

Паттерн “Flyweight”

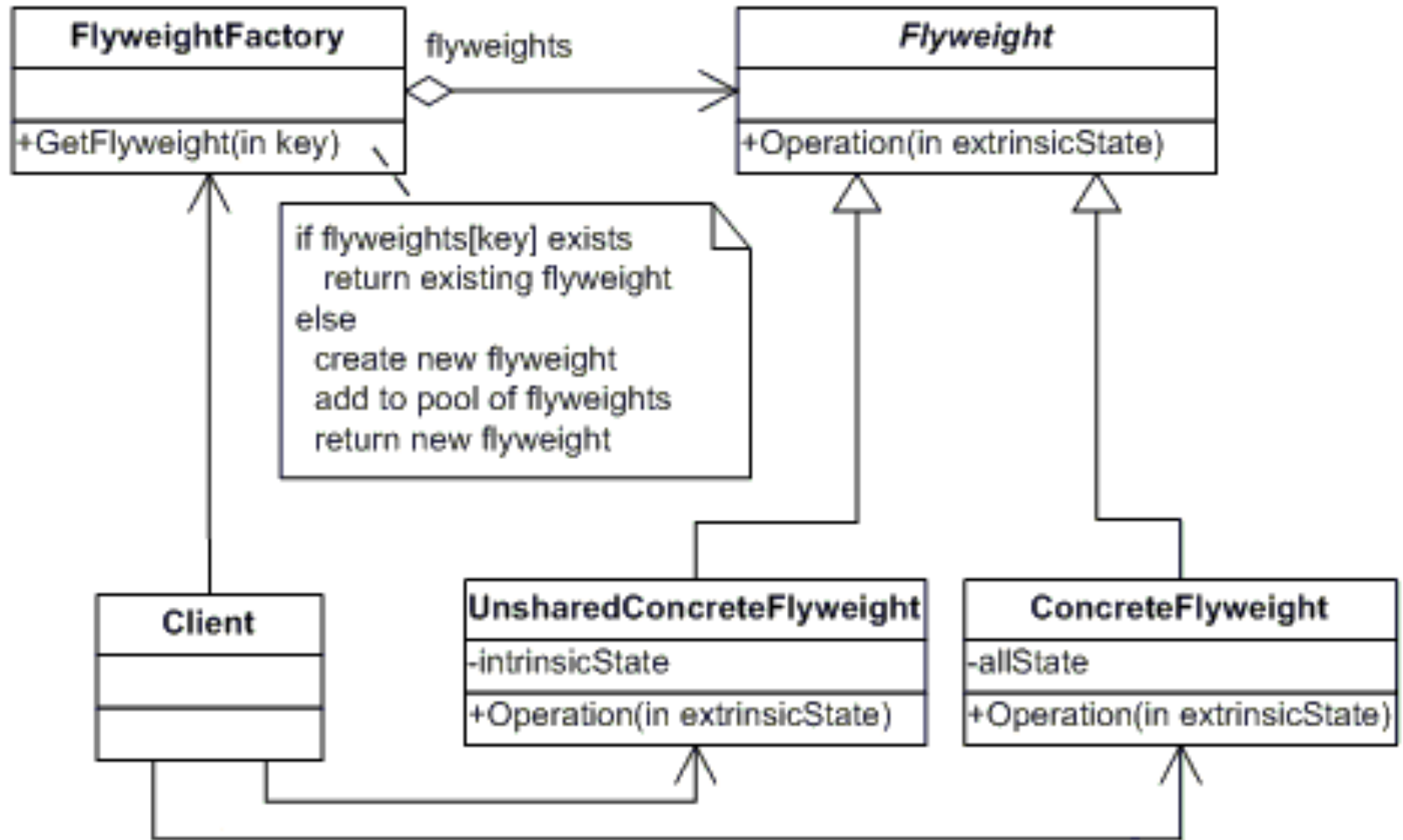
Тип паттерна

- Flyweight — структурный шаблон проектирования, т.е. образующий из классов/объектов более крупную структуру
- Представляет собой объект (для клиента выглядит как фабрика объектов)

Назначение паттерна

- Оптимизировать затраты при работе с большим количеством мелких объектов
- Представлять объекты уникальными в разных частях программы, когда они таковыми не являются

Красивая картинка



Применять, если

- Используется большое число схожих объектов
- Часть состояния объектов можно вынести во внешний интерфейс
- Приложение не проверяет идентичность объектов

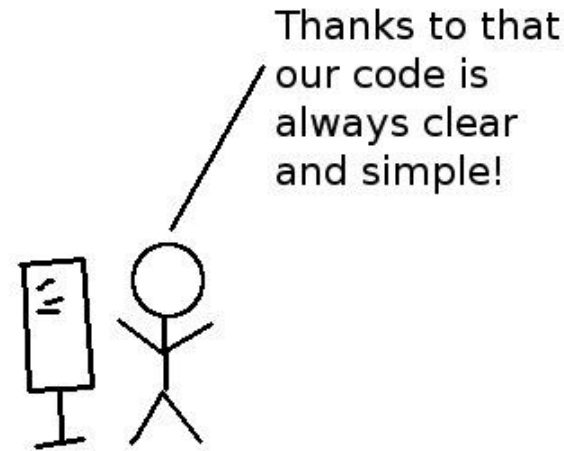
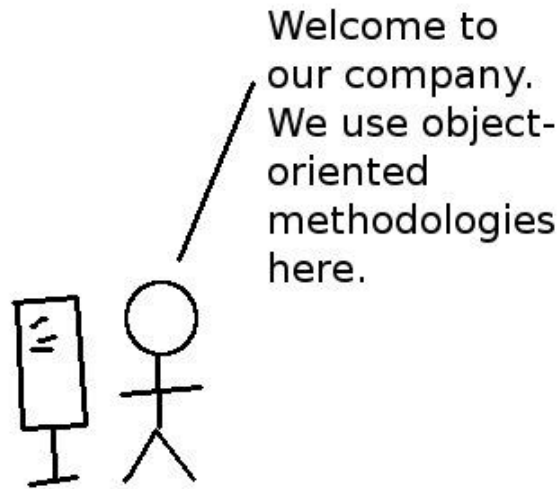
Особенности

- Является комбинацией паттернов Object Pool и Abstract Factory (Singleton - вариативно)
- Разделяет внешнее и внутреннее состояние объекта-приспособленца в зависимости от контекста
- Представляет объекты уникальными, что при неосторожном использовании может вылиться в различные сайд-эффекты

Примеры использования

- Пул различных экранов приложения
- Хранение символов в текстовом редакторе
- Service-oriented architecture (SOA)

Смешная картинка



Sample code:
Singleton::GetInstance()-
>GetFactory()->Obj
().GetWrapper().GetProxy(0)-
>GetAdapter
(Manager::GetInstance()-
>GetObjectIndexByName(s))-
>AccessState().GetParam
(PARAM_ENUM_NAME)-
>AsString(true)

