

Условие:

Вариант Г. 27

27	Преподаватель	Учебный курс
----	---------------	--------------

1. «Преподаватель» и «Учебный курс» связаны соотношением один-ко-многим. Выведите список всех учебного курса, у которых название начинается с буквы «А», и список преподавателя, в котором они работают.
2. «Преподаватель» и «Учебный курс» связаны соотношением один-ко-многим. Выведите список учебных курсов с максимальной зарплатой преподавателя в каждом учебном курсе, отсортированный по максимальной зарплате.
3. «Преподаватель» и «Учебный курс» связаны соотношением многие-ко-многим. Выведите список всех связанных преподавателей и учебного курса, отсортированный по преподавателя, сортировка по учебному курсу произвольная.

```
#!/usr/bin/env python

from operator import itemgetter

class teachers:
    """Преподаватель"""
    def __init__(self, id, name, price, trainingcourse_id):
        self.id = id
        self.name = name
        self.price = price
        self.trainingcourse_id = trainingcourse_id

class Trainingcourse:
    """Учебный курс"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class teachersTrainingcourse:
    """Учебный курс к преподавателю, для реалиации многие-ко-многим"""
    def __init__(self, teachers_id, trainingcourse_id):
        self.teachers_id = teachers_id
        self.trainingcourse_id = trainingcourse_id

if __name__ == "__main__":
```

```

Trainingcourse = [
    Trainingcourse(0, "Бизнес"),
    Trainingcourse(1, "Дизайн"),
    Trainingcourse(2, "Разработка игр"),
    Trainingcourse(3, "Информационные технологии"),
    Trainingcourse(4, "Развитие личности"),
    Trainingcourse(5, "Автошкола"),
]

teachers = [
    teachers(0, "Путин", 250000, 0),
    teachers(1, "Медведев", 150000, 1),
    teachers(2, "Карпов", 28000, 2),
    teachers(3, "Семенова", 164000, 3),
    teachers(4, "Рубин", 20800, 3),
    teachers(5, "Шанаурина", 1047000, 3),
    teachers(6, "Алейников", 110000, 3),
    teachers(7, "Крутова", 120000, 2),
    teachers(8, "Малкина", 1300, 2),
    teachers(9, "Распашнов", 260000, 1),
    teachers(10, "Бузова", 3020, 4),
    teachers(11, "Пугачева", 4080, 5),
    teachers(12, "Касас", 72000, 5),
    teachers(13, "Аистов", 6040, 4),
    teachers(14, "Жириновский", 7070, 3),
]

teachers_Trainingcourse = [
    teachersTrainingcourse(0, 0),
    teachersTrainingcourse(1, 1),
    teachersTrainingcourse(2, 2),
    teachersTrainingcourse(3, 3),
    teachersTrainingcourse(3, 4),
    teachersTrainingcourse(3, 5),
    teachersTrainingcourse(5, 5),
    teachersTrainingcourse(4, 4),
    teachersTrainingcourse(8, 1),
    teachersTrainingcourse(11, 5),
]

# Соединение данных один-ко-многим
one_to_many = [(d.name, d.price, s.name)
    for s in Trainingcourse
    for d in teachers
    if d.trainingcourse_id==s.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(s.name, ds.trainingcourse_id, ds.teachers_id)
    for s in Trainingcourse
    for ds in teachers_Trainingcourse

```

```

        if s.id==ds.trainingcourse_id]
    many_to_many = [(d.name, d.price, Trainingcourse_name)
        for Trainingcourse_name, trainingcourse_id, teachers_id in
many_to_many_temp
        for d in teachers if d.id==teachers_id]

    print("Задание Г1")
    res_11 = {}
    selected_Trainingcourse = [one_traicour[2] for one_traicour in one_to_many if
one_traicour[2].startswith('a') or one_traicour[2].startswith('A')]
    for Trainingcourse_name in selected_Trainingcourse:
        teachers_for_traicour = [(one_teachers[0],one_teachers[1]) for
one_teachers in one_to_many if one_teachers[2]==Trainingcourse_name]
        res_11.update({Trainingcourse_name:teachers_for_traicour})
    print(res_11)
    print()

    print("Задание Г2")
    res_12_unsorted = []
    for s in Trainingcourse:
        s_teachers = list(filter(lambda i: i[2]==s.name, one_to_many))
        if len(s_teachers) > 0:
            s_prices = [price for _,price,_ in s_teachers]
            s_price_max = max(s_prices)
            res_12_unsorted.append((s.name, s_price_max))

    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)
    print()

    print("Задание Г3")
    res_13 = {}
    Trainingcourse.sort(key=lambda one_Trainingcourse: one_Trainingcourse.name)
    for s in Trainingcourse:
        s_teachers = list(filter(lambda i: i[2]==s.name, many_to_many))
        s_teachers_names = [x for x,_,_ in s_teachers]
        res_13[s.name] = s_teachers_names
    print(res_13)

```

```

PS c:\Users\Лиза\OneDrive\Рабочий стол\Баумана 7-ий семестр\РПИ\РК1> c:: cd 'c:\Users\Лиза\OneDrive\Рабочий стол\Баумана 7-ий семестр\РПИ\РК1'; & 'c:\Users\Лиза\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\Лиза\.vscode\extensions\ms-python.python-2021.10.1336267007\pythonFiles\lib\python\debugpy\launcher' '63027' '--' 'c:\User
s\Лиза\OneDrive\Рабочий стол\Баумана 7-ий семестр\РПИ\РК1\Шанаурина .ру'
Задание Г1
{'Автошкола': [('Пугачева', 4080), ('Касас', 72000)]}

Задание Г2
[('Информационные технологии', 1047000), ('Дизайн', 260000), ('Бизнес', 250000), ('Разработка игр', 120000), ('Автошкола', 72000), ('Развитие личности', 6040)]

Задание Г3
{'Автошкола': ['Крутова', 'Шанаурина', 'Пугачева'], 'Бизнес': ['Путин'], 'Дизайн': ['Медведев', 'Малкина'], 'Информационные технологии': ['Крутова'], 'Развитие личности': ['
Крутова', 'Рубин'], 'Разработка игр': ['Карлов']}
PS c:\Users\Лиза\OneDrive\Рабочий стол\Баумана 7-ий семестр\РПИ\РК1>

```