

## Пример текста программы:

```
# используется для сортировки
```

```
from operator import itemgetter
```

```
class Driver:
```

```
    """Водитель"""
```

```
    def __init__(self, id, name, salary, garage_id):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.salary = salary # Произвольный количественный признак (зарплата)
```

```
        self.garage_id = garage_id
```

```
class Garage:
```

```
    """Автопарк"""
```

```
    def __init__(self, id, name):
```

```
        self.id = id
```

```
        self.name = name
```

```
class DriverGarage:
```

```
    """Водители автопарка' для реализации связи многие-ко-многим"""
```

```
    def __init__(self, driver_id, garage_id):
```

```
        self.driver_id = driver_id
```

```
        self.garage_id = garage_id
```

```
def main():
```

```
    """Основная функция"""
```

```
    # Автопарки
```

```
    garages = [
```

```
Garage(1, "Альтернативный автопарк"),  
Garage(2, "Северный автопарк"),  
Garage(3, "Южный автопарк"),  
Garage(4, "Западный автопарк"),  
Garage(5, "Восточный автопарк"),  
]
```

# Водители

```
drivers = [  
    Driver(1, "Иванов", 40000, 1),  
    Driver(2, "Петров", 45000, 2),  
    Driver(3, "Сидоров", 38000, 3),  
    Driver(4, "Козлов", 50000, 1),  
    Driver(5, "Новиков", 42000, 4),  
    Driver(6, "Волков", 39000, 2),  
    Driver(7, "Овчинников", 41000, 5),  
    Driver(8, "Медведев", 44000, 3),  
]
```

# Связь многие-ко-многим

```
drivers_garages = [  
    DriverGarage(1, 1),  
    DriverGarage(2, 2),  
    DriverGarage(3, 3),  
    DriverGarage(4, 1),  
    DriverGarage(5, 4),  
    DriverGarage(6, 2),  
    DriverGarage(7, 5),  
    DriverGarage(8, 3),  
]
```

```

# Добавим еще пару связей для демонстрации

DriverGarage(1, 2), # Иванов работает и в Центральном, и в Северном

DriverGarage(3, 5), # Сидоров работает и в Южном, и в Восточном

]

print("--- Исходные данные ---")

print("Автопарки:")

for g in garages:

    print(f" ID: {g.id}, Название: {g.name}")

print("\nВодители:")

for d in drivers:

    print(f" ID: {d.id}, ФИО: {d.name}, Зарплата: {d.salary}, ID Автопарка: {d.garage_id}")

print("\nСвязи Водитель-Автопарк:")

for dg in drivers_garages:

    print(f" ID Водителя: {dg.driver_id}, ID Автопарка: {dg.garage_id}")


# Соединение данных один-ко-многим

one_to_many = [(d.name, d.salary, g.name, g.id)

                for g in garages

                for d in drivers

                if d.garage_id == g.id]


# Соединение данных многие-ко-многим

many_to_many_temp = [(g.name, g.id, dg.driver_id)

                      for g in garages

                      for dg in drivers_garages

                      if g.id == dg.garage_id]

many_to_many = [(d.name, d.salary, garage_name, d.id)

                 for garage_name, garage_id, driver_id in many_to_many_temp

```

```
for d in drivers
    if d.id == driver_id]
```

```
print("\n\n--- Выполнение запросов ---")
```

# Задание Д1: Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.

```
print("\nЗадание Д1: Водители, фамилия которых заканчивается на 'ов', и их автопарки")
```

```
res_1 = [(d_name, garage_name)
          for d_name, _, garage_name, _ in one_to_many
          if d_name.endswith('ов')]
```

```
# Сортировка по фамилии водителя
```

```
res_1_sorted = sorted(res_1, key=itemgetter(0))
print(res_1_sorted)
```

