_id, table_id):
        self.base_id = base_id
        self.table_id = table_id
```

database.py

```
from .dataclasses import Base, Table, TableBase

class Database():
    base: list[Base] = [
        Base(1, "Учебное заведение"),
        Base(2, "Магазин"),
        Base(3, "Социальная сеть"),
        Base(4, "Проект управления")
    ]
    table: list[Table] = [
        Table(1, "Студенты", 1, 5),
        Table(2, "Курсы", 3, 4),
        Table(3, "Преподаватели", 2, 6),
        Table(4, "Товары", 2, 7),
        Table(5, "Заказы", 1, 5),
        Table(6, "Клиенты", 4, 3),
        Table(7, "Пользователи", 1, 8),
        Table(8, "Сообщения", 1, 4),
        Table(9, "Друзья", 3, 2),
        Table(10, "Проекты", 1, 6),
        Table(11, "Задачи", 2, 5),
        Table(12, "Сотрудники", 3, 4)
    ]
    table_base: list[TableBase] = [
        TableBase(1, 3),
        TableBase(1, 9),
        TableBase(1, 5),
        TableBase(1, 11),
        TableBase(1, 7),
        TableBase(2, 2),
        TableBase(2, 8),
        TableBase(2, 4),
        TableBase(2, 6),
        TableBase(2, 10),
        TableBase(2, 12),
        TableBase(3, 10),
        TableBase(3, 1),
        TableBase(3, 5),
        TableBase(3, 7),
        TableBase(3, 3),
        TableBase(4, 12),
        TableBase(4, 6),
        TableBase(4, 9),
        TableBase(4, 3)
    ]

    def __init__(self):
        # Соединение данных один-ко-многим
        self.one_to_many = [(t.name, t.kol, b.name)
                             for b in self.base
                             for t in self.table
                             if t.base_id == b.id]

        # Соединение данных многие-ко-многим
        self.many_to_many_temp = [(b.name, tb.table_id)
                                    for b in self.base
                                    for tb in self.table_base if tb.base_id == b.id]

        self.many_to_many = [(base_name, t.name, t.kol)
                              for base_name, table_id in self.many_to_many_temp
                              for t in self.table if t.id == table_id]
```

conftest.py

```
import pytest

from tasks.database import Database

@pytest.fixture(scope='session')
def base():
    return Database()
```

test_task1.py

```
from tasks.task123 import task1
from tests.conftest import base

class TestTask1:
    def test_task1_work(self, base):
        out = task1(base)
        assert len(out) == 2
        assert out[0] == ('Заказы', 5, 'Учебное заведение')
        assert out[1] == ('Задачи', 5, 'Магазин')
```

test_task1.py

```
from tasks.task123 import task2
from tests.conftest import base

class TestTask2:
    def test_task2_work(self, base):
        out = task2(base)
        assert len(out) == 4
        assert out == [
            ['Магазин', 5],
            ['Учебное заведение', 4],
            ['Проект управления', 3],
            ['Социальная сеть', 2],
        ]
```

test_task1.py

```
from tasks.task123 import task3
from tests.conftest import base

class TestTask3:
    def test_task3_work(self, base):
        out = task3(base)
        assert len(out) == 20
        assert out == [
            ('Магазин', 'Товары', 7),
            ('Социальная сеть', 'Студенты', 5),
            ('Магазин', 'Сотрудники', 4),
            ('Проект управления', 'Сотрудники', 4),
            ('Магазин', 'Сообщения', 4),
            ('Магазин', 'Проекты', 6),
            ('Социальная сеть', 'Проекты', 6),
            ('Учебное заведение', 'Преподаватели', 6),
            ('Социальная сеть', 'Преподаватели', 6),
            ('Проект управления', 'Преподаватели', 6),
            ('Учебное заведение', 'Пользователи', 8),
            ('Социальная сеть', 'Пользователи', 8),
            ('Магазин', 'Курсы', 4),
            ('Магазин', 'Клиенты', 3),
            ('Проект управления', 'Клиенты', 3),
            ('Учебное заведение', 'Заказы', 5),
            ('Социальная сеть', 'Заказы', 5),
```

```
        ('Учебное заведение', 'Задачи', 5),  
        ('Учебное заведение', 'Друзья', 2),  
        ('Проект управления', 'Друзья', 2),  
    ]
```

Результат:

```
===== 3 passed in 0.09s =====  
(.venv) PS C:\Users\kiril\Files\PycharmProjects\Labs\git_dir\RK2_318> pytest  
===== test session starts =====  
platform win32 -- Python 3.12.7, pytest-8.3.4, pluggy-1.5.0  
rootdir: C:\Users\kiril\Files\PycharmProjects\Labs\git_dir\RK2_318  
collected 3 items  
  
tests\test_tasks\test_task1.py . [ 33%]  
tests\test_tasks\test_task2.py . [ 66%]  
tests\test_tasks\test_task3.py . [100%]  
  
===== 3 passed in 0.12s =====
```