

Отчет по контрольной работе

Описание проекта

В рамках контрольной работы реализован программный код для обработки данных о водителях и автопарках. Основная задача заключается в создании классов для представления информации о водителях и автопарках, а также выполнения различных операций над этими данными в соответствии с условием задания.

Условие задания

1. Создать два класса данных, связанных отношениями один-ко-многим и многие-ко-многим.
 - Классы должны содержать соответствующие идентификаторы и атрибуты в соответствии с предметной областью.
2. Создать списки объектов классов с тестовыми данными (3-5 записей) и настроить связи между ними.
3. Разработать три запроса для обработки данных:
 - Список объектов с их связями, отсортированный по одному из атрибутов.
 - Список объектов с суммарным количественным признаком, отсортированный по этому признаку.
 - Список объектов с фильтрацией по определённому условию.

Классы и их назначение

Класс `Driver`

Представляет информацию о водителе.

Поля:

- `driver_id`: ID водителя (уникальный идентификатор).
- `name`: Имя водителя.
- `experience`: Стаж водителя (в годах).
- `fleet_id`: ID автопарка, к которому относится водитель.

Класс `Fleet`

Представляет информацию об автопарке.

Поля:

- `fleet_id`: ID автопарка (уникальный идентификатор).
- `name`: Название автопарка.

Класс `DriverFleet`

Реализует связь "многие-ко-многим" между водителями и автопарками.

Поля:

- `driver_id`: ID водителя.
- `fleet_id`: ID автопарка.

Данные

Водители

```
[
    Driver(1, "Иванов", 5, 1),
    Driver(2, "Петров", 10, 1),
    Driver(3, "Сидоров", 3, 2),
    Driver(4, "Кузнецов", 8, 3),
    Driver(5, "Алексеев", 6, 2),
]
```

Автопарки

```
[
    Fleet(1, "Центральный автопарк"),
    Fleet(2, "Южный автопарк"),
    Fleet(3, "Северный автопарк"),
]
```

Связь многие-ко-многим

```
[
    DriverFleet(1, 1),
    DriverFleet(2, 1),
    DriverFleet(3, 2),
    DriverFleet(4, 3),
]
```

```
DriverFleet(5, 2),  
]
```

Реализованные функции

1. Список водителей по автопаркам

Функция `list_drivers_by_fleet` сортирует автопарки по названию и выводит список водителей, принадлежащих каждому автопарку.

Пример вывода:

```
Центральный автопарк: Иванов, Петров  
Южный автопарк: Сидоров, Алексеев  
Северный автопарк: Кузнецов
```

2. Список автопарков по суммарному стажу водителей

Функция `list_fleets_by_experience` вычисляет суммарный стаж водителей для каждого автопарка, сортирует автопарки по этому показателю и выводит результат.

Пример вывода:

```
Центральный автопарк: 15 лет стажа  
Южный автопарк: 9 лет стажа  
Северный автопарк: 8 лет стажа
```

3. Список автопарков с водителями, в названии которых есть "автопарк"

Функция `list_fleets_with_drivers_containing_word` находит автопарки, в названии которых содержится слово "автопарк", и выводит список водителей, связанных с ними.

Пример вывода:

```
Центральный автопарк: Иванов, Петров  
Южный автопарк: Сидоров, Алексеев  
Северный автопарк: Кузнецов
```

Выводы

В результате работы было реализовано:

- Моделирование данных о водителях и автопарках с использованием классов.
- Запросы для получения информации о водителях и автопарках.
- Использование сортировок и фильтрации для обработки данных.

Код успешно решает поставленные задачи и демонстрирует навыки работы с объектами и коллекциями в Python.