

**Шаблон фасада** (от англ. Facade) - это шаблон проектирования программного обеспечения, обычно используемый в объектно-ориентированном программировании. По аналогии с фасадом в архитектуре, фасад — это объект, который служит фронтальным интерфейсом, маскирующим более сложный базовый или структурный код. Фасад может:

- улучшить удобочитаемость и удобство использования программной библиотеки, замаскировав взаимодействие с более сложными компонентами за единым (и часто упрощенным) API
- Предоставление контекстно-зависимого интерфейса для более общих функций (в комплекте с контекстно-зависимой проверкой ввода)
- служат отправной точкой для более широкого рефакторинга монолитных или тесно связанных систем в пользу более слабо связанного кода

Разработчики часто используют шаблон проектирования фасада, когда система очень сложна и трудна для понимания, потому что система имеет много взаимозависимых классов или потому что ее исходный код недоступен. Этот шаблон скрывает сложности более крупной системы и обеспечивает более простой интерфейс для клиента. Обычно он включает в себя один класс-оболочку, который содержит набор членов, необходимых клиенту. Эти участники получают доступ к системе от имени фасадного клиента и скрывают детали реализации.

Шаблон проектирования является одним из двадцати трех известных *шаблонов проектирования GoF*, которые описывают, как решать повторяющиеся проблемы проектирования для разработки гибкого и многократно используемого объектно-ориентированного программного обеспечения, то есть объектов, которые легче реализовать, изменить, протестировать и повторно использовать.

Какие проблемы может решить шаблон дизайна фасада?

- Чтобы сделать сложную подсистему более удобной в использовании, для набора интерфейсов в подсистеме должен быть предусмотрен простой интерфейс.
- Зависимости от подсистемы должны быть сведены к минимуму.

Клиенты, которые обращаются к сложной подсистеме, напрямую ссылаются (зависят от) на множество различных объектов, имеющих разные интерфейсы (тесная связь), что затрудняет реализацию, изменение, тестирование и повторное использование клиентов.