##### Try (Если выдает срабатывает Catch, если нет Catch не срабатывает) For example: если выдает исключение тогда к примеру catch пишет об ошибке. Catch Finally( Срабатывает в любом случае не зависимо выдало Try исключение или нет) Патерны: Singleton (одиночка)

##### Один из самых известных и, пожалуй, самых спорных паттернов. Представьте, что в городе требуется организовать связь между жителями. С одной стороны мы можем связать всех жителей между собой протянув между ними кабели телефонных линий, но полагаю вы понимаете насколько такая система неверна. Например, как затратно будет добавить еще одного жителя в связи (протянуть по еще одной линии к каждому жителю). Чтобы этого избежать, мы создаем телефонную станцию, которая и будет нашим «одиночкой». Она одна, всегда, и если кому-то потребуется связаться с кем-то, то он может это сделать через данную телефонную станцию, потому что все обращаются только к ней. Factory (фабрика)

Суть паттерна практически полностью описывается его названием. Когда вам требуется получать какие-то объекты, например пакеты сока, вам совершенно не нужно знать как их делают на фабрике. Вы просто говорите «сделайте мне пакет апельсинового сока», а «фабрика» возвращает вам требуемый пакет. Как? Всё это решает сама фабрика, например «копирует» уже существующий эталон. Основное предназначение «фабрики» в том, чтобы можно было при необходимости изменять процесс «появления» пакета сока, а самому потребителю ничего об этом не нужно было сообщать, чтобы он запрашивал его как и прежде.  
  
***Это те что я когда то юзал их разумеется больше***

## SQL The HEAD Method-Другими словами, если GET / users возвращает список пользователей, то HEAD / users выполнят тот же запрос, но не вернут список пользователей. Метод DELETE

**Метод DELETE удаляет указанный ресурс.**

## Метод OPTIONS

## **Метод OPTIONS описывает параметры связи для целевого ресурса.** Метод PUT

**PUT используется для отправки данных на сервер для создания / обновления ресурса.Освежил в памяти!)**