# Описание паттерна

Паттерн легковес/приспособленец (Flyweight) описывает, как совместно разделять очень мелкие объекты без чрезмерно высоких издержек. Каждый объект-приспособленец имеет две части: внутреннее и внешнее состояния. Внутреннее состояние хранится (разделяется) в приспособленце и состоит из информации, не зависящей от его контекста. Внешнее состояние хранится или вычисляется объектами-клиентами и передается приспособленцу при вызове его методов.

# Выбранный пример

При разработке компьютерных игр можно встретить много случаев, когда можно оптимизировать потребление памяти через структурный паттерн Flyweight. К таким примерам можно отнести создание юнитов в стратегии, количество видов которых ограничено, или создание «плиточного» ландшафта из определённого набора плиток.

Однако для данной работы был выбран пример игры-приложения, в котором участвует генерация леса из большого количества деревьев.

В игре есть определенное количество видов деревьев, поэтому на местности могут встречаться похожие деревья. Поэтому внутреннее состояние можно определить, как модель дерева (имя, скелет (состоящий из полигонов), масштаб, текстура). К внешнему состоянию можно отнести такие изменяющиеся параметры как позиция дерева на карте, угол поворота (если дерево спилили, например).

Применяя Flyweight мы оптимизируем память, создавая только определенное количество деревьев для возможного повторного использования вместо того, чтобы создавать тысячи деревьев в памяти со своими скелетами, текстурами и т.д.

# Сравнение с другими паттернами

Данный шаблон практически невозможно сравнивать с другими структурными паттернами, т.к. нет структурного паттерна, который применялся бы в схожих задачах по оптимизации памяти. Однако, среди порождающих паттернов есть паттерн, который как и Flyweight использует пул и решает задачу оптимизации потребления памяти – это Object Pool паттерн.

Паттерны Flyweight и Object Pool отличаются тем, как они используются. Объекты при применении Object Pool могут одновременно использоваться только одним клиентом. Объект должен быть извлечен из пула, затем клиент использует его, а затем клиент должен вернуть объект обратно в пул. Также может существовать несколько идентичных экземпляров объектов, вплоть до максимальной емкости пула.

Напротив, объект flyweight является одноэлементным, и он может использоваться одновременно несколькими клиентами. Объединенные в пул объекты Object Pool могут быть изменяемыми, и они обычно не должны быть потокобезопасными, поскольку, как правило, только один поток будет использовать конкретный экземпляр одновременно. Объекты flyweight в наилучшем варианте должны быть неизменяемыми (возможно, если flyweight объект изменяем, то скорее всего это уже не flyweight объект).

Шаблон Flyweight лучше подходит для поставленного примера нежели Object Pool, т.к. нет необходимости содержать в пуле идентичные экземпляры видов деревьев, а также потребуется большее количество клиентов для использования одного и того же объекта.