**Лабораторная №5**

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

Представим нам нужно компьютезировать процесс продажи футбольным клубом игрока(при условии, что все требования устраивают оба клуба. Здесь будет участвовать несколько участников:

1)Клуб, заинтересованный в игроке

2)Спортивный директор клуба, в котором находится нужный игрок

3)Агент игрока

4)Сам игрок

Обычно все эти роли взаимодействуют не друг с другом(не считаем нечестные сделки напрямую с игроком), а через спортивного директора, то есть он выполняет роль «посредника» между всеми ролями. При таком условии задачи для реализации подойдет шаблон проектирования «Посредник», так как тогда мы получим ряд преимуществ:

1)Не будет сильной связанности между ролями

2)Между объектами будет связь один-ко-всем, а не все-ко-всем, что упростит взаимодействие между ними

3)Само взаимодействие выносится в отдельный интерфейс, что помогает лучше понимать программу и менять ее в будущем

4)Становится проще управлять отношениями между объектами(Централизуется управление в одном месте)

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 посредника преимущество в том, что он убирает отношения напрямую между каждым объектом системы, как и сказано в задании, что роли не контактируют между собой кроме как через спортивного директора, так что из этих шаблонов посредник – единственный который этому соответствует.

И, например, почему мы используем не Фасад, потому 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. А посредник централизует общение между компонентами системы. Компоненты системы знают только о существовании Посредника, у них нет прямого доступа к другим компонентам.

Вывод: используем для реализации паттерн «Посредник». Собственно, как и указано в задании, куда б мы без него делись 😊.