# Задание Д2: Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате

```
print("\nЗадание Д2: Автопарки с средней зарплатой водителей, отсортированные по средней зарплате")
```

```
garage_avg_salaries_unsorted = []
```

```
for g in garages:
```

```
    # Список водителей автопарка g
```

```
    g_drivers = list(filter(lambda x: x[2] == g.name, one_to_many))
```

```
    if len(g_drivers) > 0:
```

```
        # Зарплаты водителей автопарка
```

```
        g_salaries = [salary for _, salary, _, _ in g_drivers]
```

```
        # Средняя зарплата
```

```
        avg_salary = sum(g_salaries) / len(g_salaries)
```

```
        garage_avg_salaries_unsorted.append((g.name, round(avg_salary, 2)))
```

```
# Сортировка по средней зарплате
```

```
garage_avg_salaries_sorted = sorted(garage_avg_salaries_unsorted, key=itemgetter(1))
```

```
print(garage_avg_salaries_sorted)
```

# Задание Д3: Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

```
print("\nЗадание Д3: Автопарки, название которых начинается с 'А', и их водители")
```

```
res_3 = {}
```

```
for g in garages:
```

```
    if g.name.startswith('А'):
```

```
        # Список водителей автопарка g через many_to_many связь
```

```
        g_drivers = list(filter(lambda x: x[2] == g.name, many_to_many))
```

```
        # Только ФИО водителей
```

```
        driver_names = [name for name, _, _ in g_drivers]
```

```
        # Добавляем результат в словарь
```

```
        res_3[g.name] = driver_names
```

```
print(res_3)
```

```
if __name__ == "__main__":
```

```
    main()
```

## **Исходные данные:**

Автопарки:

ID: 1, Название: Альтернативный автопарк

ID: 2, Название: Северный автопарк

ID: 3, Название: Южный автопарк

ID: 4, Название: Западный автопарк

ID: 5, Название: Восточный автопарк

#### Водители:

ID: 1, ФИО: Иванов, Зарплата: 40000, ID Автопарка: 1

ID: 2, ФИО: Петров, Зарплата: 45000, ID Автопарка: 2

ID: 3, ФИО: Сидоров, Зарплата: 38000, ID Автопарка: 3

ID: 4, ФИО: Козлов, Зарплата: 50000, ID Автопарка: 1

ID: 5, ФИО: Новиков, Зарплата: 42000, ID Автопарка: 4

ID: 6, ФИО: Волков, Зарплата: 39000, ID Автопарка: 2

ID: 7, ФИО: Овчинников, Зарплата: 41000, ID Автопарка: 5

ID: 8, ФИО: Медведев, Зарплата: 44000, ID Автопарка: 3

#### Связи Водитель-Автопарк:

ID Водителя: 1, ID Автопарка: 1

ID Водителя: 2, ID Автопарка: 2

ID Водителя: 3, ID Автопарка: 3

ID Водителя: 4, ID Автопарка: 1

ID Водителя: 5, ID Автопарка: 4

ID Водителя: 6, ID Автопарка: 2

ID Водителя: 7, ID Автопарка: 5

ID Водителя: 8, ID Автопарка: 3

ID Водителя: 1, ID Автопарка: 2

ID Водителя: 3, ID Автопарка: 5

## **Выполнение запросов**

Задание Д1: Водители, фамилия которых заканчивается на 'ов', и их автопарки

[('Волков', 'Северный автопарк'), ('Иванов', 'Альтернативный автопарк'), ('Козлов', 'Альтернативный автопарк'), ('Новиков', 'Западный автопарк'), ('Овчинников', 'Восточный автопарк'), ('Петров', 'Северный автопарк'), ('Сидоров', 'Южный автопарк')]

Задание Д2: Автопарки с средней зарплатой водителей, отсортированные по средней зарплате

[('Южный автопарк', 41000.0), ('Восточный автопарк', 41000.0), ('Северный автопарк', 42000.0), ('Западный автопарк', 42000.0), ('Альтернативный автопарк', 45000.0)]

Задание Д3: Автопарки, название которых начинается с 'А', и их водители

{'Альтернативный автопарк': ['Иванов', 'Козлов']}