Текст программы

Текст RK2.py

```
from operator import itemgetter


class Pupil:
    def __init__(self, id, fio, skip, class_id):
        self.id = id
        self.fio = fio
        self.skip = skip
        self.class_id = class_id


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


class PupilClass:
    def __init__(self, class_id, pupil_id):
        self.class_id = class_id
        self.pupil_id = pupil_id


def one_to_many(classes, pupils):
    return [(p.fio, p.skip, c.name)
            for c in classes
            for p in pupils
            if p.class_id == c.id]


def many_to_many(classes, pupils_classes, pupils):
    temp = [(c.name, pc.class_id, pc.pupil_id)
            for c in classes
            for pc in pupils_classes
            if c.id == pc.class_id]
    return [(p.fio, p.skip, class_name)
            for class_name, class_id, pupil_id in temp
            for p in pupils
            if p.id == pupil_id]


def task1(one_to_many_data):
    return sorted(one_to_many_data, key=itemgetter(2))


def task2(one_to_many_data, classes):
    result_unsorted = []
    for c in classes:
        c_pupils = list(filter(lambda i: i[2] == c.name, one_to_many_data))
        if len(c_pupils) > 0:
            c_skips = [skip for _, skip, _ in c_pupils]
            c_skips_sum = sum(c_skips)
            result_unsorted.append((c.name, c_skips_sum))
    return sorted(result_unsorted, key=itemgetter(1), reverse=True)


def task3(many_to_many_data, classes):
    result = {}
    for c in classes:
        if 'A' in c.name:
            c_pupils = list(filter(lambda i: i[2] == c.name, many_to_many_data))
            c_pupils_names = [x for x, _, _ in c_pupils]
            result[c.name] = c_pupils_names
    return result


def main():
    classes = [
        SchClass(1, '1А'),
        SchClass(2, '1Б'),
        SchClass(3, '1В'),
        SchClass(11, '2А'),
        SchClass(22, '3Б'),
        SchClass(33, '4Г'),
    ]

    pupils = [
        Pupil(1, 'Аксенов', 3, 1),
        Pupil(2, 'Соломахин', 4, 1),
        Pupil(3, 'Лемзиков', 6, 2),
        Pupil(4, 'Семенов', 0, 2),
        Pupil(5, 'Уваров', 10, 3),
    ]

    pupils_classes = [
        PupilClass(1, 1),
        PupilClass(1, 2),
        PupilClass(2, 3),
        PupilClass(2, 4),
        PupilClass(3, 5),
        PupilClass(11, 1),
        PupilClass(11, 2),
        PupilClass(22, 3),
        PupilClass(22, 4),
        PupilClass(33, 5),
    ]

    otm = one_to_many(classes, pupils)
    mtm = many_to_many(classes, pupils_classes, pupils)

    print('Задание А1')
    res_11 = task1(otm)
    print(res_11)

    print('\nЗадание А2')
    res_12 = task2(otm, classes)
    print(res_12)

    print('\nЗадание А3')
    res_13 = task3(mtm, classes)
    print(res_13)


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

Текст test.py

```
import unittest
from RK2 import *


class TestSchoolTasks(unittest.TestCase):

    def setUp(self):
        # Создаем тестовые данные
        self.classes = [
            SchClass(1, '1А'),
            SchClass(2, '1Б'),
            SchClass(3, '1В'),
            SchClass(11, '2А'),
            SchClass(22, '3Б'),
            SchClass(33, '4Г'),
        ]

        self.pupils = [
            Pupil(1, 'Аксенов', 3, 1),
            Pupil(2, 'Соломахин', 4, 1),
            Pupil(3, 'Лемзиков', 6, 2),
            Pupil(4, 'Семенов', 0, 2),
            Pupil(5, 'Уваров', 10, 3),
        ]

        self.pupils_classes = [
            PupilClass(1, 1),
            PupilClass(1, 2),
            PupilClass(2, 3),
            PupilClass(2, 4),
            PupilClass(3, 5),
            PupilClass(11, 1),
            PupilClass(11, 2),
            PupilClass(22, 3),
            PupilClass(22, 4),
            PupilClass(33, 5),
        ]

    def test_task1(self):
        otm = one_to_many(self.classes, self.pupils)
        result = task1(otm)
        expected_result = [('Аксенов', 3, '1А'), ('Соломахин', 4, '1А'),
                           ('Лемзиков', 6, '1Б'), ('Семенов', 0, '1Б'),
                           ('Уваров', 10, '1В')]
        self.assertEqual(result, expected_result)

    def test_task2(self):
        otm = one_to_many(self.classes, self.pupils)
        result = task2(otm, self.classes)
        expected_result = [('1В', 10), ('1А', 7), ('1Б', 6)]
        self.assertEqual(result, expected_result)

    def test_task3(self):
        mtm = many_to_many(self.classes, self.pupils_classes, self.pupils)
        result = task3(mtm, self.classes)
        expected_result = {'1А': ['Аксенов', 'Соломахин'],
                           '2А': ['Аксенов', 'Соломахин'],}
        self.assertEqual(result, expected_result)

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

Результаты выполнения РК1:

Задание А1

```
[('Аксенов', 3, '1А'), ('Соломахин', 4, '1А'), ('Лемзиков', 6, '1Б'), ('Семенов', 0,
'1Б'), ('Уваров', 10, '1В')]
```

Задание А2

```
[('1В', 10), ('1А', 7), ('1Б', 6)]
```

Задание А3

```
{'1А': ['Аксенов', 'Соломахин'], '2А': ['Аксенов', 'Соломахин']}
```

Результаты выполнения РК2:

```
...
----------------------------------------------------------------------
Ran 3 tests in 0.005s

OK
```

Изменим значение expected_result в 3 тесте на:

```
{'1А': ['Уваров', 'Лемзиков'],

 '2А': ['Аксенов', 'Соломахин'],}
```

```
..F
======================================================================
FAIL: test_task3 (__main__.TestSchoolTasks)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "D:\VSprojects\RK2\RK2\test.py", line 58, in test_task3
    self.assertEqual(result, expected_result)
AssertionError: {'1А': ['Аксенов', 'Соломахин'], '2А': ['Аксенов', 'Соломахин']}
!= {'1А': ['Уваров', 'Лемзиков'], '2А': ['Аксенов', 'Соломахин']}
- {'1А': ['Аксенов', 'Соломахин'], '2А': ['Аксенов', 'Соломахин']}
+ {'1А': ['Уваров', 'Лемзиков'], '2А': ['Аксенов', 'Соломахин']}

----------------------------------------------------------------------
Ran 3 tests in 3.717s

FAILED (failures=1)
```