

# Торжков Максим ИУ5-51Б

## РК №1 Вариант 20

### Вариант А.

1. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список всех связанных деталей и поставщиков, отсортированный по поставщикам, сортировка по деталям произвольная.
2. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список поставщиков с суммарной ценой деталей у каждого поставщика, отсортированный по суммарной цене.
3. «Поставщик» и «Деталь» связаны соотношением многие-ко-многим. Выведите список всех поставщиков, у которых в названии присутствует слово «ИП», и список деталей, которые они поставляют.

### Текст программы:

```
from operator import itemgetter

class Det:
    """Деталь"""

    def __init__(self, id, name, cost, sup_id):
        self.id = id
        self.name = name
        self.cost = cost
        self.sup_id = sup_id

class Sup:
    """Поставщик"""

    def __init__(self, id, name, phone):
        self.id = id
        self.name = name
        self.phone = phone

class DetSup:
    """
    'Детали поставщика' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """

    def __init__(self, sup_id, det_id):
        self.sup_id = sup_id
        self.det_id = det_id

# Поставщики
sups = []
```

```

Sup(1, 'ООО МОСДЕТАЛЬ', '89157489390'),
Sup(2, 'ОАО ДетальСтрой', '89176374536'),
Sup(3, 'ИП Петров', '89156478479'),

Sup(11, 'ООО ПИТЕРДЕТАЛЬ', '89163748364'),
Sup(22, 'ОАО ДетальКонструкт', '83746778283'),
Sup(33, 'ИП Симонов', '47839020404'),
]

# Детали
dets = [
    Det(1, 'Гвоздь', 1000, 1),
    Det(2, 'Штуцер', 5000, 2),
    Det(3, 'Шестерня', 7000, 3),
    Det(4, 'Винт', 5000, 3),
    Det(5, 'Гайка', 700, 3),
]

dets_sup = [
    DetSup(1, 1),
    DetSup(2, 2),
    DetSup(3, 3),
    DetSup(3, 4),
    DetSup(3, 5),

    DetSup(11, 1),
    DetSup(22, 2),
    DetSup(33, 3),
    DetSup(33, 4),
    DetSup(33, 5),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(e.name, e.cost, d.name)
                    for d in sups
                    for e in dets
                    if e.sup_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, ed.sup_id, ed.det_id)
                           for d in sups
                           for ed in dets_sup
                           if d.id == ed.sup_id]

    many_to_many = [(e.name, e.cost, sup_name)
                     for sup_name, sup_id, det_id in many_to_many_temp
                     for e in dets if e.id == det_id]

    print('Задание A1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    print('\nЗадание A2')
    res_12_unsorted = []
    # Перебираем всех поставщиков
    for s in sups:
        # Список деталей поставщиков
        s_dets = list(filter(lambda i: i[2] == s.name, one_to_many))
        # Если поставщик не пустой
        if len(s_dets) > 0:

```

```

# Цены деталей поставщика
s_costs = [cost for _, cost, _ in s_dets]
# Суммарная цена деталей поставщика
s_costs_sum = sum(s_costs)
res_12_unsorted.append((s.name, s_costs_sum))

# Сортировка по суммарной цене
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание A3')
res_13 = {}
# Перебираем всех поставщиков
for s in sups:
    # Проверка на наличия ИП в названии поставщика
    if 'ИП' in s.name:
        # Список деталей поставщиков
        s_dets = list(filter(lambda i: i[2] == s.name, many_to_many))
        # Только название деталей
        s_dets_names = [x for x, _, _ in s_dets]
        # Добавляем результат в словарь
        # ключ - поставщик, значение - название детали
        res_13[s.name] = s_dets_names

print(res_13)

if __name__ == '__main__':
    main()

```

## Вывод программы:

```

C:\Users\makst\AppData\Local\Programs\Python\Python37-32\python.exe C:/Users/makst/python/rkl/main.py
Задание A1
[('Шестерня', 7000, 'ИП Петров'), ('Винт', 5000, 'ИП Петров'), ('Гайка', 700, 'ИП Петров'), ('Штуцер', 5000, 'ОАО ДетальСтрой'), ('Гвоздь', 1000, 'ООО МОСДЕТАЛЬ')]

Задание A2
[('ИП Петров', 12700), ('ОАО ДетальСтрой', 5000), ('ООО МОСДЕТАЛЬ', 1000)]

Задание A3
{'ИП Петров': ['Шестерня', 'Винт', 'Гайка'], 'ИП Симонов': ['Шестерня', 'Винт', 'Гайка']}

Process finished with exit code 0

```

### Задание A1

[('Шестерня', 7000, 'ИП Петров'), ('Винт', 5000, 'ИП Петров'), ('Гайка', 700, 'ИП Петров'), ('Штуцер', 5000, 'ОАО ДетальСтрой'), ('Гвоздь', 1000, 'ООО МОСДЕТАЛЬ')]

### Задание A2

[('ИП Петров', 12700), ('ОАО ДетальСтрой', 5000), ('ООО МОСДЕТАЛЬ', 1000)]

### Задание A3

{'ИП Петров': ['Шестерня', 'Винт', 'Гайка'], 'ИП Симонов': ['Шестерня', 'Винт', 'Гайка']}