

Московский государственный технический университет

имени Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Отчёт по рубежному контролю №2

Выполнила:

Рыжкова Юлия Николаевна

Группа ИУ5-31Б

Проверил:

Гапанюк Юрий Евгеньевич

Кафедра ИУ5

Москва 2021г.

Задание

- 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).

Текст программы

Файл VKIT_RK.py

```
# используется для сортировки
from operator import itemgetter

class Conductor:
    """Дирижёр"""
    def __init__(self, id, fio):
        self.id = id
        self.fio = fio

class Orchestra:
    """Оркестр"""
    def __init__(self, id, name, num, con_id):
        self.id = id
        self.name = name
        self.num = num #количество музыкантов в оркестре
        self.con_id = con_id

class ConOrc:
    """
    'Дирижёр оркестра' для реализации связи
    многие-ко-многим
    """
    def __init__(self, con_id, orc_id):
        self.con_id = con_id
        self.orc_id = orc_id

# Оркестры
orcs = [
    Orchestra(1, 'оркестр имени Осипова', 50, 3),
    Orchestra(2, 'симфонический оркестр Московской
    филармонии', 33, 5),
    Orchestra(3, 'военный оркестр', 118, 2),
    Orchestra(11, 'сводный духовой оркестр', 24, 1),
    Orchestra(22, 'Смуглянка', 12, 5),
    Orchestra(33, 'Гленн Миллер', 68, 4),
]

# Дирижёры
cons = [
    Conductor(1, 'Иванов'),
    Conductor(2, 'Петров'),
    Conductor(3, 'Смирнов'),
```

```
Conductor(4, 'Спичка'),  
Conductor(5, 'Артемьев'),  
]
```

```
con_orcs = [  
    ConOrc(3,1),  
    ConOrc(4,1),  
    ConOrc(5,2),  
    ConOrc(2,3),  
    ConOrc(3,3),  
    ConOrc(5,3),  
    ConOrc(1,11),  
    ConOrc(5,22),  
    ConOrc(1,33),  
    ConOrc(4,33),  
    ConOrc(5,33),  
]
```

```
def taskA1(one_to_many):  
    """  
    «Дирижёр» и «Оркестр» связаны соотношением  
    один-ко-многим.  
    Выведите список всех связанных дирижёров и оркестров,  
    отсортированный по оркестрам,  
    сортировка по дирижёрам произвольная.  
    """  
    return sorted(one_to_many, key=lambda x:  
str(x[1]).lower())
```

```
def taskA2(one_to_many):  
    """  
    «Дирижёр» и «Оркестр» связаны соотношением  
    один-ко-многим.  
    Выведите список дирижёров с суммарным количеством  
    музыкантов в каждом оркестре,  
    отсортированный по количеству музыкантов.  
    """  
    res_unsorted = []  
    # Перебираем всех дирижёров  
    for c in cons:  
        # Список оркестров дирижёра  
        c_orcs = list(filter(lambda i: i[0]==c.fio,  
one_to_many))  
        # Если список не пустой  
        if len(c_orcs) > 0:  
            # Количество музыкантов оркестра  
            o_nums = [num[2] for num in c_orcs]
```

```

        # Суммарное количество музыкантов оркестра
        o_nums_sum = sum(o_nums)
        res_unsorted.append((c.fio, o_nums_sum))
    # Сортировка по количеству музыкантов
    return sorted(res_unsorted, key=itemgetter(1),
reverse=True)

```

```

def taskA3(many_to_many):
    """
    «Дирижёр» и «Оркестр» связаны соотношением
    многие-ко-многим.
    Выведите список всех оркестров, у которых в названии
    присутствует слово «оркестр»,
    и список работающих в них дирижёров.
    """
    res = {}
    # Перебираем все оркестры
    for o in orcs:
        if 'оркестр' in o.name:
            # Список дирижёров оркестра
            o_cons = list(filter(lambda i: i[0] == o.name,
many_to_many))
            # Только ФИО дирижёров
            o_cons_names = [x[2] for x in o_cons]
            # Добавляем результат в словарь
            # ключ - оркестр, значение - список фамилий
            res[o.name] = o_cons_names
    return res

```

```

def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
    one_to_many = [(c.fio, o.name, o.num)
        for o in orcs
        for c in cons
        if o.con_id==c.id]
    # Соединение данных многие-ко-многим
    many_to_many_temp = [(c.fio, co.orc_id, co.con_id)
        for c in cons
        for co in con_orcs
        if c.id==co.con_id]
    many_to_many = [(o.name, o.num, con_name)
        for con_name, orc_id, con_id in many_to_many_temp
        for o in orcs
        if o.id==orc_id]
    print('Задание A1')
    print(taskA1(one_to_many))

