Шаблон проектирования – команда.

Пример приложения – система «умный дом» (конкретно, пользовательская программа).

Например, «вечерняя программа» : включить свет на улице, начать обогрев дома.

Программа содержит в себе список команд, которые реализуют конкретные объекты (переключатель режима работы ламп, обогреватель), но сама программа не контактирует с ними напрямую.

Существует 4 паттерна, реализующих связь между отправителями запросов и получателями:

* Команда
  + Применяется, когда необходимо ставить операции в очередь, выполнять их по расписанию или передавать по сети.
* Цепочка обязанностей
  + Когда программа должна обрабатывать разнообразные запросы несколькими способами, но заранее неизвестно, какие конкретно запросы будут приходить и какие обработчики для них понадобятся.

(можно связать обработчиков в одну цепь и при получении запроса поочередно спрашивать каждого, не хочет ли он обработать запрос).

* + Не подходит, так как мы знаем, что включать освещение будет конкретная лампа.
* Посредник
  + Паттерн проектирования, который позволяет уменьшить связанность множества классов между собой, благодаря перемещению этих связей в один класс-посредник (отправитель и получатель общаются через посредника).
  + Не подходит, так как нет необходимости в общении лампы и обогревателя, нужна только связь программа -> лампа и программа -> обогреватель
* Наблюдатель
  + Применяется, когда после изменения состояния одного объекта требуется что-то сделать в других, но мы не знаем наперёд, какие именно объекты должны отреагировать.
  + Больше подходит для программы безопасности, то есть некоторые обработчики «подписаны» на датчики (например, на датчик дыма подписана система вызова пожарной бригады и система оповещения жильцов)