

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

**Факультет «Радиотехнический»
Кафедра «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №1

Выполнил:
студент группы РТ5-31Б
Бондарь М. В.

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий
Евгеньевич

Москва, 2025 г.

Текст программы

```
from operator import itemgetter

class Driver:
    """Водитель"""
    def __init__(self, id, fio, salary, park_id):
        self.id = id
        self.fio = fio
        self.salary = salary
        self.park_id = park_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, park_id, driver_id):
        self.park_id = park_id
        self.driver_id = driver_id

# Автопарки
autoparks = [
    Autopark(1, 'Городской Автопарк'),
    Autopark(2, 'Логистик Транс'),
    Autopark(3, 'Сити Мобил Парк'),
]

# Водители
drivers = [
    Driver(1, 'Алексеев', 50000, 1),
    Driver(2, 'Петров', 80000, 2),
    Driver(3, 'Сидоров', 45000, 2),
    Driver(4, 'Иванов', 35000, 3),
    Driver(5, 'Смирнов', 40000, 3),
]

# Связь многие-ко-многим
drivers_autoparks = [
    DriverAutopark(1, 1), # Алексеев в Городском
    DriverAutopark(2, 2), # Петров в Логистик
    DriverAutopark(2, 3), # Сидоров в Логистик
    DriverAutopark(3, 4), # Иванов в Сити
    DriverAutopark(3, 5), # Смирнов в Сити
    DriverAutopark(3, 2), # Петров подрабатывает в Сити (пример связи
М:М)
```

```

]

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

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

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(p.name, dp.park_id, dp.driver_id)
                          for p in autoparks
                          for dp in drivers_autoparks
                          if p.id == dp.park_id]

    many_to_many = [(d.fio, d.salary, park_name)
                     for park_name, park_id, driver_id in
many_to_many_temp
                     for d in drivers if d.id == driver_id]

    print('Задание E1')
    # Выведите список всех автопарков, у которых в названии присутствует
слово «Парк»,
    # и список работающих в них водителей.
    res_11 = {}
    for p in autoparks:
        if 'Парк' in p.name:
            d_s = [d.fio for d in drivers if d.park_id == p.id]
            res_11[p.name] = d_s

    # Вывод результата
    for park, workers in res_11.items():
        print(f'{park}: {" ".join(workers)}')

    print('\nЗадание E2')
    # Выведите список автопарков со средней зарплатой водителей в каждом
автопарке,
    # отсортированный по средней зарплате.
    res_12_unsorted = []
    # Перебираем все автопарки
    for p in autoparks:
        # Список зарплат водителей этого парка
        d_salaries = list(filter(lambda i: i[2] == p.name, one_to_many))

        if len(d_salaries) > 0:
            # Зарплаты
            sals = [salary for _, salary, _ in d_salaries]
            # Средняя зарплата
            avg_salary = round(sum(sals) / len(sals), 2)
            res_12_unsorted.append((p.name, avg_salary))

```

```

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

print('\nЗадание E3')
# Выведите список всех водителей, у которых фамилия начинается с
буквы «П»,
# и названия их автопарков (Многие-ко-многим).
res_13 = {}
# Получаем всех водителей на букву П
target_drivers = [d for d in drivers if d.fio.startswith('П')]

for d in target_drivers:
    # Ищем парки для водителя
    d_parks = list(filter(lambda i: i[0] == d.fio, many_to_many))
    d_parks_names = [x for _, x in d_parks]
    res_13[d.fio] = d_parks_names

print(res_13)

if __name__ == '__main__':
    main()

```

Вывод программы

Задание E1

Сити Мобил Парк: Иванов, Смирнов

Задание E2

[('Логистик Транс', 62500.0), ('Городской Автопарк', 50000.0), ('Сити Мобил Парк', 37500.0)]

Задание E3

{'Петров': ['Логистик Транс', 'Сити Мобил Парк']}

Process finished with exit code 0