

**Московский государственный технический
университет 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 №3 (А)

Выполнил:
студент группы ИУ5-51Б
Аляев Д. В.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2021 г.

Задание:

1. «Автопарк» и «Водитель» связаны соотношением один-ко-многим. Выведите список всех связанных водителей и автопарков, отсортированный по отделам, сортировка по сотрудникам произвольная.
2. «Автопарк» и «Водитель» связаны соотношением один-ко-многим. Выведите список автопарков с суммарной зарплатой водителей в каждом автопарке, отсортированный по суммарной зарплате.
3. «Автопарк» и «Водитель» связаны соотношением многие-ко-многим. Выведите список всех автопарков, у которых в названии присутствует слово «авто», и список работающих в них водителей.

Текст программы:

```
# используется для сортировки
from operator import itemgetter

class Driver:
    """Водитель"""

    def __init__(self, id, fio, sal, Autopark_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.Autopark_id = Autopark_id

class Autopark:
    """Автопарк"""

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

class DriverAutopark:
    """
    'Водители автопарка' для реализации
    связи многие-ко-многим
    """

    def __init__(self, Autopark_id, Driver_id):
        self.Autopark_id = Autopark_id
        self.Driver_id = Driver_id

# Отделы
autoparks = [
    Autopark(1, 'Goodway'),
    Autopark(2, 'Экспресс-авто'),
    Autopark(3, 'Автопарк-Гарант'),

    Autopark(11, 'Goodway (другой)'),
    Autopark(22, 'Экспресс-авто (другой)'),
    Autopark(33, 'Автопарк-Гарант (другой)'),
]

# Сотрудники
drivers = [
    Driver(1, 'Артамонов', 25000, 1),
```

```

Driver(2, 'Петров', 35000, 2),
Driver(3, 'Иваненко', 45000, 3),
Driver(4, 'Иванов', 35000, 3),
Driver(5, 'Иванин', 25000, 3),
]

drivers_autoparks = [
    DriverAutopark(1, 1),
    DriverAutopark(2, 2),
    DriverAutopark(3, 3),
    DriverAutopark(3, 4),
    DriverAutopark(3, 5),

    DriverAutopark(11, 1),
    DriverAutopark(22, 2),
    DriverAutopark(33, 3),
    DriverAutopark(33, 4),
    DriverAutopark(33, 5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(e.fio, e.sal, d.name)
                   for d in autoparks
                   for e in drivers
                   if e.Autopark_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_tDriver = [(d.name, ed.Autopark_id, ed.Driver_id)
                             for d in autoparks
                             for ed in drivers_autoparks
                             if d.id == ed.Autopark_id]

    many_to_many = [(e.fio, e.sal, Autopark_name)
                    for Autopark_name, Autopark_id, Driver_id in
many_to_many_tDriver
                    for e in drivers if e.id == Driver_id]

    print('Задание A1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    print('\nЗадание A2')
    res_12_unsorted = []
    # Перебираем все автопарки
    for d in autoparks:
        # Список водителей автопарка
        d_drivers = list(filter(lambda i: i[2] == d.name, one_to_many))
        # Если автопарк не пустой
        if len(d_drivers) > 0:
            # Зарплаты водителей автопарка
            d_sals = [sal for _, sal, _ in d_drivers]
            # Суммарная зарплата водителей автопарка
            d_sals_sum = sum(d_sals)
            res_12_unsorted.append((d.name, d_sals_sum))

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

    print('\nЗадание A3')
    res_13 = {}
    # Перебираем все автопарки
    for d in autoparks:
        # Ищем слово "авто" в названии автопарка вне зависимости от регистра

```

```

        if 'авто' in d.name.lower():
            # Список водителей автопарка
            d_drivers = list(filter(lambda i: i[2] == d.name, many_to_many))
            # Только ФИО водителей
            d_drivers_names = [x for x, _, _ in d_drivers]
            # Добавляем результат в словарь
            # ключ - автопарк, значение - список фамилий
            res_13[d.name] = d_drivers_names

    print(res_13)

if __name__ == '__main__':
    main()

```

Результаты программы:

Задание A1

```

[('Артамонов', 25000, 'Goodway'), ('Иваненко', 45000, 'Автопарк-Гарант'),
 ('Иванов', 35000, 'Автопарк-Гарант'), ('Иванин', 25000, 'Автопарк-Гарант'),
 ('Петров', 35000, 'Экспресс-авто')]

```

Задание A2

```

[('Автопарк-Гарант', 105000), ('Экспресс-авто', 35000), ('Goodway', 25000)]

```

Задание A3

```

{'Экспресс-авто': ['Петров'], 'Автопарк-Гарант': ['Иваненко', 'Иванов',
 'Иванин'], 'Экспресс-авто (другой)': ['Петров'], 'Автопарк-Гарант (другой)':
 ['Иваненко', 'Иванов', 'Иванин']}

```

Process finished with exit code 0