

Patrones de Diseño

Los patrones de software son soluciones de diseño reusables a problemas recurrentes.

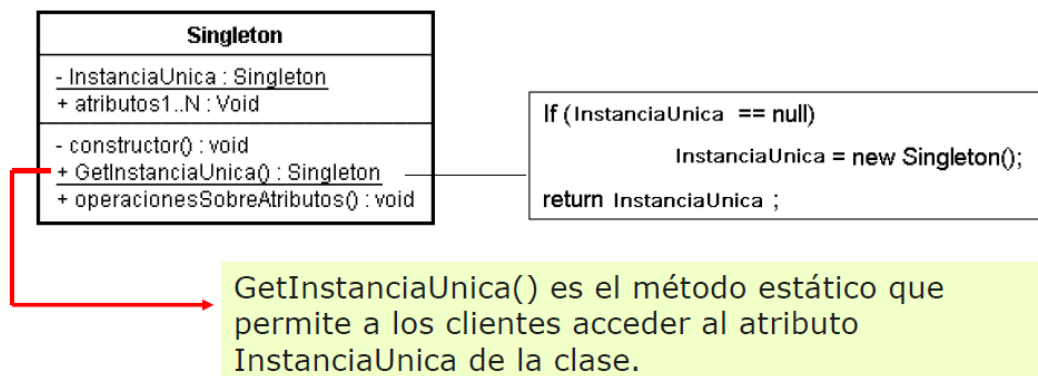
Clasificación de Patrones

- Creacionales
Resuelven problemas relativos a la creación de objetos.
- Estructurales
Resuelven problemas relativos a la composición de clases y objetos.
- Comportamiento
Resuelven problemas relativos a la interacción entre los objetos y la distribución de responsabilidades.

Creacionales	Estructurales	Comportamiento
Singleton Abstract Factory Factory Method	Composite Proxy Facade	Command Iterator Mediator Template Method Visitor Observer

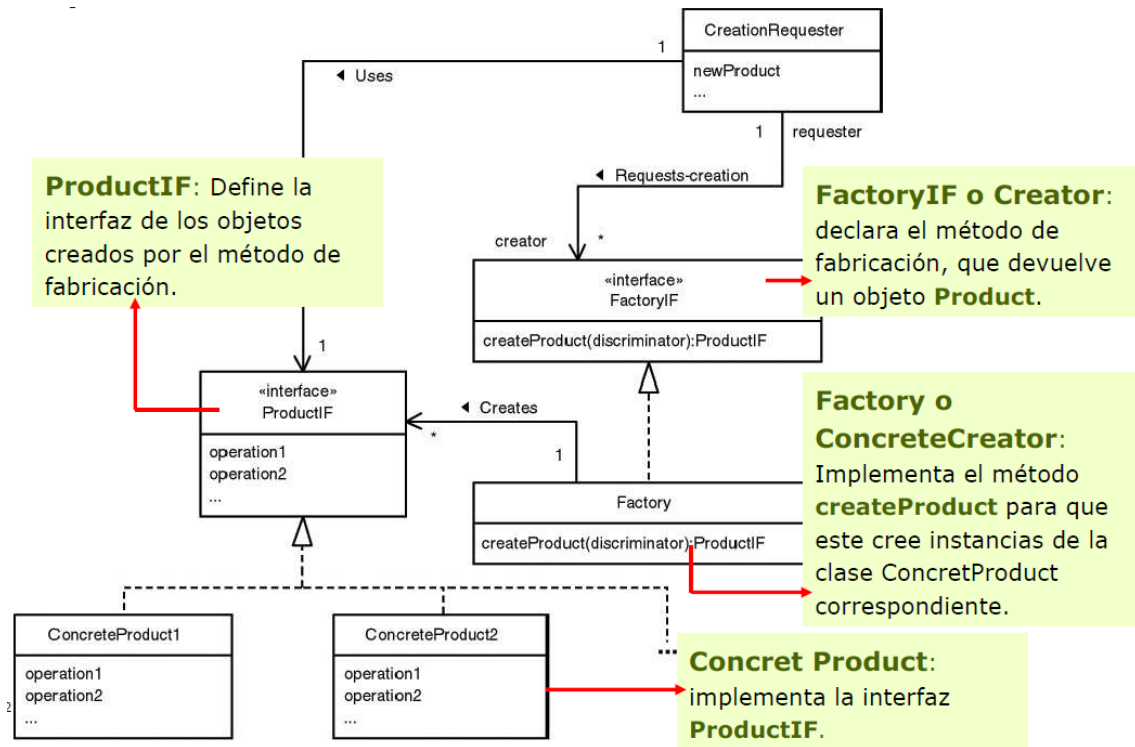
Patron Singleton

Asegura que solo existe una instancia de cada clase, y que todos los objetos que la usan accederan a la misma instancia.



