

Код

```
# вариант запроса Д
# вариант предметной области 4 : компьютер - дисплейный класс
from operator import itemgetter

class Computer:
    # компьютер
    def __init__(self, id, name, cost, disp_class_id):
        self.id = id
        self.name = name
        self.cost = cost
        self.disp_class_id = disp_class_id

class DisplayClass:
    # дисплейный класс
    def __init__(self, id, name):
        self.id = id
        self.name = name

class CompDispClass:
    # компьютеры дисплейного класса для реализации связи
    # многие-ко-многим
    def __init__(self, disp_class_id, comp_id):
        self.disp_class_id = disp_class_id
        self.comp_id = comp_id

# дисплейные классы
disp_classes = [
    DisplayClass(1, "Аудитория 325"),
    DisplayClass(2, "Аудитория 501"),
    DisplayClass(3, "Лабораторная 611"),

    DisplayClass(4, "Лабораторная 728"),
    DisplayClass(5, "Лабораторная 210"),
    DisplayClass(6, "Аудитория 409")
]

# компьютеры
comps = [
    Computer(1, "ASUS", 15000, 1),
    Computer(2, "DELL", 50000, 2),
    Computer(3, "ACER", 45500, 2),
    Computer(4, "MAC", 120000, 3),
    Computer(5, "ASUS", 15000, 3),
    Computer(6, "MSI", 75000, 3),
    Computer(7, "DELL", 50000, 3)
]

comps_disp_classes = [
    CompDispClass(1, 1),
    CompDispClass(2, 2),
    CompDispClass(2, 3),
    CompDispClass(3, 4),
    CompDispClass(3, 5),
    CompDispClass(3, 6),
    CompDispClass(3, 7),

    CompDispClass(4, 1),
    CompDispClass(5, 2),
    CompDispClass(5, 3),
```

```

CompDispClass(6, 4),
CompDispClass(6, 5),
CompDispClass(6, 6),
CompDispClass(6, 7),
]

def main():
    # соединение данных один-ко-многим
    one_to_many = [(c.name, c.cost, o.name)
                    for o in disp_classes
                    for c in comps
                    if c.disp_class_id == o.id]

    # соединение данных многие-ко-многим
    many_to_many_temp = [(o.name, co.disp_class_id, co.comp_id)
                          for o in disp_classes
                          for co in comps_disp_classes
                          if o.id == co.disp_class_id]

    many_to_many = [(c.name, c.cost, disp_class_name)
                     for disp_class_name, disp_class_id, comp_id in
many_to_many_temp
                     for c in comps if c.id == comp_id]

    print('Задание Д1')
    res1 = []
    for i in one_to_many:
        if i[0][-2:] == "US":
            res1.append(i[0:3:2])
    print(res1)

    print('\nЗадание Д2')
    res2_unsorted = []
    for d_c in disp_classes:
        d_c_comps = list(filter(lambda i: i[2] == d_c.name, one_to_many))
        if len(d_c_comps) > 0:
            d_c_cost = [cost for _, cost, _ in d_c_comps]
            d_c_cost_sum = sum(d_c_cost)
            d_c_cost_count = len(d_c_cost)
            d_c_cost_average = d_c_cost_sum / d_c_cost_count
            res2_unsorted.append((d_c.name, int(d_c_cost_average)))
    res2 = sorted(res2_unsorted, key=itemgetter(1), reverse=True)
    print(res2)

    print('\nЗадание Д3')
    res3 = {}
    for d_c in disp_classes:
        if d_c.name[0] == "J":
            d_c_comps = list(filter(lambda i: i[2] == d_c.name,
many_to_many))
            d_c_comps_names = [x for x, _, _ in d_c_comps]
            res3[d_c.name] = d_c_comps_names
    print(res3)

if __name__ == '__main__':
    main()

```

Результат

```
C:\Lab_Python\RIP_labs\RK1\venv\Scripts\python.exe C:/Lab_Python/RIP_labs/RK1/RK1.py
```

```
Задание Д1
```

```
[('ASUS', 'Аудитория 325'), ('ASUS', 'Лабораторная 611')]
```

```
Задание Д2
```

```
[('Лабораторная 611', 65000), ('Аудитория 501', 47750), ('Аудитория 325', 15000)]
```

```
Задание Д3
```

```
{'Лабораторная 611': ['MAC', 'ASUS', 'MSI', 'DELL'], 'Лабораторная 728': ['ASUS'], 'Лабораторная 210': ['DELL', 'ACER']}
```

```
Process finished with exit code 0
```