

# Урок 11



## Паттерн. Singleton.

Singleton (Синглтон, одиночка) относится к классу порождающих паттернов. Он используется для создания всего одного экземпляра класса, и гарантирует, что во время работы программы не появиться второй. Применяется в случаях когда критично иметь всегда только один экземпляр объекта и чтобы его свойства могли быть сохранены и доступны по всей программы с одними и теми же значениями.



# Паттерн. Фабрика

Назначение фабрики в том, чтобы создавать объекты. Этот шаблон обычно реализуется в виде классов или в виде статических методов классов и преследует следующие цели:

- Выполнение повторяющихся операций, необходимых при создании похожих объектов
- Предложить пользователям фабрики способ создания объектов без необходимости знать их тип (класс) на этапе компиляции



## Паттерн. Модуль.

Применяется для разделения функционала на логические части в которых есть общедоступные методы и свойства, а также скрытые для доступа из вне свойства методы для не нарушения функционала. Обычно модуль использует только локальные переменные и не использует глобальные, делая таким образом модуль независимой частью и делает его возможным для переноса на другие проекты.



# Паттерн. Итератор.

Шаблон итератора применяется, когда имеется объект, содержащий совокупность данных. Эти данные могут храниться в виде сложной структуры, а вам необходимо обеспечить удобный доступ к каждому элементу этой структуры. Пользователи вашего объекта не обязаны знать, как организованы ваши данные, — им необходим доступ к отдельным элементам.

Объект, реализующий шаблон итератора, должен предоставить метод `next()`. При последующем обращении метод `next()` должен вернуть следующий элемент, и только вам решать, что означает понятие «следующий» для вашей конкретной структуры данных.



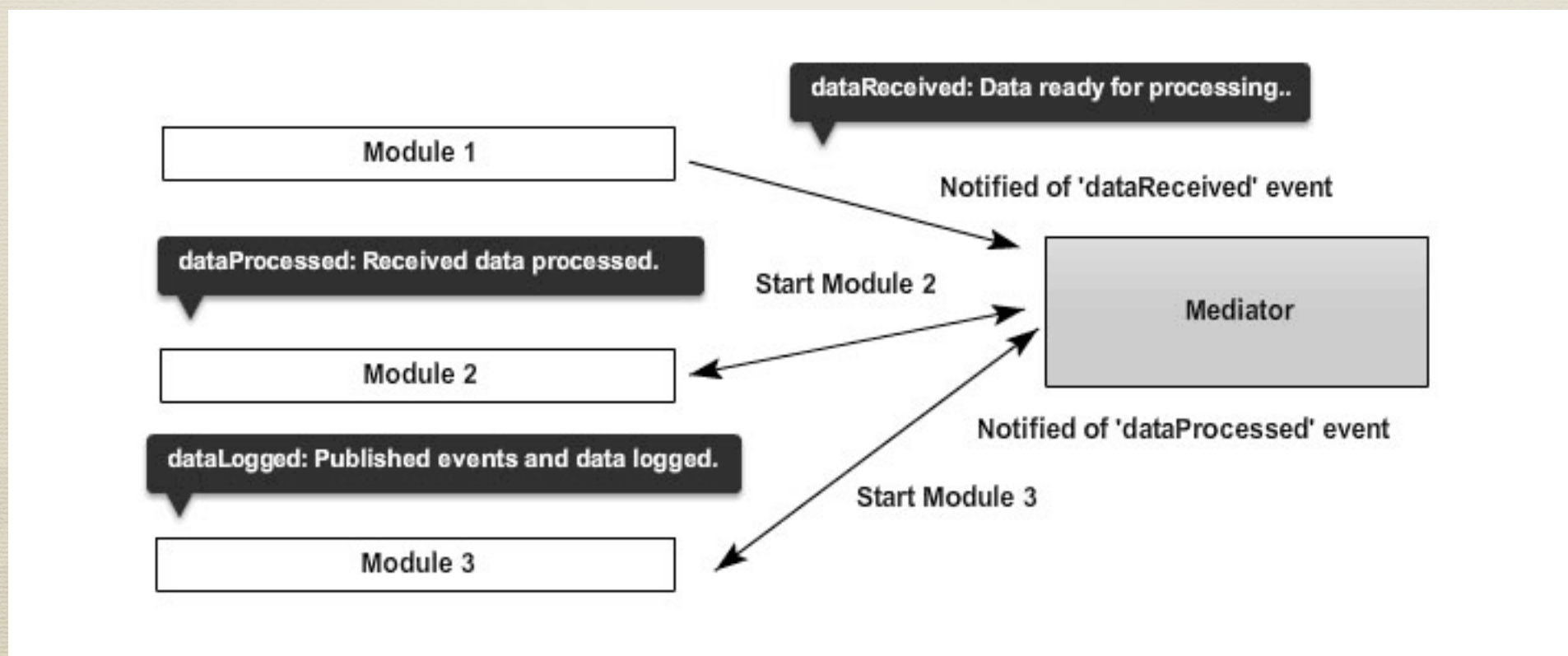
# Паттерн. Декоратор

Это структурный шаблон проектирования, предназначенный для динамического подключения дополнительного поведения к объекту. Шаблон Декоратор предоставляет гибкую альтернативу практике создания подклассов с целью расширения функциональности. Функционал может добавляться и удаляться в зависимости от условий.



# Паттерн. Медиатор

Медиатор выступает в качестве посредника в общении между различными модулями, инкапсулируя их взаимодействие. Кроме того, этот шаблон проектирования, предотвращая прямое взаимодействие различных компонентов системы, способствует ослаблению связей в коде. Он так же помогает в решении проблем, связанных с зависимостями модулей.





# Паттерн. Наблюдатель

Все события, возникающие в броузерах (mouseover, keypress и другие), являются примерами реализации этого шаблона. Этот шаблон иногда называют механизмом собственных событий, то есть событий, которые возбуждаются программно, в отличие от тех, что возбуждаются броузером. Еще одно название этого шаблона – подписчик/издатель.

Основная идея шаблона состоит в ослаблении связи между объектами. Вместо вызова одним объектом метода другого объекта некоторый объект подписывается на получение извещений об определенных событиях от другого объекта.

Подписчик также называется наблюдателем, а объект, за которым ведется наблюдение, называется издателем, или объектом наблюдения. Издатель оповещает (вызывает) всех подписчиков о наступлении какого-то важного события, и часто сообщение передается им в форме объекта события.