

Эссе Паттерны GoF.

212 Завьялова Милана

Паттерн проектирования — это часто встречаемое решение определенной проблемы при проектировании архитектуры программ. В отличие от готовых функций или библиотек, паттерн нельзя просто взять и скопировать в программу. Паттерн представляет собой не какой-то конкретный код, а общую концепцию или пример решения той или иной проблемы, которое нужно будет подстроить под нужды вашей программы. Описания паттернов обычно формальны и чаще всего состоят из таких пунктов: проблемы, которую решает паттерн; мотивации к решению проблемы способом, который предлагает паттерн; структуры классов, составляющих решение; примера на одном из языков программирования; особенностей реализации в различных контекстах; связей с другими паттернами.

“Facade”. Шаблон фасад — структурный шаблон проектирования, позволяющий скрыть сложность системы путём сведения всех возможных внешних вызовов к одному объекту, делегирующему их соответствующим объектам системы.

Как обеспечить унифицированный интерфейс с набором разрозненных реализаций или интерфейсов, например, с подсистемой, если нежелательно сильное связывание с этой подсистемой или реализация подсистемы может измениться?

Определить одну точку взаимодействия с подсистемой — фасадный объект, обеспечивающий общий интерфейс с подсистемой, и возложить на него обязанность по взаимодействию с её компонентами. Фасад — это внешний объект, обеспечивающий единственную точку входа для служб подсистемы. Реализация других компонентов подсистемы закрыта и не видна внешним компонентам. Фасадный объект обеспечивает реализацию GRASP паттерна Устойчивый к изменениям (Protected Variations) с точки зрения защиты от изменений в реализации подсистемы.

Когда использовать фасад?

- Когда имеется сложная система, и необходимо упростить с ней работу. Фасад позволит определить одну точку взаимодействия между клиентом и системой.
- Когда надо уменьшить количество зависимостей между клиентом и сложной системой. Фасадные объекты позволяют отделить, изолировать компоненты системы от клиента и развивать и работать с ними независимо.
- Когда нужно определить подсистемы компонентов в сложной системе. Создание фасадов для компонентов каждой отдельной подсистемы позволит упростить взаимодействие между ними и повысить их независимость друг от друга.