**Посредник** — это поведенческий паттерн проектирования, который позволяет уменьшить связанность множества классов между собой, благодаря перемещению этих связей в один класс-посредник.

Паттерн Посредник заставляет объекты общаться не напрямую друг с другом, а через отдельный объект-посредник, который знает, кому нужно перенаправить тот или иной запрос. Благодаря этому, компоненты системы будут зависеть только от посредника, а не от десятков других компонентов.

Когда используется паттерн Посредник?

Когда имеется множество взаимосвязаных объектов, связи между которыми сложны и запутаны.

Когда необходимо повторно использовать объект, однако повторное использование затруднено в силу сильных связей с другими объектами.

Рассмотрим реальный пример. Когда создаётся программное обеспечение, то в процессе создания участвует множество людей и отделов: заказчики, программисты из отдела разработки, тестировщики из отдела тестирования и так далее. Эти отделы и заказчики чаще всего взаимодействуют между собой не напрямую, а через менеджера проектов. То есть менеджер проектов выполняет роль посредника.

Класс менеджера - ManagerMediator в методе Send() проверяет, от кого пришло сообщение, и в зависимости от отправителя перенаправляет его другому объекту с помощью методов Notify(), определенных в классе Colleague.

Таким образом, применение паттерна Посредник дает нам следующие преимущества:

* Устраняется сильная связанность между объектами Colleague
* Упрощается взаимодействие между объектами: вместо связей по типу "все-ко-всем" применяется связь "один-ко-всем"
* Взаимодействие между объектами абстрагируется и выносится в отдельный интерфейс
* Централизуется управления отношениями между объектами

Изображение выглядит как снимок экрана

Автоматически созданное описание