1. Паттерн используется для того, чтобы создать единый интерфейс (в широком понимании) для работы с некоторой компонентой системы. При этом человеку, который пользуется компонентой системы не нужно знать ее внутреннее устройство
2. Диаграмма



1. Крупные системы, разделенные между некоторым количеством программистов, либо просто некоторые крупные системы.
2. Должно быть явное разделение на доступное пользователю (внешнее) и не доступное (иначе следует использовать прокси). Из-за тонкостей с доступностью могут возникнуть проблемы и осложниться реализация, следовательно, необходимо использовать этот паттерн, когда это действительно необходимо. Легко обновлять и улучшать уже существующие методы паттерна, так как все они задействованы только во внутренней части (при это легко только тому человеку, который ее писал). Сложно обновлять интерфейс фасада (если не добавлять адаптер, что может привести к адаптеру на адаптере и так далее) и добавлять новые функции в компоненту. Необходима высокая координация (связь) между программистами.
3. Класс Math C# является явной реализацией фасада. Все математические операции реализуются через статические методы класса Math, либо его экземпляр. При этом мы совершенно не знаем, как он устроен внутри. Другой пример был приведен мной в виде программы.