

**Отчет по РК № 1 по курсу  
"Разработка Интернет-Приложений"**

Выполнил:  
Студент группы  
ИУ5-55Б  
Ширшов А. С.

Москва, МГТУ – 2021

## Задание:

1. «Детали» и «Производители» связаны отношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.
2. «Детали» и «Производители» связаны отношением один-ко-многим. Выведите список Производителей со средней ценой Деталей, отсортированный по средней цене.
3. «Детали» и «Производители» связаны отношением многие-ко-многим. Выведите список всех Производителей, у которых название начинается с буквы «А», и список деталей.

№ варианта	Класс 1	Класс 2
19	Деталь	Производитель

## Текст программы:

models/part.py

```
class Part:
    """Деталь"""

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

models/manufacture.py

```
class Manufacturer:
    """Производитель"""

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

## models/part\_to\_manufacturer.py

```
class PartToManufacturer:
    """Многие ко многим"""

    def __init__(self, part_id, manufacturer_id):
        self.part_id = part_id
        self.manufacturer_id = manufacturer_id
```

## main.py

```
from operator import itemgetter
from models.part import Part
from models.manufacturer import Manufacturer
from models.part_to_manufacturer import PartToManufacturer

# Производители
manufacturers = [
    Manufacturer(1, 'Автомар'),
    Manufacturer(2, 'Стройматериалы'),
    Manufacturer(3, 'Икс'),
    Manufacturer(4, 'Детальки'),
    Manufacturer(5, 'Авто и грузовики'),
    Manufacturer(6, 'Робототехника'),
    Manufacturer(7, 'Мастерская Богданов')
]

# Детали
parts = [
    Part(1, 'Автомобильное масло', 500, 1),
    Part(2, 'Доска метровая', 100, 2),
    Part(3, 'Тягачный ключ на 9', 200, 7),
    Part(4, 'Шестерня для генераторов', 10000, 4),
    Part(5, 'Уплотнитель болтов', 300, 4),
    Part(6, 'Цепь велосипедная', 350, 7),
    Part(7, 'Машинный диск', 2000, 5),
    Part(8, 'Транзистор', 1500, 6)
]

parts_to_manufacturers = [
    PartToManufacturer(1, 1),
    PartToManufacturer(1, 7),
    PartToManufacturer(2, 2),
    PartToManufacturer(2, 3),
    PartToManufacturer(3, 2),
    PartToManufacturer(3, 4),
    PartToManufacturer(3, 7),
    PartToManufacturer(4, 4),
    PartToManufacturer(5, 2),
    PartToManufacturer(5, 4),
    PartToManufacturer(6, 7),
    PartToManufacturer(6, 4),
    PartToManufacturer(7, 1),
    PartToManufacturer(7, 5),
    PartToManufacturer(8, 6),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(part.name, part.cost, manufacturer.name)
                    for manufacturer in manufacturers
                    for part in parts
                    if part.manufacturer_id == manufacturer.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(manufacturer.name, manufacturer.id, part_to_manufacturer.part_id)
                           for manufacturer in manufacturers
                           for part_to_manufacturer in parts_to_manufacturers
                           if manufacturer.id == part_to_manufacturer.manufacturer_id]

    many_to_many = [(part.name, part.cost, manufacturer.name)
                     for manufacturer.name, part_id, part_id in many_to_many_temp
                     for part in parts
                     if part.id == part_id]

    print("Задание J1")
    D1 = [(part.name, part.cost, manufacturer.name)
           for manufacturer in manufacturers
           for part in parts
           if part.manufacturer_id == manufacturer.id and part.name[-2:] == "ов"]
    print(D1)

    print("Задание J2")
    D2 = []
    for manufacturer in manufacturers:
        manufacturers_parts = list(filter(lambda i: i[2] == manufacturer.name, one_to_many))
        if len(manufacturers_parts) > 0:
            parts_costs = [cost for _, cost, _ in manufacturers_parts]
            parts_avg = sum(parts_costs) / len(parts_costs)
            D2.append((manufacturer.name, parts_avg))

    D2 = sorted(D2, key=itemgetter(1), reverse=True)
    print(D2)

    print("Задание J3")
    D3 = []
    for manufacturer in manufacturers:
        if manufacturer.name.lower().startswith('x'):
            manufacturer_parts = list(
                filter(lambda i: i[2] == manufacturer.name, many_to_many))
            parts_names = [x for x, _, _ in manufacturer_parts]
            D3[(manufacturer.name) = parts_names

    print(D3)

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

## Результат работы:

### Задание Д1

[('Шестерня для генераторов', 10000, 'Детальки'),  
('Упаковка болтов', 300, 'Детальки')]

### Задание Д2

[('Детальки', 5150.0), ('Авто и грузовики', 2000.0),  
('Робототехника', 1500.0), ('Автомир', 500.0),  
('Мастерская Богданов', 275.0), ('Стройматериалы',  
100.0)]

### Задание Д3

{ 'Автомир': ['Автомобильное масло', 'Машинный  
диск'], 'Авто и грузовики': ['Машинный диск'] }