

**Московский государственный технический  
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Отчет по рубежному контролю №1

Вариант 19

Выполнила:

студентка группы ИУ5-54

Слободчикова Ю.А

Подпись и дата:

Проверил:

Гапанюк Ю.Е.

Подпись и дата:

Москва 2021 г.

## Задание.

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

1. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список всех поставщиков, у которых название начинается с буквы «А», и список поставляемых деталей
2. «Поставщик» и «Деталь» связаны соотношением один-ко-многим. Выведите список поставщиков с максимальной стоимостью деталей у каждого поставщика, отсортированный по максимальной стоимости.
3. «Поставщик» и «Деталь» связаны соотношением многие-ко-многим. Выведите список всех связанных деталей и поставщиков, отсортированный по поставщикам, сортировка по деталям произвольная.

## Код программы:

```
from operator import itemgetter

class Agent:
    """Поставщик"""

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

class Detail:
    """Деталь"""

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

class AgDet:
    """
    'Поставщики деталей' для реализации
    связи многие-ко-многим
    """

    def __init__(self, detail_id, agent_id):
        self.detail_id = detail_id
        self.agent_id = agent_id

# Детали
detail = [
    Detail(1, 'желоб', 1500, 1),
    Detail(2, 'днище', 1000, 3),
    Detail(3, 'карниз', 12500, 2),
    Detail(4, 'корпус маленький', 2200, 5),
```

```

Detail(5, 'корпус большой', 4200, 3),
Detail(6, 'корпус средний', 3200, 4),
Detail(7, 'лента', 500, 1),
Detail(8, 'вал', 700, 4),
Detail(9, 'клапан', 250, 5),
]

# Поставщик
agent = [
    Agent(1, 'Авторусь'),
    Agent(2, '24Зап'),
    Agent(3, 'ААА-Авто'),
    Agent(4, 'Восход'),
    Agent(5, 'Империя Авто'),

    Agent(11, 'Вира плюс'),
    Agent(22, 'Инверс-Авто'),
    Agent(33, 'Росско'),
    Agent(44, 'Авто-закуп'),
    Agent(55, 'Гуд-Авто'),
]

Ag_Det = [
    AgDet(1, 1),
    AgDet(2, 3),
    AgDet(3, 2),
    AgDet(4, 5),
    AgDet(5, 3),
    AgDet(6, 4),
    AgDet(7, 1),
    AgDet(8, 4),
    AgDet(9, 5),

    AgDet(1, 11),
    AgDet(2, 33),
    AgDet(3, 22),
    AgDet(4, 55),
    AgDet(5, 33),
    AgDet(6, 44),
    AgDet(7, 11),
    AgDet(8, 44),
    AgDet(9, 55),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(d.name, d.cost, a.name)
                    for d in detail
                    for a in agent
                    if d.agent_id == a.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(a.name, ad.agent_id, ad.detail_id)
                          for a in agent
                          for ad in Ag_Det
                          if a.id == ad.agent_id]

    many_to_many = [(d.name, d.cost, agent_name)
                     for agent_name, agent_id, detail_id in many_to_many_temp
                     for d in detail if d.id == detail_id]

```

```

print('Задание A1')
A1 = []

for i in range(len(one_to_many)):
    A1.append([])
    if one_to_many[i][2].startswith("A"):
        A1[i].append(one_to_many[i][2])
        A1[i].append(one_to_many[i][0])
        A1[i].append(one_to_many[i][1])
    str = list(filter(None, A1))

print(str)

print('\nЗадание A2')
res_12_unsorted = []
buf = 0
# Перебираем всех поставщиков
for a in agent:
    # Список деталей поставщиков
    a_detail = list(filter(lambda i: i[2] == a.name, one_to_many))
    # Если агент не пустой
    if len(a_detail) > 0:
        # Стоимость деталей поставщиков
        a_cost = [cost for _, cost, _ in a_detail]
        # Максимальная стоимость деталей поставщика
        a_cost_max = max(a_cost)
        res_12_unsorted.append((a.name, a_cost_max))

# Сортировка по максимальной стоимости детали
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание A3')
# Сортировка по поставщику всех связанных деталей и поставщиков
res_13 = sorted(many_to_many, key=itemgetter(2))
print(res_13)

if __name__ == '__main__':
    main()

```

## Вывод:

Задание A1

```
[['Авторусь', 'желоб', 1500], ['AAA-Авто', 'днище', 1000], ['AAA-Авто', 'корпус большой', 4200], ['Автор
```

Задание A2

```
[('243ап', 12500), ('AAA-Авто', 4200), ('Восход', 3200), ('Империя Авто', 2200), ('Авторусь', 1500)]
```

Задание A3

```
[('карниз', 12500, '243ап'), ('днище', 1000, 'AAA-Авто'), ('корпус большой', 4200, 'AAA-Авто'), ('корпус
```