Фасад — это структурный паттерн, который предоставляет простой (но урезанный) интерфейс к сложной системе объектов, библиотеке или фреймворку. **Применимость:** Паттерн часто встречается в клиентских приложениях, написанных на C++, которые используют классы-фасады для упрощения работы со сложными библиотеки или API. **Признаки применения паттерна:** Фасад угадывается в классе, который имеет простой интерфейс, но делегирует основную часть работы другим классам. Чаще всего, фасады сами следят за жизненным циклом объектов сложной системы.

Посредник — это поведенческий паттерн, который упрощает коммуникацию между компонентами системы. Посредник убирает прямые связи между отдельными компонентами, заставляя их общаться друг с другом через себя. **Применимость:** Пожалуй, самое популярное применение Посредника в C++ коде — это связь нескольких компонентов GUI одной программы.

Команда — это поведенческий паттерн, позволяющий заворачивать запросы или простые операции в отдельные объекты. Это позволяет откладывать выполнение команд, выстраивать их в очереди, а также хранить историю и делать отмену. **Применимость:** Паттерн можно часто встретить в C++ коде, особенно когда нужно откладывать выполнение команд, выстраивать их в очереди, а также хранить историю и делать отмену. **Признаки применения паттерна:** Классы команд построены вокруг одного действия и имеют очень узкий контекст. Объекты команд часто подаются в обработчики событий элементов GUI. Практически любая реализация отмены использует принципа команд.





