```python
class Student:
    """Студент"""
    def __init__(self, id, surname, average_rating, group_id):
        self.id = id
        self.surname = surname
        self.average_rating = average_rating
        self.group_id = group_id


class Group:
    """Группа"""

    def __init__(self, id, group_name):
        self.id = id
        self.group_name = group_name


class StudentGroup:
    """
    'Студенты группы' для реализации связи многие-ко-многим
    """

    def __init__(self, group_id, student_id):
        self.group_id = group_id
        self.student_id = student_id


# Группы
groups = [
    Group(1, 'IU5-31B'),
    Group(2, 'IU5-32B'),
    Group(3, 'IU5-33B'),
    Group(4, 'IU7-54B'),

    Group(5, 'RT5-12B'),
    Group(6, 'RT7-24B'),
]
# Студенты
students = [
    Student(1, 'Ivanov', 3.4, 1),
    Student(2, 'Baranov', 4.2, 2),
    Student(3, 'Popov', 4, 4),
    Student(4, 'Andreev', 4.8, 5),
    Student(5, 'Sidorov', 3.7, 3),
]
student_group = [
    StudentGroup(1, 1),
    StudentGroup(1, 2),
    StudentGroup(4, 3),
    StudentGroup(3, 4),
    StudentGroup(2, 5),

    StudentGroup(5, 1),
    StudentGroup(5, 2),
    StudentGroup(6, 4),
    StudentGroup(6, 5),
]


def main():
    # Соединение данных один-ко-многим
    one_to_many = [(y.surname, y.average_rating, x.group_name)
                   for x in groups
                   for y in students
```

```
                    if y.group_id == x.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(x.group_name, y.group_id, y.student_id)
                         for x in groups
                         for y in student_group
                         if x.id == y.group_id]

    many_to_many = [(x.surname, x.average_rating, group_name)
                    for group_name, group_id, student_id in many_to_many_temp
                    for x in students if x.id == student_id]
    print('Задание Д1')
    res_11 = []
    for student_surname, student_average_rating, student_group_id in
one_to_many:
        if student_surname.endswith('ov'):
            res_11.append((student_surname, student_group_id))
    print(res_11)
    print('\nЗадание Д2')
    res_12 = {}
    for g in groups:
        g_students= list(filter(lambda i: i[2] == g.group_name, one_to_many))
        if len(g_students) > 0:
            g_students_average_ratings = [x for _, x, _ in g_students]
            res_12[g.group_name] = (sum(g_students_average_ratings) /
len(g_students_average_ratings))
    print(sorted(res_12.items(), key=lambda item: item[1]))
    print('\nЗадание Д3')
    res_13 = {}
    for g in groups:
        if g.group_name[0] == 'I':
            g_students = list(filter(lambda i: i[2] == g.group_name,
many_to_many))
            g_students_surnames = [x for x, _, _ in g_students]
            res_13[g.group_name] = g_students_surnames
    print(res_13)

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

```
Задание Д1
[('Ivanov', 'IU5-31B'), ('Baranov', 'IU5-32B'), ('Sidorov', 'IU5-33B'), ('Popov', 'IU7-54B')]

Задание Д2
[('IU5-31B', 3.4), ('IU5-33B', 3.7), ('IU7-54B', 4.0), ('IU5-32B', 4.2), ('RT5-12B', 4.8)]

Задание Д3
{'IU5-31B': ['Ivanov', 'Baranov'], 'IU5-32B': ['Sidorov'], 'IU5-33B': ['Andreev'], 'IU7-54B': ['Popov']}
```