Васильев Денис. Группа: ИУ5-52б.

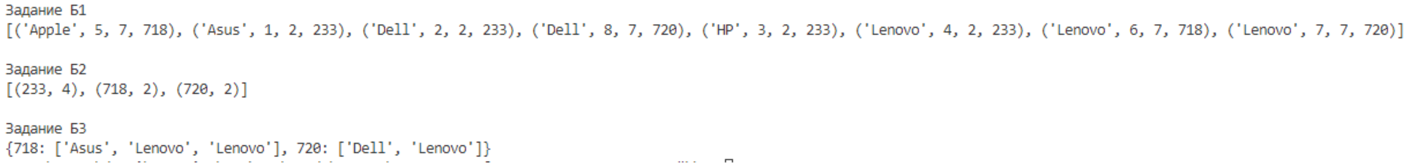
Вариант 4Б

№ варианта	Класс 1	Класс 2
4	Компьютер	Дисплейный класс

```python
from operator import itemgetter

class Computer:
    def __init__(self, id, model, cost, class_id):
        self.id = id
        self.model = model
        self.class_id = class_id
        self.cost = cost

class DisplayClass:
    def __init__(self, id, floor, num):
        self.id = id
        self.floor = floor
        self.num = num

class ComputerDisplayClass:
    def __init__(self, comp_id, disp_id):
        self.comp_id = comp_id
        self.disp_id = disp_id

computers = [
    Computer(1, "Asus",   40000, 1),
    Computer(2, "Dell",   30000, 1),
    Computer(3, "HP",     35000, 1),
    Computer(4, "Lenovo", 45000, 1),

    Computer(5, "Apple",  70000, 2),
    Computer(6, "Lenovo", 25000, 2),

    Computer(7, "Lenovo", 56000, 3),
    Computer(8, "Dell",   34000, 3),
]

classes = [
    DisplayClass(1, 2, 2_33),
    DisplayClass(2, 7, 7_18),
    DisplayClass(3, 7, 7_20),
]
```

```
computers_classes = [
    ComputerDisplayClass(1, 1),
    ComputerDisplayClass(1, 2),
    ComputerDisplayClass(2, 1),
    ComputerDisplayClass(2, 3),
    ComputerDisplayClass(3, 1),
    ComputerDisplayClass(4, 1),
    ComputerDisplayClass(5, 1),
    ComputerDisplayClass(6, 1),
    ComputerDisplayClass(6, 2),
    ComputerDisplayClass(6, 3),
    ComputerDisplayClass(7, 2),
    ComputerDisplayClass(8, 1),
]

def main():
    one_to_many = [(comp.model, comp.id, cl.floor, cl.num)
        for comp in computers
        for cl in classes
        if comp.class_id == cl.id]

    many_to_many_tmp = [(cl.floor, cl.num, cc.comp_id, cc.disp_id)
        for cl in classes
        for cc in computers_classes
        if cl.id == cc.disp_id]

    many_to_many = [(comp.model, comp.cost, cl_floor, cl_num)
        for cl_floor, cl_num, comp_id, disp_id in many_to_many_tmp
        for comp in computers
        if comp.id == comp_id]

    # Список всех связанных компьютеров и классов, отсортированный по модели
компьютеров.
    print("Задание Б1")
    res1 = sorted(one_to_many, key=itemgetter(0))
    print(res1)

    #  Список классов с количеством компьютеров в каждом классе, отсортированный
по количеству компьютеров.
    print("\nЗадание Б2")
    res2_unsorted = []
    for cl in classes:
        cc = list(filter(lambda i: i[3]==cl.num, one_to_many))
        if len(cc) > 0:
            class_comps = [comp_id for _,comp_id,_,_ in cc]
            class_comps_count = len(class_comps)
            res2_unsorted.append((cl.num, class_comps_count))
    res2 = sorted(res2_unsorted, key=itemgetter(1), reverse=True)
    print(res2)
    # Список моделей компьютеров из классов, номер которых начинается на 7 и номера их классов
    print("\nЗадание Б3")
    res3 = {}
    for cl in classes:
        if str(cl.num).startswith('7'):
            cc = list(filter(lambda i: i[3]==cl.num, many_to_many))
            cc_nums = [x for x,_,_,_ in cc]
            res3[cl.num] = cc_nums

    print(res3)

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

Результат работы программы:



```
Задание Б1
[('Apple', 5, 7, 718), ('Asus', 1, 2, 233), ('Dell', 2, 2, 233), ('Dell', 8, 7, 720), ('HP', 3, 2, 233), ('Lenovo', 4, 2, 233), ('Lenovo', 6, 7, 718), ('Lenovo', 7, 7, 720)]

Задание Б2
[(233, 4), (718, 2), (720, 2)]

Задание Б3
{718: ['Asus', 'Lenovo', 'Lenovo'], 720: ['Dell', 'Lenovo']}
```