



**Министерство науки и высшего образования Российской
Федерации Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Базовые компоненты интернет-технологий»
Отчет по рубежному контролю №1**

Вариант № В-19

**Выполнил:
студент группы ИУ5-33Б
Требуков Д.А.**

Проверил:

2021 г.

Условия рубежного контроля №1 по курсу БКИТ

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

- 1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
- 2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
- 3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Для реализации запроса №2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».

Предметная область

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

Код программы

```
1  from operator import itemgetter
2
3
4  class Detail:
5      """Детали"""
6      def __init__(self, id, name, value, manufacturer_id):
7          self.id = id
8          self.name = name
9          self.value = value
10         self.manufacturer_id = manufacturer_id
11
12     class Manufacturer:
13         """Производители"""
14         def __init__(self, id, name):
15             self.id = id
16             self.name = name
17
18     class ManufactDetail:
19         """Детали от производителей"""
20         def __init__(self, manufacturer_id, detail_id):
21             self.manufacturer_id = manufacturer_id
22             self.detail_id = detail_id
23
```

```

23 #Производители
24 manufacturers = [
25     Manufacturer(1, 'Schlieckmann'),
26     Manufacturer(2, 'Klokkerholm'),
27     Manufacturer(3, 'Signeda'),
28
29
30     Manufacturer(11, 'Sigma'),
31     Manufacturer(22, 'Tyg'),
32     Manufacturer(33, 'Kito')
33 ]
34
35 #Детали
36 details = [
37     Detail(1, 'Кузов', 15000, 1),
38     Detail(2, 'Поршни', 20000, 1),
39     Detail(3, 'Ремень ГРМ', 5000, 2),
40     Detail(4, 'Подушки двигателя', 25000, 3),
41     Detail(5, 'Система охлаждения', 7000, 3)
42 ]
43
44 detail_manufacts = [
45     ManufactDetail(1, 1),
46     ManufactDetail(1, 2),
47     ManufactDetail(2, 3),
48     ManufactDetail(3, 4),
49     ManufactDetail(3, 5),
50
51     ManufactDetail(11, 1),
52     ManufactDetail(22, 2),
53     ManufactDetail(22, 3),
54     ManufactDetail(33, 4),
55     ManufactDetail(33, 5)
56 ]
57
58 def res_1(arr):
59     answer_list = []
60     for detail, value, manufacturer in arr:
61         if detail[0] == 'П':
62             answer_list.append([detail, manufacturer])
63     return answer_list
64
65 def res_2(arr):
66     answer_list = [[arr[0][2], arr[0][1]]]
67     for detail, value, manufacturer in arr:
68         if manufacturer == answer_list[len(answer_list)-1][0]:
69             if value < answer_list[len(answer_list)-1][1]:
70                 answer_list[len(answer_list)-1][1] = value
71             else:
72                 answer_list.append([manufacturer, value])
73     return sorted(answer_list, key=itemgetter(1))
74
75 def res_3(arr):
76     answer_list = []
77     for detial, value, manufacturer in arr:
78         answer_list.append([detial, manufacturer])
79     return sorted(answer_list, key=itemgetter(0))
80
81 def main():
82     # Соединение данных один-ко-многим
83     one_to_many = [(d.name, d.value, m.name)
84                     for m in manufacturers
85                     for d in details
86                     if d.manufacturer_id == m.id]

```

```

86 # Соединение данных многие-ко-многим
87 many_to_many_temp = [(m.name, md.manufacturer_id, md.detail_id)
88                       for m in manufacturers
89                       for md in detail_manufacs
90                       if m.id == md.manufacturer_id]
91
92 many_to_many = [(d.name, d.value, m_name)
93                 for m_name, m_id, d_id in many_to_many_temp
94                 for d in details if d.id == d_id]
95
96 print('Задание B1')
97 """
98     «Производитель» и «Деталь» связаны соотношением один-ко-многим.
99     Выведите список всех деталей, у которых названия
100     начинаются с буквы «П», и названия их производителей.
101 """
102 print(res_1(one_to_many))
103
104 print('Задание B2')
105 """
106     «Производитель» и «Деталь» связаны соотношением один-ко-многим.
107     Выведите список производителей с минимальной стоимостью деталей
108     у каждого производителя, отсортированный по минимальной стоимости.
109 """
110 print(res_2(one_to_many))
111
112 print('Задание B3')
113 """
114     «Производитель» и «Деталь» связаны соотношением многие-ко-многим.
115     Выведите список всех связанных производителей и деталей,
116     отсортированный по деталям, сортировка по производителям произвольная.
117 """
118 print(res_3(many_to_many))
119
120 if __name__ == "__main__":
121     main()
122

```

Результаты работы программы

```

C:\Users\trebu\OneDrive\университет\2 курс\Базовые компоненты интернет-технологий (лаб)>python main.py
Задание B1
[['Поршни', 'Schlieckmann'], ['Подушки двигателя', 'Signeda']]
Задание B2
[['Klokkerholm', 5000], ['Signeda', 7000], ['Schlieckmann', 15000]]
Задание B3
[['Кузов', 'Schlieckmann'], ['Кузов', 'Sigma'], ['Подушки двигателя', 'Signeda'], ['Подушки двигателя', 'Kito'],
 ['Поршни', 'Schlieckmann'], ['Поршни', 'Tyg'], ['Ремень ГРМ', 'Klokkerholm'], ['Ремень ГРМ', 'Tyg'], ['Система охлаждения', 'Signeda'], ['Система охлаждения', 'Kito']]

```

Задание B1

[['Поршни', 'Schlieckmann'], ['Подушки двигателя', 'Signeda']]

Задание B2

[['Klokkerholm', 5000], ['Signeda', 7000], ['Schlieckmann', 15000]]

Задание B3

[['Кузов', 'Schlieckmann'], ['Кузов', 'Sigma'], ['Подушки двигателя', 'Signeda'], ['Подушки двигателя', 'Kito'], ['Поршни', 'Schlieckmann'], ['Поршни', 'Tyg'], ['Ремень ГРМ', 'Klokkerholm'], ['Ремень ГРМ', 'Tyg'], ['Система охлаждения', 'Signeda'], ['Система охлаждения', 'Kito']]