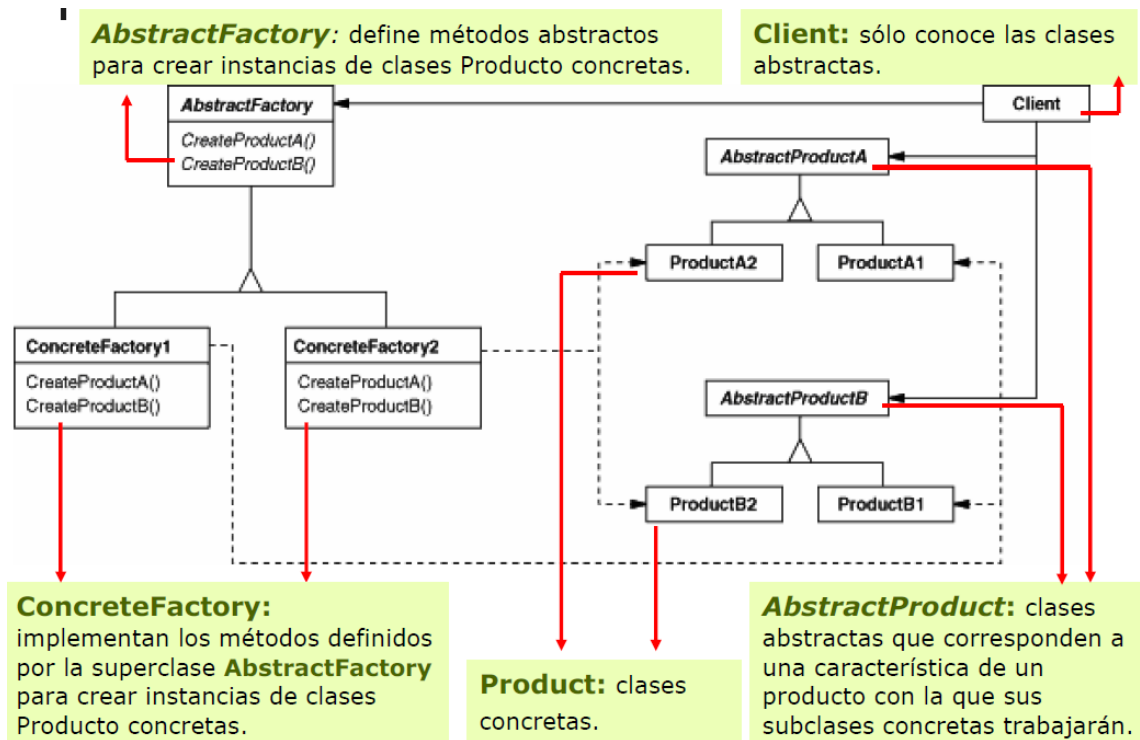
Patron Factory Method

Define una interfaz para la creación de objetos, pero delega en las subclases la decisión de qué o cual clase instanciar.



