Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчёт по рубежному контролю №2

Выполнил:
студент группы ИУ5-32Б
Еремихин Владислав
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2021 г.

Описание задания

(Вариант предметной области - 5, вариант запросов - Б)

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

Текст программы

Файл **mus_orch.py**:

```
from operator import itemgetter


class Musician:
    """Музыкант"""

    def __init__(self, id, fio, salary, orch_id):
        self.id = id
        self.fio = fio
        self.salary = salary
        self.orch_id = orch_id


class Orchestra:
    """Оркестр"""

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


class MusOrch:
    """
    'Музыканты оркестра' для реализации
    связи многие-ко-многим
    """

    def __init__(self, orch_id, mus_id):
        self.orch_id = orch_id
        self.mus_id = mus_id


def task_1(orchestras, musicians):
    # Соединение данных один-ко-многим
    one_to_many = [
```

```
        (m.fio, m.salary, o.name)
        for o in orchestras
        for m in musicians
        if m.orch_id == o.id
    ]
    return sorted(one_to_many, key=itemgetter(0))


def task_2(orchestras, musicians):
    one_to_many = [
        (m.fio, m.salary, o.name)
        for o in orchestras
        for m in musicians
        if m.orch_id == o.id
    ]
    res = list()
    # Перебираем все оркестры
    for o in orchestras:
        # Список музыкантов в оркестре
        o_mus = list(filter(lambda i: i[2] == o.name, one_to_many))
        # Если оркестр не пустой
        if len(o_mus) > 0:
            res.append((o.name, len(o_mus)))
    return sorted(res, key=itemgetter(1), reverse=True)


def task_3(orchestras, musicians, mus_orch):
    many_to_many = [
        (m.fio, m.salary, o.name)
        for o in orchestras
        for m in musicians
        for relation in mus_orch
        if o.id == relation.orch_id and m.id == relation.mus_id
    ]
    res = dict()
    for m in musicians:
        if m.fio.endswith("ов"):
            # Ищем оркестры конкретного музыканта
            m_orchs = list(filter(lambda x: x[0] == m.fio, many_to_many))
            # Получаем их названия
            m_orchs_names = [x[2] for x in m_orchs]
            res[m.fio] = m_orchs_names
    return res


def main():
    # Оркестры
    orchestras = [
        Orchestra(1, "Electric Light Orchestra"),
        Orchestra(2, "Mariinsky Theatre Orchestra"),
        Orchestra(3, "Moscow Chamber Orchestra"),
```

```
        Orchestra(4, "Ukrainian Radio Symphony Orchestra"),
        Orchestra(5, "National Symphony Orchestra of Ukraine"),
    ]

    # Музыканты
    musicians = [
        Musician(1, "Линн", 70000, 1),
        Musician(2, "Тэнди", 67000, 1),
        Musician(3, "Смирнов", 20000, 2),
        Musician(4, "Карпов", 32000, 3),
        Musician(5, "Прокопенко", 14000, 4),
        Musician(6, "Карчук", 12000, 5),
        Musician(7, "Соловьёва", 14000, 3),
    ]

    # Оркестры и музыканты
    mus_orch = [
        MusOrch(1, 1),
        MusOrch(1, 2),
        MusOrch(2, 3),
        MusOrch(3, 4),
        MusOrch(3, 7),
        MusOrch(4, 5),
        MusOrch(5, 6),
        MusOrch(1, 3),
        MusOrch(4, 1),
        MusOrch(2, 4),
        MusOrch(3, 3),
    ]

    print("Задание Б1")
    res_1 = task_1(orchestras, musicians)
    [print(el) for el in res_1]

    print("\nЗадание Б2")
    res_2 = task_2(orchestras, musicians)
    [print(el) for el in res_2]

    print("\nЗадание Б3")
    res_3 = task_3(orchestras, musicians, mus_orch)
    [print(k, v) for k, v in res_3.items()]


if __name__ == "__main__":
    main()
```

