

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

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

Отчет по рубежному контролю №1
по курсу «Разработка интернет-приложений».

Выполнил:
студент группы ИУ5-51Б
Плотников Ф.С.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2021 г

Задание

- 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.

Вариант A19

1. «Производитель» и «Деталь» связаны соотношением один-ко-многим. Выведите список всех связанных деталей и производителей, отсортированный по производителям, сортировка по деталям произвольная.
2. «Производитель» и «Деталь» связаны соотношением один-ко-многим. Выведите список деталей со средней выручкой производителей по каждой детали, отсортированный по средней выручке.
3. «Производитель» и «Деталь» связаны соотношением многие-ко-многим. Выведите список всех деталей, название которых длиннее 5 букв, и список производителей.

Текст программы:

```
1  from operator import itemgetter
2
3
4  class Producer:
5      """Производитель"""
6
7      def __init__(self, id, name, revenue, DET_id):
8          self.id = id
9          self.name = name
10         self.revenue = revenue
11         """выручка в млн. руб., информация берётся из интернета"""
12         self.DET_id = DET_id
13         """Одна из производимых деталей"""
14
15
16  class Detail:
17      """Детали"""
18
19      def __init__(self, id, name):
20          self.id = id
21          self.name = name
22
23
24  class ProducerDET:
25      """Производители по деталям"""
26
27      def __init__(self, DET_id, PROD_id):
28          self.DET_id = DET_id
29          self.PROD_id = PROD_id
30
```

```
31
32  # Детали
33  DETs = [
34      Detail(1, 'Shaft'),
35      Detail(2, 'Key'),
36      Detail(3, 'Sheave'),
37      Detail(4, 'Sprocket'),
38      Detail(5, 'Gear'),
39      Detail(6, 'Nut'),
40      Detail(7, 'Crankshaft')
41  ]
42
43  # Производители
44  PRODs = [
45      Producer(1, 'Metmash', 428, 1),
46      Producer(2, 'Sarzni', 8, 6),
47      Producer(3, 'Szinw', 260, 2),
48      Producer(4, 'Avtodetal', 3084, 1),
49      Producer(5, 'Oameh', 95, 4),
50      Producer(6, 'Milleavorselmash', 140, 3),
51      Producer(7, 'Chelzto', 20, 3),
52      Producer(8, 'Mehz', 425, 6),
53      Producer(9, 'Omzavod', 1037, 7),
54      Producer(10, 'Demz', 180, 3),
55      Producer(11, 'KZM', 40, 5),
56      Producer(12, 'Chzm', 127, 2)
57  ]
```

```

58
59     # Производители по деталям
60     PROD_in_DET = [
61         ProducerDET(1, 1),
62         ProducerDET(1, 4),
63         ProducerDET(2, 3),
64         ProducerDET(2, 12),
65         ProducerDET(3, 6),
66         ProducerDET(3, 7),
67         ProducerDET(3, 10),
68         ProducerDET(4, 5),
69         ProducerDET(5, 11),
70         ProducerDET(6, 2),
71         ProducerDET(6, 8),
72         ProducerDET(7, 9),
73
74         ProducerDET(1, 12),
75         ProducerDET(2, 6),
76         ProducerDET(3, 9),
77         ProducerDET(4, 1),
78         ProducerDET(5, 12),
79     ]
80
81

```

```

82 def main():
83     """Основная функция"""
84
85     # Соединение данных один-ко-многим
86     one_to_many = [(prod.name, prod.revenue, det.name)
87                     for prod in PRODs
88                     for det in DETs
89                     if prod.DET_id == det.id]
90
91     print('\n\n\nЗадание A1')
92     # сортировка по возрастанию выручки производителей
93     print(sorted(one_to_many, key=itemgetter(1)))
94
95     # Соединение данных многие-ко-многим
96     many_to_many_temp = [(det.name, proddet.DET_id, proddet.PROD_id)
97                           for det in DETs
98                           for proddet in PROD_in_DET
99                           if det.id == proddet.DET_id]
100
101     print('\n\nЗадание A2')
102     # Т.к. была взята выручка, то сменил на среднюю выручку
103     res_12_unsorted = []
104     # Считаем среднюю выручку производителей
105     for det in DETs:
106         # Сумма выручек производителей
107         DET_sum = sum(prod.revenue for prod in PRODs if (det.id == prod.DET_id))
108         # Количество производителей для детали
109         DET_number_of_prods = len(list(filter(lambda i: i[2] == det.name, one_to_many)))
110         # Средняя выручка производителей, занимающихся производством этой детали
111         DET_average = DET_sum / DET_number_of_prods

```

```

112     res_12_unsorted.append((det.name, DET_average))
113     # Сортировка по выручке
114     print(sorted(res_12_unsorted, key=itemgetter(1)))
115
116     many_to_many = [(prod.name, prod.revenue, DET_name)
117                     for DET_name, DET_id, prod_id in many_to_many_temp
118                     for prod in PRODs
119                     if prod.id == prod_id]
120
121     print('\nЗадание A3')
122     # вывод Производителей для деталей, названия которых длиннее 5 букв
123     res_13 = {}
124     for det in DETs:
125         if len(det.name) > 5:
126             # Список деталей
127             det_in_prod = list(filter(lambda i: i[2] == det.name, many_to_many))
128             # Оставляем только названия
129             prod_in_det_names = [name for name, _, _ in det_in_prod]
130             # Добавляем результат в словарь
131             # ключ - деталь, значение - список производителей
132             res_13[det.name] = prod_in_det_names
133     print(res_13)
134
135
136 if __name__ == '__main__':
137     main()

```

Вывод программы:

```

Задание A1
[('Sarzni', 8, 'Nut'), ('Chelzto', 20, 'Sheave'), ('KZM', 40, 'Gear'), ('Ogmeh', 95, 'Sprocket'), ('Chzm', 127, 'Key'), ('Millevorselmash', 140, 'Sheave'), ('Demz', 180, 'Sheave'), ('Szinw', 260, 'Key'), ('Mehz', 425, 'Nut'), ('Metmash', 428, 'Shaft'), ('Ompzavod', 1037, 'Crankshaft'), ('Avtodetal', 3084, 'Shaft')]

Задание A2
[('Gear', 40.0), ('Sprocket', 95.0), ('Sheave', 113.33333333333333), ('Key', 193.5), ('Nut', 216.5), ('Crankshaft', 1037.0), ('Shaft', 1756.0)]

Задание A3
{'Sheave': ['Millevorselmash', 'Chelzto', 'Demz', 'Ompzavod'], 'Sprocket': ['Ogmeh', 'Metmash'], 'Crankshaft': ['Ompzavod']}

```