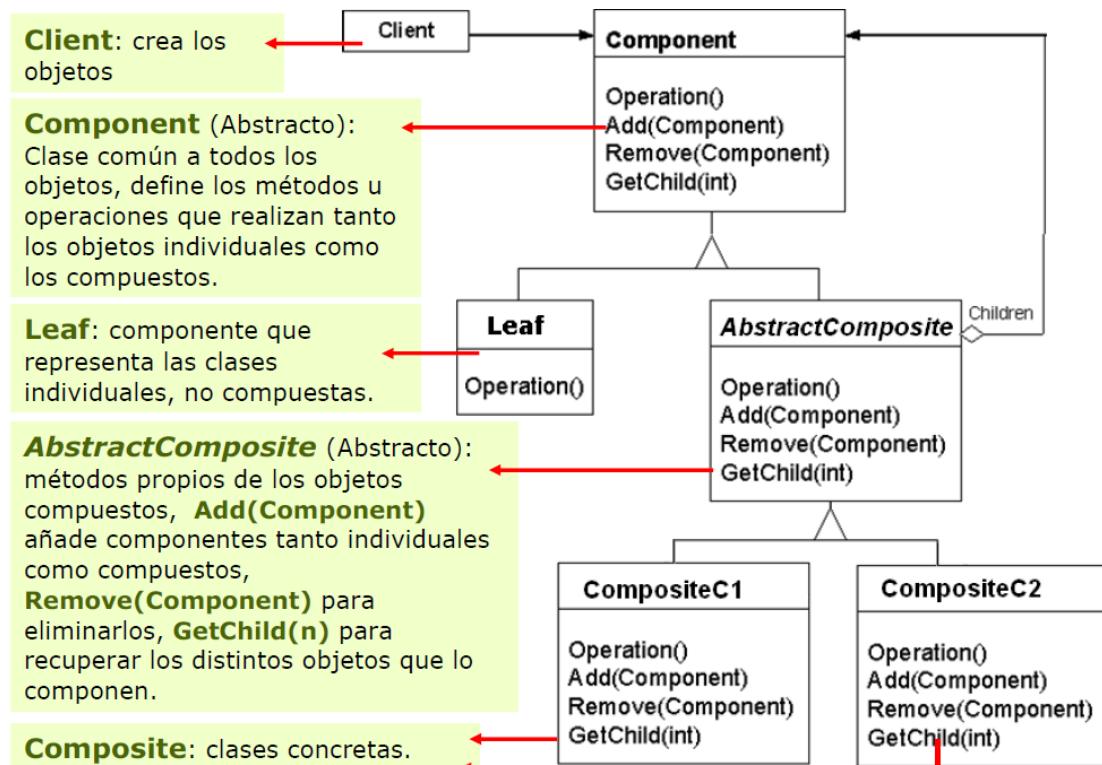
Patron Abstract Factory

Provee una interface que permite crear familias de objetos relacionados o dependientes, sin especificar sus clases concretas.



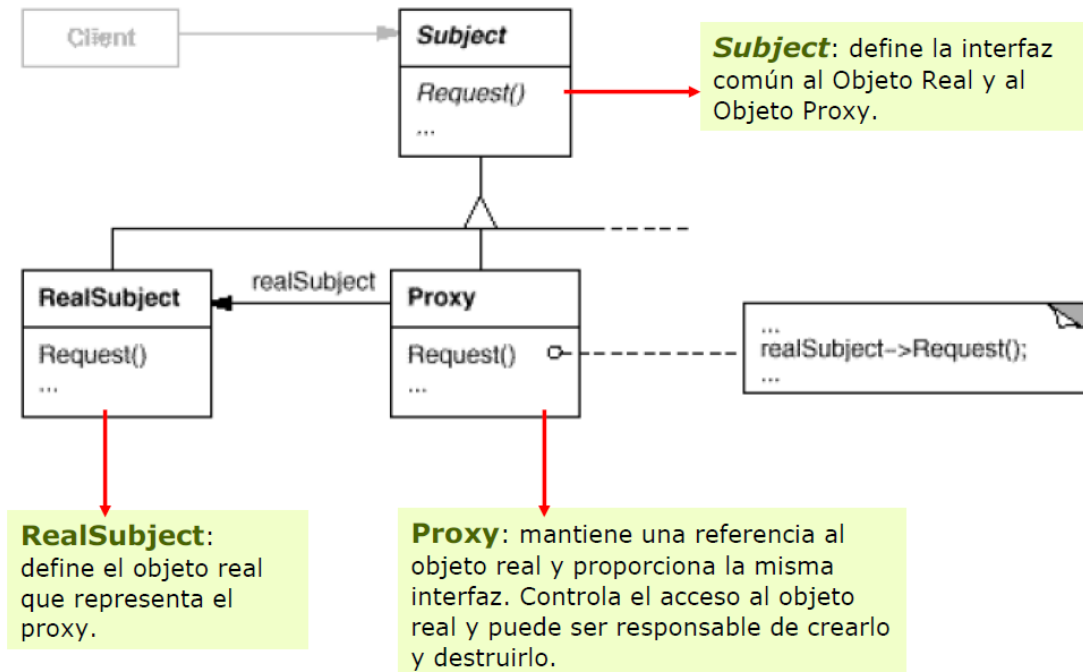
Patron Composite

Permite que el cliente maneje indistintamente objetos individuales o composiciones de objetos y los trate como una estructura compuesta uniforme.



