

**Московский государственный технический  
университет им. Н.Э. Баумана**

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

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК №2  
Вариант запросов: В  
Вариант предметной области: 17

Выполнил:  
Студент группы ИУ5-33Б  
Лупарев Сергей

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю. Е.

Москва, 2023 г.

## Полученное задание

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).

### Файл main.py

```
from operator import itemgetter

class Director:
    def __init__(self, id_, fio_, orchestra_id_, salary_):
        self.id = id_
        self.fio = fio_
        self.orchestra_id = orchestra_id_
        self.salary = salary_

class Orchestra:
    def __init__(self, id_, name_):
        self.id = id_
        self.name = name_

class OrcDir:
    def __init__(self, orchestra_id_, director_id_):
        self.orchestra_id = orchestra_id_
        self.director_id = director_id_

def task1(mass):
    res = []
    for i in mass:
        if i[2].startswith('A'):
            res.append([i[2], i[3]])
            #print('ФИО: {}, Оркестр: {}'.format(i[2], i[3]))
    return res

def task2(mass):
    mass_help = dict()
    res = []
    for i in mass:
        if i[3] not in mass_help:
            mass_help[i[3]] = i[4]
        else:
            mass_help[i[3]] = min(mass_help[i[3]], i[4])
    mass_help = sorted(mass_help.items(), key = lambda item: item[1])
    for i in mass_help:
        res.append([i[0], i[1]])
        #print('Оркестр: {}, Минимальная зарплата: {}'.format(i[0], i[1]))
    return res
```

```

def task3(mass):
    res = []
    mass = sorted(mass, key = itemgetter(2))
    for i in mass:
        #print('ФИО: {}, Оркестр: {}'.format(i[2], i[3]))
        res.append([i[2], i[3]])
    return res

def data_generate():
    orchestras = [
        Orchestra(1, "Первый"),
        Orchestra(2, "Ленина"),
        Orchestra(3, "Октябрьский"),
        Orchestra(4, "Симфонический"),
        Orchestra(5, "Оркестр имени Некрасова")]

    directors = [
        Director(1, "Иванов Сергей Викторович", 1, 245000),
        Director(2, "Смирнов Александр Игоревич", 1, 372500),
        Director(3, "Абрамов Дмитрий Алексеевич", 1, 198000),
        Director(4, "Петров Андрей Сергеевич", 2, 415000),
        Director(5, "Соколов Владимир Николаевич", 2, 286000),
        Director(6, "Михайлов Иван Васильевич", 2, 595000),
        Director(7, "Анатолийев Алексей Александрович", 3, 162000),
        Director(8, "Абакян Денис Павлович", 3, 329000),
        Director(9, "Волков Александр Владимирович", 3, 443000),
        Director(10, "Козлов Дмитрий Иванович", 4, 217500),
        Director(11, "Николаев Андрей Петрович", 4, 525000),
        Director(12, "Морозов Максим Сергеевич", 4, 342000),
        Director(13, "Некрасов Иван Олегович", 5, 267500),
        Director(14, "Андреев Артем Николаевич", 5, 383000),
        Director(15, "Семёнов Антон Александрович", 5, 295000)]

```

```

connections = [
    OrcDir(1, 1),
    OrcDir(1, 2),
    OrcDir(1, 3),
    OrcDir(2, 4),
    OrcDir(2, 5),
    OrcDir(2, 6),
    OrcDir(3, 7),
    OrcDir(3, 8),
    OrcDir(3, 9),
    OrcDir(4, 10),
    OrcDir(4, 11),
    OrcDir(4, 12),
    OrcDir(5, 13),
    OrcDir(5, 14),
    OrcDir(5, 15)]
return orchestras, directors, connections

def main():
    orchestras, directors, connections = data_generate()
    main_list = list()
    for i in directors:
        for j in orchestras:
            if i.orchestra_id == j.id:
                main_list.append((i.id, j.id, i.fio, j.name, i.salary))
    print(task1(main_list))
    print(task2(main_list))
    print(task3(main_list))

if __name__ == '__main__':
    main()

