**Лабораторная №5**

**4) Создайте документ Microsoft Word, содержащий пример приложения и аргументацию выбора шаблона проектирования.**

Класс Фасада предоставляет простой интерфейс для сложной логики одной или нескольких подсистем. Фасад делегирует запросы клиентов соответствующим объектам внутри подсистемы. Фасад также отвечает за управление их жизненным циклом. Все это защищает клиента от нежелательной сложности подсистемы.

Методы Фасада удобны для быстрого доступа к сложной функциональности подсистем. Однако клиенты получают только часть возможностей подсистемы.

Подсистема может принимать запросы либо от фасада, либо от клиента напрямую. В любом случае, для Подсистемы Фасад – это еще один клиент, и он не является частью Подсистемы.

Некоторые фасады могут работать с разными подсистемами одновременно.

Клиентский код работает со сложными подсистемами через простой интерфейс, предоставляемый Фасадом. Когда фасад управляет жизненным циклом подсистемы, клиент может даже не знать о существовании подсистемы. Такой подход позволяет держать сложность под контролем.

В клиентском коде могут быть уже созданы некоторые объекты подсистемы. В этом случае может оказаться целесообразным инициализировать Фасад с этими объектами вместо того, чтобы позволить Фасаду создавать новые экземпляры.

По сравнению с шаблонами «Цепочка обязанностей», «Команда», «Наблюдатель»

Цепочка обязанностей передаёт запрос последовательно через цепочку потенциальных получателей, ожидая, что какой-то из них обработает запрос.

Команда устанавливает косвенную одностороннюю связь от отправителей к получателям.

Наблюдатель передаёт запрос одновременно всем заинтересованным получателям, но позволяет им динамически подписываться или отписываться от таких оповещений.

У нашего фасада преимущество в том, что он создаёт упрощённый интерфейс к подсистеме, не внося в неё никакой добавочной функциональности. Сама подсистема не знает о существовании фасада. Классы подсистемы общаются друг с другом напрямую, так что из этих шаблонов фасад – единственный который этому соответствует.

И, например, почему мы используем не посредник, потому что он убирает отношения напрямую между каждым объектом системы, что роли не контактируют между собой кроме как через спортивного директора фасада.

Вывод: используем для реализации паттерн «Фасад». Собственно, как и указано в задании.