

Рубежный контроль №1 по курсу «Базовые компоненты интернет-технологий»

Вариант 19Д

Предметная область

19	Деталь	Производитель
----	--------	---------------

Запросы (адаптированные для данной ПО)

«Производитель» и «Деталь» связаны соотношением один-ко-многим. Выведите список всех деталей, у которых наименование начинается на «А», и названия их производителей.

«Производитель» и «Деталь» связаны соотношением один-ко-многим. Выведите список производителей со средней ценой деталей у каждого производителя, отсортированный по средней цене (по убыванию).

«Производитель» и «Деталь» связаны соотношением многие-ко-многим. Выведите список всех производителей, у которых название начинается с буквы «Р», и список деталей, которые они производят.

Листинг программы

```
# используется для сортировки
from operator import itemgetter

class Detail:
    """Деталь"""

    def __init__(self, id, name, price, prod_id):
        self.id = id
        self.name = name
        self.price = price
        self.prod_id = prod_id

class Producer:
    """Производитель"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class DetProd:
    """
    'Детали производителя' для реализации
    связи многие-ко-многим
    """

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

# Детали
```

```

details = [
    Detail(1, 'RM-C', 500, 1),
    Detail(2, 'Элемент питания R03', 50, 1),
    Detail(3, 'Элемент питания R20', 100, 3),
    Detail(4, 'Транзистор полевой', 70, 8),
    Detail(5, 'Микросхема', 25, 8),
    Detail(6, 'Лампа', 100, 5),
    Detail(7, 'Блок питания', 2000, 10),
    Detail(8, 'Адаптер сетевой', 1500, 10),
    Detail(9, 'Аккумулятор R6', 100, 1),
    Detail(10, 'Реле', 280, 1),
]

# Производители
producers = [
    Producer(1, 'PANASONIC'),
    Producer(2, 'JVC'),
    Producer(3, 'ROBITON'),
    Producer(4, 'PHILIPS'),
    Producer(5, 'ФОТОН'),
    Producer(6, 'КОСМОС'),
    Producer(7, 'CAMELION'),
    Producer(8, 'GP'),
    Producer(9, 'TOSHIBA'),
    Producer(10, 'MEAN WELL')
]

dets_prods = [
    DetProd(1, 1),
    DetProd(1, 2),
    DetProd(1, 3),
    DetProd(1, 4),

    DetProd(2, 1),
    DetProd(2, 5),
    DetProd(2, 6),
    DetProd(2, 7),

    DetProd(3, 1),
    DetProd(3, 3),
    DetProd(3, 5),
    DetProd(3, 6),

    DetProd(4, 8),
    DetProd(4, 9),

    DetProd(5, 8),

    DetProd(6, 4),
    DetProd(6, 5),
    DetProd(6, 6),
    DetProd(6, 7),

    DetProd(7, 10),

    DetProd(8, 3),
    DetProd(8, 10),

    DetProd(9, 1),
    DetProd(9, 3),

    DetProd(10, 1),
]

```

```

def main():
    # Основная функция

    # Соединение данных один-ко-многим
    one_to_many = [(d.name, d.price, p.name)
                    for p in producers
                    for d in details
                    if d.prod_id == p.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(p.name, dp.prod_id, dp.det_id)
                           for p in producers
                           for dp in dets_prods
                           if p.id == dp.prod_id]

    many_to_many = [(d.name, d.price, p_name)
                     for p_name, p_id, d_id in many_to_many_temp
                     for d in details if d.id == d_id]

    print('Задание Д1')
    res_11 = filter(lambda x: x[0].startswith("A"), one_to_many)
    print(list(res_11))

    print('\nЗадание Д2')
    res_12_unsorted = []
    # Перебираем всех производителей
    for p in producers:
        # Список деталей производителя
        p_dets = list(filter(lambda i: i[2] == p.name, one_to_many))
        # Если список не пустой
        if len(p_dets) > 0:
            # Цены деталей производителя
            p_prices = [price for _, price, _ in p_dets]
            # Средняя цена деталей производителя
            d_avg_prices = sum(p_prices) / len(p_prices)
            res_12_unsorted.append((p.name, d_avg_prices))

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

    print('\nЗадание Д3')
    res_13 = {}
    # Перебираем всех производителей
    for p in producers:
        if p.name.startswith("P"):
            # Список деталей производителя
            p_dets = list(filter(lambda i: i[2] == p.name, many_to_many))
            # Только наименования деталей
            p_dets_names = [x for x, _, _ in p_dets]
            # Добавляем результат в словарь
            # ключ - производитель, значение - список наименований деталей
            res_13[p.name] = p_dets_names

    print(res_13)

if __name__ == '__main__':
    main()

```

Результат выполнения программы

Задание Д1

[('Аккумулятор R6', 100, 'PANASONIC'), ('Адаптер сетевой', 1500, 'MEAN WELL')]

Задание Д2

[('MEAN WELL', 1750.0), ('PANASONIC', 232.5), ('ROBITON', 100.0), ('ФОТОН', 100.0), ('GP', 47.5)]

Задание Д3

{'PANASONIC': ['RM-C', 'Элемент питания R03', 'Элемент питания R20', 'Аккумулятор R6', 'Реле'],
'PHILIPS': ['RM-C', 'Лампа']}