# Описание паттерна «Фасад»

Шаблон «фасад» - структурный шаблон проектирования, позволяющий скрыть сложность системы путём сведения всех возможных вызовов к одному объекту.

Фасад – внешний объект, обеспечивающий общую точку доступа для всех подсистем сложной системы. Данный объект предоставляет упрощенный унифицированный интерфейс для работы с системой без необходимости обращения к внутренним подсистемам. Фасад устойчив к изменениям подсистем.

# Постановка задачи

Имеется файловое хранилище в котором находятся некоторые файлы (например, различные документы, изображения и т.д.). Доступ к хранилищу происходит через логин/пароль. Требуется реализовать следующий функционал:

* Взятие файла
* Загрузка файла в хранилище
* Редактирование файла с блокировкой от изменений другими пользователями

# Описание проблемы

В системе используется много различных сущностей: пользователь, хранилище, различные файловые редакторы. Каждая из описанных выше функций требует использования сразу нескольких из них. Таким образом, возникает несколько проблем:

1. Код неустойчив к изменениям подсистем.
2. Код тяжело масштабировать.
3. Общая запутанность кода.

# Аргументация выбора паттерна

Использование шаблона «Фасад» предоставляет общий интерфейс, не зависящий от внутреннего устройства подсистем. При необходимости интерфейс можно легко расширить, добавляя новые функции.

Существует очень похожий на фасад паттерн, решающий подобную задачу – паттерн Front Controller. Данный шаблон предоставляет общую точку входа для всех запросов, приходящих системе и обрабатывает их с помощью функций подсистем. Основная разница между этими шаблонами состоит в том, что фасад лишь является оберткой над подсистемами и не содержит дополнительной логики, в то время как Front Controller не имеет таких ограничений.

Выбор одного из двух шаблонов сводится к тому, можно ли реализовать необходимые функции только с помощью использования функций подсистем. В данном примере можно обойтись без применения дополнительной логики, поэтому выберем фасад.