```

## Файл RK2.py

```
import unittest
from main import *

class TestDirector(unittest.TestCase):

    def test_director_creation(self):
        d = Director(1, "Иванов Сергей Викторович", 1, 245000)
        self.assertEqual(d.id, 1)
        self.assertEqual(d.fio, "Иванов Сергей Викторович")
        self.assertEqual(d.orchestra_id, 1)
        self.assertEqual(d.salary, 245000)

class TestOrchestra(unittest.TestCase):

    def test_orchestra_creation(self):
        o = Orchestra(1, "Первый")
        self.assertEqual(o.id, 1)
        self.assertEqual(o.name, "Первый")

class TestOrcDir(unittest.TestCase):

    def test_connection_creation(self):
        c = OrcDir(1, 1)
        self.assertEqual(c.orchestra_id, 1)
        self.assertEqual(c.director_id, 1)

class TestTaskExecution(unittest.TestCase):
    def setUp(self):
        self.orchestras, self.directors, self.connections = data_generate()
        self.main_list = list()
        for i in self.directors:
            for j in self.orchestras:
                if i.orchestra_id == j.id:
                    self.main_list.append((i.id, j.id, i.fio, j.name, i.salary))

    def test_task1(self):
        test_result = task1(self.main_list)
        goal_result = [
            ["Абрамов Дмитрий Алексеевич", "Первый"],
            ["Анатольев Алексей Александрович", "Октябрьский"],
            ["Абакян Денис Павлович", "Октябрьский"],
            ["Андреев Артем Николаевич", "Оркестр имени Некрасова"]
        ]
        self.assertEqual(test_result, goal_result)

    def test_task2(self):
        test_result = task2(self.main_list)
        goal_result = [
            ['Октябрьский', 162000],
            ['Первый', 198000],
            ['Симфонический', 217500],
            ['Оркестр имени Некрасова', 267500],
            ['Ленина', 286000]
        ]
        self.assertEqual(test_result, goal_result)

    def test_task3(self):
        test_result = task3(self.main_list)
        goal_result = [
            ['Абакян Денис Павлович', 'Октябрьский'],
            ['Абрамов Дмитрий Алексеевич', 'Первый'],
            ['Анатольев Алексей Александрович', 'Октябрьский'],
            ['Андреев Артем Николаевич', 'Оркестр имени Некрасова'],
            ['Волков Александр Владимирович', 'Октябрьский'],
            ['Иванов Сергей Викторович', 'Первый'],
            ['Козлов Дмитрий Иванович', 'Симфонический'],
            ['Михайлов Иван Васильевич', 'Ленина'],
            ['Морозов Максим Сергеевич', 'Симфонический'],
            ['Некрасов Иван Олегович', 'Оркестр имени Некрасова'],
            ['Николаев Андрей Петрович', 'Симфонический'],
            ['Петров Андрей Сергеевич', 'Ленина'],
            ['Семёнов Антон Александрович', 'Оркестр имени Некрасова'],
            ['Смирнов Александр Игоревич', 'Первый'],
            ['Соколов Владимир Николаевич', 'Ленина']
        ]
        self.assertEqual(test_result, goal_result)

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

## Результат выполнения

### В случае корректного выполнения

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\CBeer\Labs_PICKUP\RK2\RK2.py =====
.....
-----
Ran 6 tests in 0.026s

OK
>>> |
```

### В случае ошибки

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\CBeer\Labs_PICKUP\RK2\RK2.py =====
.....F
=====
FAIL: test_task3 (__main__.TestTaskExecution)
-----
Traceback (most recent call last):
  File "D:\CBeer\Labs_PICKUP\RK2\RK2.py", line 72, in test_task3
    self.assertEqual(test_result, goal_result)
AssertionError: Lists differ: [['Абакян Денис Павлович', 'Октябрьский'], [[653 с
hars]на']] != [['Обакян Денис Павлович', 'Октябрьский'], [[653 chars]на']]

First differing element 0:
['Абакян Денис Павлович', 'Октябрьский']
['Обакян Денис Павлович', 'Октябрьский']

Diff is 808 characters long. Set self.maxDiff to None to see it.
-----
Ran 6 tests in 0.011s

FAILED (failures=1)
>>> |
```