

Исправленный код РК1, пригодный для модульного тестирования:

```
from operator import itemgetter

class pk:
    """Компьютер"""
    def __init__(self, id, processor, ram, classroom_id):
        self.id = id
        self.processor = processor
        self.ram = ram
        self.classroom_id = classroom_id

class classroom:
    """Аудитория"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class classroom_pk:
    """Компьютер в аудитории(для связи многие ко многим)"""
    def __init__(self, classroom_id, pk_id):
        self.pk_id = pk_id
        self.classroom_id = classroom_id

#Аудитории
classrooms = [
    classroom(1, "Дисплейный класс физики"),
    classroom(2, "Дисплейный класс информационного направления"),
    classroom(3, "Дисплейный класс химии"),
    classroom(4, "Дисплейный класс математического направления"),
    classroom(5, "Дисплейный класс географии"),
    classroom(6, "Дисплейный класс направления информационной безопасности"),
]

#Компьютеры
pks = [
    pk(1, "Intel i5", 8, 1),
    pk(2, "Intel i7", 16, 2),
    pk(3, "Intel i9", 16, 3),
    pk(4, "Ryzen 5", 32, 3),
    pk(5, "Ryzen 7", 8, 3),
]

classrooms_pks = [
    classroom_pk(1,1),
    classroom_pk(2,2),
    classroom_pk(3,3),
    classroom_pk(3,4),
```

```

classroom_pk(3,5),
classroom_pk(4,1),
classroom_pk(5,2),
classroom_pk(6,3),
classroom_pk(6,4),
classroom_pk(6,5),
]

#Сортировка по аудиториям
def first(spisok):
    res_1 = sorted(spisok, key=itemgetter(2))
    return res_1

#Сортировка аудиторий по суммарной ram
def second(spisok):
    res_2 = []
    temp_res2 = dict()
    for i in spisok:
        if i[2] in temp_res2:
            temp_res2[i[2]] += i[1]
        else:
            temp_res2[i[2]] = i[1]
    for i in temp_res2.keys():
        res_2.append((i, temp_res2[i]))
    res_2 = sorted(res_2, key=itemgetter(1))
    return res_2

#Список всех аудиторий, в названии которых есть слово "направления" с процессо-
рами компьютеров в них
def third(spisok, slovo):
    res_3 = []
    temp_res3 = dict()
    for i in spisok:
        if slovo in i[2]:
            if i[2] in temp_res3:
                temp = temp_res3[i[2]]
                temp.append(i[0])
                temp_res3[i[2]] = temp
            else:
                temp = []
                temp.append(i[0])
                temp_res3[i[2]] = temp
    return temp_res3

def main():
    one_to_many = [(p.processor, p.ram, c.name)
                    for c in classrooms
                    for p in pks
                    if p.classroom_id==c.id]

    many_to_many_temp = [(c.name, p.classroom_id, p.pk_id)

```

```

        for c in classrooms
        for p in classrooms_pks
        if c.id==p.classroom_id]

many_to_many = [(p.processor, p.ram, classroom_name)
    for classroom_name, classroom_id, pk_id in many_to_many_temp
    for p in pks if p.id==pk_id]

print('Задание A1')
print(first(one_to_many))
print('\nЗадание A2')
print(second(one_to_many))
print('\nЗадание A3')
print(third(many_to_many, 'направления'))

if __name__ == '__main__':
    main()

```

rk1_tests.py:

```

import RK1
from operator import itemgetter
import unittest

class tests(unittest.TestCase):
    def test_first(self):
        test_spisok = [(3,3,3),(1,1,1),(2,2,2)]
        sp1 = RK1.first(test_spisok)
        sp2 = sorted(test_spisok, key = itemgetter(2))
        self.assertEqual(sp1,sp2)

    def test_second(self):
        test_spisok = [('Имя1', 1, 'Аудитория1'), ('Имя2', 2, 'Аудитория2'),
('Имя3', 3, 'Аудитория3'), ('Имя4', 4, 'Аудитория3')]
        sp1 = RK1.second(test_spisok)
        sp2 = [('Аудитория1', 1), ('Аудитория2', 2), ('Аудитория3', 7)]
        self.assertEqual(sp1,sp2)

    def test_third(self):
        test_spisok = [('Имя1', 1, 'Аудитория1 да'), ('Имя2', 2, 'Аудитория2'),
('Имя3', 3, 'Аудитория3 да'), ('Имя4', 4, 'Аудитория3 да')]
        sp1 = RK1.third(test_spisok, 'да')
        sp2 = {'Аудитория1 да': ['Имя1'], 'Аудитория3 да': ['Имя3', 'Имя4']}
        self.assertEqual(sp1,sp2)

```

Результат выполнения тестов:

```
PS C:\Users\Andrey\Desktop\инфа> python -m unittest rk1_tests.py
```

```
...
```

```
-----
```

```
Ran 3 tests in 0.000s
```

```
OK
```

```
PS C:\Users\Andrey\Desktop\инфа> █
```