

## Применение паттернов

1. Singleton. Применяется тогда, когда должен быть ровно один экземпляр некоторого класса, легко доступный всем клиентам; единственный экземпляр должен расширяться путем порождения подклассов, и клиентам нужно иметь возможность работать с расширенным экземпляром без модификации своего кода.
2. Abstract Factory. Применяется в случае, когда система не должна зависеть от того, как создаются, komponуются и представляются входящие в неё объекты; входящие в семейство взаимосвязанные объекты должны использоваться вместе и вам необходимо обеспечить выполнение этого ограничения; система должна конфигурироваться одним из семейств составляющих её объектов; требуется предоставить библиотеку объектов, раскрывая только их интерфейсы, но не реализацию.
3. Builder. Применяется, когда алгоритм создания сложного объекта не должен зависеть от того, из каких частей состоит объект и как они стыкуются между собой; процесс конструирования должен обеспечивать различные представления конструируемого объекта.
4. Strategy. Используется, когда имеется много родственных классов, отличающихся только поведением (стратегия позволяет сконфигурировать класс, задав одно из возможных поведений); нужно иметь несколько разных вариантов алгоритма; в алгоритме содержатся данные, которых клиент не должен «знать».