Паттерн Фасад

Для демонстрации паттерна «Фасад» было решено взять пример работы доставки пиццы. Но на самом деле, под такой шаблон подходит любое производство, способное принять заказ, выполнить его и доставить клиенту.

Почему именно «Фасад»?

Клиент, желая получить доставку пиццы домой (что особенно актуально сейчас) делает заказ в приложении либо звонит по указанным телефонам. Так как мы работаем с программированием, то разберём именно вариант с заказом в приложении.

При заказе клиент оформляет всё в приложении и хочет получить результат – привезённую домой пиццу. Ему не должно быть видно насколько поварам тяжело её готовить, сколько поваров для него работают, кто и в каких условиях её доставляет. Т.е. его необходимо изолировать от сложной системы передачи его заказа работникам, затем курьерам и т.п. Как раз для этого подходит шаблон «Фасад», он позволяет оставить для пользователя простой интерфейс, скрыв от него остальные, более сложные компоненты системы.

А вдруг подойдут другие шаблоны под задачу?

* Декоратор: не подойдёт, т.к. нам не нужно добавлять какие-либо дополнительные новые функции объектам.
* Адаптер: не подойдёт, т.к. не нужно определять новый интерфейс, нам достаточно использовать существующий.
* Компоновщик: не подойдёт, т.к. клиенту не нужно управлять частями классов, ему нужно, чтобы все методы классов были выполнены полностью.
* Посредник: не подойдёт, т.к. взаимодействующие объекты данного примера должны быть тесно связаны друг с другом. Посредник же обеспечивает слабосвязанность множества взаимосвязанных объектов.