

Рубежный контроль №1

Вариант №Г7

```

class Mic:
    """Микропроцессор"""
    def __init__(self, id, name, price, comp_id):
        self.id = id
        self.name = name
        self.price = price
        self.comp_id = comp_id

class Comp:
    """Компьютер"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Mic_Comp:
    """
    'Микропроцессоры компьютера' для реализации
    связи многие-ко-многим
    """
    def __init__(self, comp_id, mic_id):
        self.comp_id = comp_id
        self.mic_id = mic_id

#Компьютеры
comps = [
    Comp(1, 'Acer Z5230'),
    Comp(2, 'Lenovo Y570'),
    Comp(3, 'Asus RoG1720'),
    Comp(4, 'Apple MacBook13')
]

# Процессоры
mics = [
    Mic(1, 'Intel i5 7880', 25000, 1),
    Mic(2, 'Intel i7 3620', 32000, 2),
    Mic(3, 'Amd Ryzen X5', 42000, 3),
    Mic(4, 'Intel Celeron G3900', 15000, 3),
    Mic(5, 'Amd FX5300', 20000, 3),
    Mic(6, 'Apple M1', 27000, 4)
]

mic_comps = [
    Mic_Comp(1,1),
    Mic_Comp(2,2),
    Mic_Comp(3,3),
    Mic_Comp(3,4),
    Mic_Comp(3,5),
    Mic_Comp(4,6)
]

one_to_many = [(m.name, m.price, c.name)
                for m in mics
                for c in comps
                if m.comp_id == c.id]

many_to_many = [(m.name, m.price, comp_name)
                 for comp_name, comp_id, mic_id in many_to_many_temp
                 for m in mics if m.id==mic_id]

"""##Задание Г1
«Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
"""
print("Г1")
print(list(filter(lambda x: x[2].startswith("А"), one_to_many)))

"""##Задание Г2
«Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
"""
print("Г2")
res_g2 = []
for c in comps:
    # Список микропроцессоров компьютера
    c_mic = list(filter(lambda i: i[2] == c.name, one_to_many))
    # Если у компьютера есть микропроцессоры
    if len(c_mic) > 0:
        # Максимальная зарплата
        max_price = max([price for _, price, _ in c_mic])
        res_g2.append((c.name, max_price))
print(sorted(res_g2, key = lambda x: x[1]))

"""##Задание Г3
«Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.
"""
print("Г3")
print(sorted(many_to_many, key = lambda x: x[2]))

```

Результаты выполнения программы:

```

Г1
[('Intel i5 7880', 25000, 'Acer Z5230'), ('Amd Ryzen X5', 42000, 'Asus RoG1720'), ('Intel Celeron G3900', 15000, 'Asus RoG1720'), ('Amd FX5300', 20000, 'Asus RoG1720'), ('Apple M1', 27000, 'Apple MacBook13')]
Г2
[('Acer Z5230', 25000), ('Apple MacBook13', 27000), ('Lenovo Y570', 32000), ('Asus RoG1720', 42000)]
Г3
[('Intel i5 7880', 25000, 'Acer Z5230'), ('Apple M1', 27000, 'Apple MacBook13'), ('Amd Ryzen X5', 42000, 'Asus RoG1720'), ('Intel Celeron G3900', 15000, 'Asus RoG1720'), ('Amd FX5300', 20000, 'Asus RoG1720'), ('Intel i7 3620', 32000, 'Lenovo Y570')]

```