Задача.

Дан ООП часть кода, который описывает бизнес-логику загрузки в некую базу данных о собаках.

Не имеет значения куда и откуда. Поставляется список собак с их свойствами: порода, цвет, картинка, имя, имя собственника, возраст. Сам ООП написан с избытком.

Требуется:

- почистить ООП код, оставить только непосредственный код для данного процесса

- определить паттерны проектирования, которые здесь использованы, ответить на вопрос - зачем и почему их здесь применили и нужны ли они вообще (последнее будет исходить от вашей чистки кода)

- добавить клиентский код. На вход построить загрузку данных из CSV  файла. По ходу ООП реализовать функции profile() и findDog(). Ограничений в количестве клиентского кода нет (зависит от вашего времени)

- запустить код (по возможности)

- нарисовать UML-диаграмму

В примере используются 2 паттерна.

Легковес (Flyweight).

Структурный шаблон проектирования, который позволяет сократить использование памяти приложением путем хранения "общего" состояния объектов только один раз, вместо хранения этого состояния в каждом из создаваемых объектов.

*Используется для хранения общих свойств, характерных для породы собак.*

Фабричный метод (Factory Method).

Порождающий шаблон. Его сутью является возможность предоставить потомкам класса каждому по-своему реализовывать метод, создающий какой-либо объект.

*Изначально был интерфейс, который предполагал реализовать метод fill().* В контексте задачи, мне показалось это лишним.