Файл **test_mus_orch.py**:

```
import unittest
from mus_orch import Musician, Orchestra, MusOrch, task_1, task_2, task_3
```

```python
class TestRK1Tasks(unittest.TestCase):

    def test_task_1(self):
        orchestras = [
            Orchestra(1, "Electric Light Orchestra"),
            Orchestra(2, "Mariinsky Theatre Orchestra"),
            Orchestra(3, "Moscow Chamber Orchestra"),
            Orchestra(4, "Ukrainian Radio Symphony Orchestra"),
            Orchestra(5, "National Symphony Orchestra of Ukraine"),
        ]
        musicians = [
            Musician(1, "Линн", 70000, 1),
            Musician(2, "Тэнди", 67000, 1),
            Musician(3, "Смирнов", 20000, 2),
            Musician(4, "Карпов", 32000, 3),
            Musician(5, "Прокопенко", 14000, 4),
            Musician(6, "Карчук", 12000, 5),
            Musician(7, "Соловьёва", 14000, 3),
        ]

        answer = [
            ('Карпов', 32000, 'Moscow Chamber Orchestra'),
            ('Карчук', 12000, 'National Symphony Orchestra of Ukraine'),
            ('Линн', 70000, 'Electric Light Orchestra'),
            ('Прокопенко', 14000, 'Ukrainian Radio Symphony Orchestra'),
            ('Смирнов', 20000, 'Mariinsky Theatre Orchestra'),
            ('Соловьёва', 14000, 'Moscow Chamber Orchestra'),
            ('Тэнди', 67000, 'Electric Light Orchestra')
        ]

        self.assertEqual(task_1(orchestras, musicians), answer)

    def test_task_2(self):
        orchestras = [
            Orchestra(1, "Electric Light Orchestra"),
            Orchestra(2, "Mariinsky Theatre Orchestra"),
            Orchestra(3, "Moscow Chamber Orchestra"),
            Orchestra(4, "Ukrainian Radio Symphony Orchestra"),
            Orchestra(5, "National Symphony Orchestra of Ukraine"),
        ]
        musicians = [
            Musician(1, "Линн", 70000, 1),
            Musician(2, "Тэнди", 67000, 1),
            Musician(3, "Смирнов", 20000, 2),
            Musician(4, "Карпов", 32000, 3),
            Musician(5, "Прокопенко", 14000, 4),
            Musician(6, "Карчук", 12000, 5),
            Musician(7, "Соловьёва", 14000, 3),
        ]
```

```
        answer = [
            ('Electric Light Orchestra', 2),
            ('Moscow Chamber Orchestra', 2),
            ('Mariinsky Theatre Orchestra', 1),
            ('Ukrainian Radio Symphony Orchestra', 1),
            ('National Symphony Orchestra of Ukraine', 1)
        ]

        self.assertEqual(task_2(orchestras, musicians), answer)

    def test_task_3(self):
        orchestras = [
            Orchestra(1, "Electric Light Orchestra"),
            Orchestra(2, "Mariinsky Theatre Orchestra"),
            Orchestra(3, "Moscow Chamber Orchestra"),
            Orchestra(4, "Ukrainian Radio Symphony Orchestra"),
            Orchestra(5, "National Symphony Orchestra of Ukraine"),
        ]
        musicians = [
            Musician(1, "Линн", 70000, 1),
            Musician(2, "Тэнди", 67000, 1),
            Musician(3, "Смирнов", 20000, 2),
            Musician(4, "Карпов", 32000, 3),
            Musician(5, "Прокопенко", 14000, 4),
            Musician(6, "Карчук", 12000, 5),
            Musician(7, "Соловьёва", 14000, 3),
        ]
        mus_orch = [
            MusOrch(1, 1),
            MusOrch(1, 2),
            MusOrch(2, 3),
            MusOrch(3, 4),
            MusOrch(3, 7),
            MusOrch(4, 5),
            MusOrch(5, 6),
            MusOrch(1, 3),
            MusOrch(4, 1),
            MusOrch(2, 4),
            MusOrch(3, 3)
        ]

        answer = {
            'Смирнов': ['Electric Light Orchestra', 'Mariinsky Theatre Orchestra',
'Moscow Chamber Orchestra'],
            'Карпов': ['Mariinsky Theatre Orchestra', 'Moscow Chamber Orchestra']
        }

        self.assertEqual(task_3(orchestras, musicians, mus_orch), answer)
```

```python
if __name__ == "__main__":
    unittest.main()
```

Результат выполнения программы

```
Microsoft Windows [Version 6.3.9600]
(c) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защище

C:\Users\pvrtss>cd /d E:\st\bkit\rk\rk2

E:\st\bkit\rk\rk2>python -m unittest
...
----------------------------------------------------------------------
Ran 3 tests in 0.009s

OK

E:\st\bkit\rk\rk2>_
```