**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] == "Л":  
 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**