```

```

print('\nЗадание A2')
print(taskA2(one_to_many))
print('\nЗадание A3')
print(taskA3(many_to_many))
if __name__ == '__main__':
    main()

```

Файл tests.py

```

import unittest
from BKIT_RK import *
class Tests(unittest.TestCase):
    def setUp(self):
        self.orcs = [
            Orchestra(1, 'оркестр имени Осипова', 50,
3),
            Orchestra(2, 'симфонический оркестр
Московской филармонии', 33, 5),
            Orchestra(3, 'военный оркестр', 118, 2),
            Orchestra(11, 'сводный духовой оркестр',
24, 1),
            Orchestra(22, 'Смуглянка', 12, 5),
            Orchestra(33, 'Гленн Миллер', 68, 4),
        ]
        self.cons = [
            Conductor(1, 'Иванов'),
            Conductor(2, 'Петров'),
            Conductor(3, 'Смирнов'),
            Conductor(4, 'Спичка'),
            Conductor(5, 'Артемьев'),
        ]
        self.con_orcs = [
            ConOrc(3,1),
            ConOrc(4,1),
            ConOrc(5,2),
            ConOrc(2,3),
            ConOrc(3,3),
            ConOrc(5,3),
            ConOrc(1,11),
            ConOrc(5,22),
            ConOrc(1,33),
            ConOrc(4,33),
            ConOrc(5,33),
        ]

```

```

        self.one_to_many = [(c.fio, o.name, o.num)
                             for o in self.orcs
                             for c in self.cons
                             if o.con_id == c.id]
        self.many_to_many_temp = [(c.fio, co.orc_id,
co.con_id)
                                for c in self.cons
                                for co in self.con_orcs
                                if c.id == co.con_id]
        self.many_to_many = [(o.name, o.num,
con_name)
                              for con_name, orc_id, con_id
in self.many_to_many_temp
                              for o in self.orcs
                              if o.id == orc_id]

```

```

def test_taskA1(self):
    expected = [('Петров', 'военный оркестр',
118), ('Спичка', 'Гленн Миллер', 68), ('Смирнов',
'оркестр имени Осипова', 50), ('Иванов', 'сводный
духовой оркестр', 24), ('Артемьев', 'симфонический
оркестр Московской филармонии', 33), ('Артемьев',
'Смутлянка', 12)]
    self.assertEqual(expected,
taskA1(self.one_to_many))

```

```

def test_taskA2(self):
    expected = [('Петров', 118), ('Спичка', 68),
('Смирнов', 50), ('Артемьев', 45), ('Иванов', 24)]
    self.assertEqual(expected,
taskA2(self.one_to_many))

```

```

def test_taskA3(self):
    expected = {'оркестр имени Осипова':
['Смирнов', 'Спичка'], 'симфонический оркестр
Московской филармонии': ['Артемьев'], 'военный
оркестр': ['Петров', 'Смирнов', 'Артемьев'],
'сводный духовой оркестр': ['Иванов']}
    self.assertEqual(expected,
taskA3(self.many_to_many))

```

```

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

```

Результаты выполнения программы

```
Run: BKIT_RK
/Users/artisia/PycharmProjects/Lab_python_RK1/venv/bin/python /Users/artisia/PycharmProjects/Lab_python_RK1/BKIT_RK.py

Задание A1
[('Петров', 'военный оркестр', 118), ('Спичка', 'Гленн Миллер', 68), ('Смирнов', 'оркестр имени Осипова', 50), ('Иванов', 'сводный духовой оркестр', 24), ('Артемьев', 'симфонический оркестр Московской филармонии', 33), ('Артемьев', 'Смуглянка', 12)]

Задание A2
[('Петров', 118), ('Спичка', 68), ('Смирнов', 50), ('Артемьев', 45), ('Иванов', 24)]

Задание A3
{'оркестр имени Осипова': ['Смирнов', 'Спичка'], 'симфонический оркестр Московской филармонии': ['Артемьев'], 'военный оркестр': ['Петров', 'Смирнов', 'Артемьев'], 'сводный духовой оркестр': ['Иванов']}

Process finished with exit code 0
```

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for running, debugging, and testing. The 'Test Results' window is open, showing the command: `./Users/artisia/PycharmProjects/lab_python_RK1/venv/bin/python "/Applications/PyCharm CE.app/Contents/plugins/python-ce/helpers/pycharm/_jb_unittest_runner.py" --path /Users/artisia/PycharmProjects/lab_python_RK1/tests.py`. The output shows 'Testing started at 17:53 ...', 'Launching unittests with arguments python -m unittest /Users/artisia/PycharmProjects/lab_python_RK1/tests.py in /Users/artisia/PycharmProjects/lab_python_RK1', 'Ran 3 tests in 0.007s', and 'OK'. The status bar at the bottom indicates 'Python 3.8 (lab_python_RK1)'.