

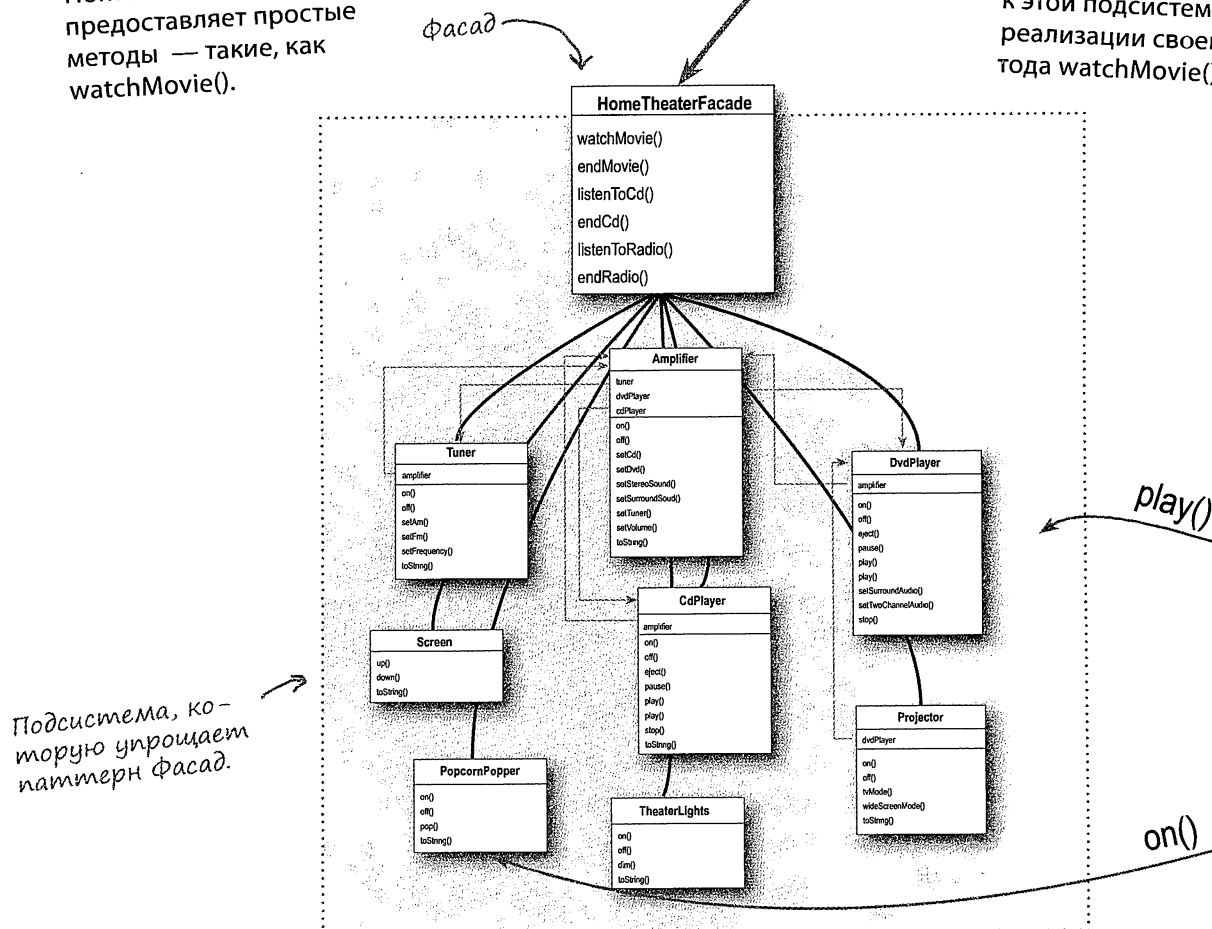
Свет, камера, фасад!

Фасад — именно то, что нам нужно: мы берем сложную подсистему и для упрощения работы с ней реализуем фасадный класс, который предоставляет общий, более удобный интерфейс. Не беспокойтесь; вся мощь сложной подсистемы остается в вашем распоряжении, но если вам нужен только упрощенный интерфейс — пользуйтесь Фасадом.

Давайте посмотрим, как работает паттерн Фасад:

- 1 Мы создадим фасадный интерфейс для домашнего кинотеатра. Класс `HomeTheaterFacade` предоставляет простые методы — такие, как `watchMovie()`.

- 2 Класс фасада рассматривает компоненты домашнего кинотеатра как подсистему и обращается с вызовами к этой подсистеме для реализации своего метода `watchMovie()`.





- 3 Клиентский код теперь обращается с вызовами к фасадному классу домашнего кинотеатра, а не к подсистеме. Таким образом, чтобы посмотреть фильм, мы вызываем всего один метод — `watchMovie()`, — а тот берет на себя все взаимодействие с освещением, DVD-проигрывателем, проектором, усилителем, экраном и т. д.



- 4 В паттерне Фасад подсистема остается открытой для непосредственного использования. Если вам понадобится расширенная функциональность классов подсистемы — обращайтесь к ним напрямую.