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»

Отчет по рубежному контролю №2

Выполнил:
студент группы ИУ5-32
Вольвач Михаил
Подпись и дата:

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

2022

Изменённый текст программы

```
from operator import itemgetter

from RK1_files.DB import data


# 1:M
def one_to_many():
    return [(m.sur, m.sal, o.name)
            for o in data.orchs
            for m in data.muss
            if m.orch_id == o.id]


def many_to_many_tmp():
    return [(o.name, mo.orch_id, mo.mus_id)
            for o in data.orchs
            for mo in data.mus_orchs
            if o.id == mo.orch_id]


def many_to_many():
    return [(m.sur, m.sal, orch_name)
            for orch_name, _, mus_id in many_to_many_tmp()
            for m in data.muss if m.id == mus_id]


def B1():
    return sorted(one_to_many(), key=itemgetter(0))


def B2():
    res_unsorted = []

    for o in data.orchs:
        o_data = {}
        # Список музыкантов оркестра
        o_muss = list(filter(lambda i: i[2] == o.name, one_to_many()))
        # print(o.name, len(o_muss))
        o_data["name"] = o.name
        o_data["len"] = len(o_muss)
        res_unsorted.append(o_data)

    return sorted(res_unsorted, key=itemgetter("len"), reverse=True)


def B3():
    res = []
    for m in data.muss:
        m_data = {}
        if "ов" in m.sur:
            m_data["sur"] = m.sur
            # Список оркестров музыканта
            m_orchs = [mtmt[0] for mtmt in many_to_many_tmp() if mtmt[1] ==
m.orch_id]
            m_data["orchs"] = list(set(m_orchs))
            res.append(m_data)
    return res


def main():
    """Основная функция"""
```

```
    print("Задание Б1")
    res_b1 = B1()
    print(res_b1)

    print("Задание Б2")
    res_b2 = B2()
    [print(i) for i in res_b2]

    print("Задание Б3")
    res_b3 = B3()
    [print(i, end="\n") for i in res_b3]


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

Тестирование:

```
import unittest

from RK1_files.rk1_classes.musician import Musician
from RK1_files.rk1_classes.orchestra import Orchestra
from RK1_files.rk1_classes.musorchestra import MusOrch
from RK1_files.main import B1
from RK1_files.main import B2
from RK1_files.main import B3


class Test(unittest.TestCase):
    def setUp(self) -> None:
        self.orchs = [
            Orchestra(1, "Духовой"),
            Orchestra(2, "Струнный"),
            Orchestra(3, "Симфонический"),

            Orchestra(11, "Духовой (другой)"),
            Orchestra(22, "Струнный (другой)"),
            Orchestra(33, "Симфонический (другой"),
        ]

        self.muss = [
            Musician(1, "Иванов", "Флейта", 40000, 1),
            Musician(2, "Петрова", "Арфа", 30000, 3),
            Musician(3, "Сидоров", "Скрипка", 35000, 2),
            Musician(4, "Иваненко", "Арфа", 45000, 2),
            Musician(5, "Иванин", "Тромбон", 50000, 1),
        ]

        self.mus_orchs = [
            MusOrch(1, 1),
            MusOrch(1, 5),
            MusOrch(2, 3),
            MusOrch(3, 2),
            MusOrch(3, 4),

            MusOrch(11, 1),
            MusOrch(11, 5),
            MusOrch(22, 3),
            MusOrch(33, 2),
            MusOrch(33, 4),
        ]

    def test_B1(self):
        expected_result = [('Иваненко', 45000, 'Струнный оркестр'),
('Иванин', 50000, 'Духовой оркестр'), ('Иванов', 40000, 'Духовой оркестр'),
('Петрова', 30000, 'Симфонический оркестр'), ('Сидоров', 35000, 'Струнный
оркестр')]
        res = B1()
        self.assertEqual(res, expected_result)

    def test_B2(self):
        expected_result = [{'name': 'Духовой оркестр', 'len': 2},
                           {'name': 'Струнный оркестр', 'len': 2},
                           {'name': 'Симфонический оркестр', 'len': 1},
                           {'name': 'Духовой (другой) оркестр', 'len': 0},
                           {'name': 'Струнный (другой) оркестр', 'len': 0},
                           {'name': 'Симфонический (другой оркестр', 'len':
0}]
        res = B2()
        self.assertEqual(res, expected_result)
```

```
    def test_B3(self):
        expected_result = [{'sur': 'Иванов', 'orchs': ['Духовой оркестр']},
                           {'sur': 'Петрова', 'orchs': ['Симфонический
оркестр']},
                           {'sur': 'Сидоров', 'orchs': ['Струнный оркестр']}]
        res = B3()
        self.assertEqual(res, expected_result)


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

Результат тестирования:

```
Testing started at 12:58 ...
Launching pytest with arguments /Users/mikhail/Documents/BKIT/RK2/test.py --no-header --no-summary -q in /Users/mikhail/Documents/BKIT/RK2

============================= test session starts ==============================
collecting ... collected 3 items

test.py::Test::test_B1 PASSED                                            [ 33%]
test.py::Test::test_B2 PASSED                                            [ 66%]
test.py::Test::test_B3 PASSED                                            [100%]

============================== 3 passed in 0.02s ===============================

Process finished with exit code 0
```