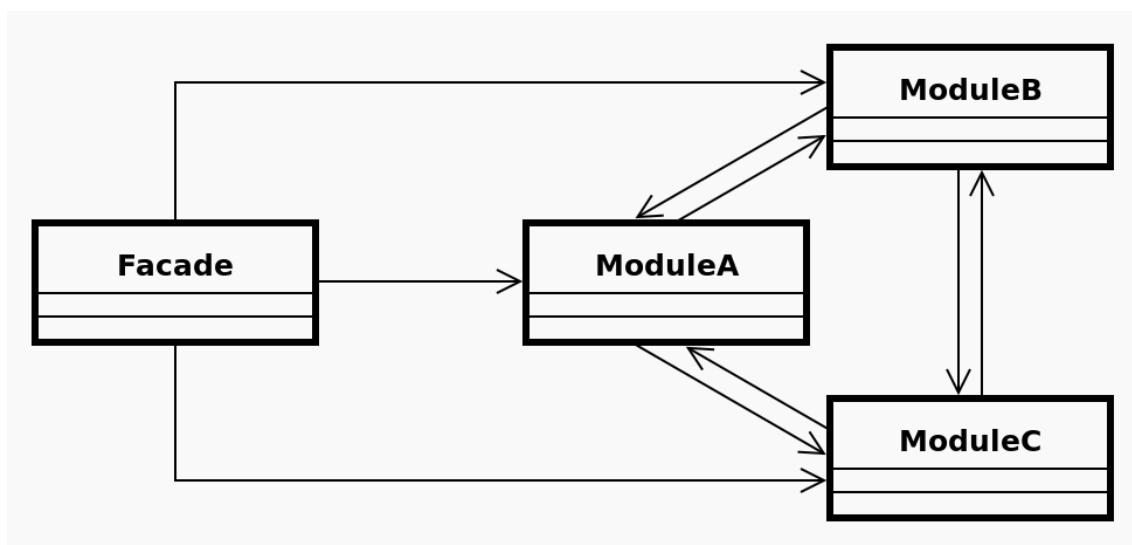
Patron Proxy

Un objeto sustituye a otro para controlar el acceso al mismo, generalmente por motivos de eficiencia.



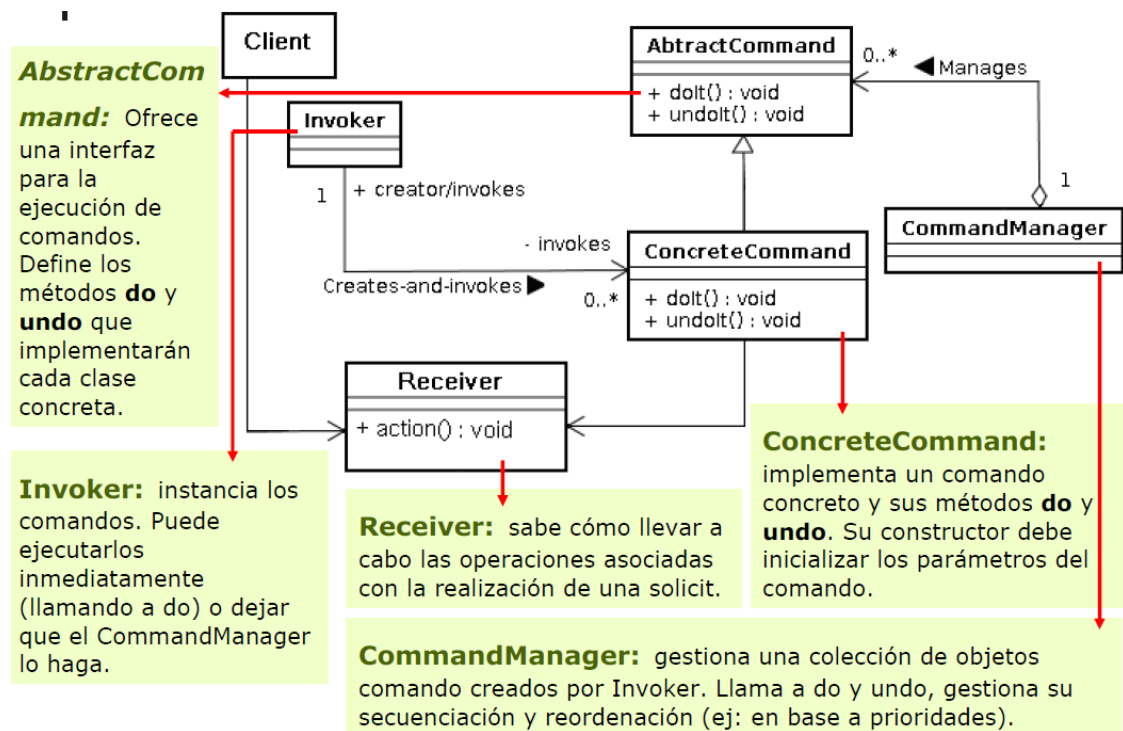
Patron Facade

El patrón fachada viene motivado por la necesidad de estructurar un entorno de programación y reducir su complejidad con la división en subsistemas, minimizando las comunicaciones y dependencias entre éstos.



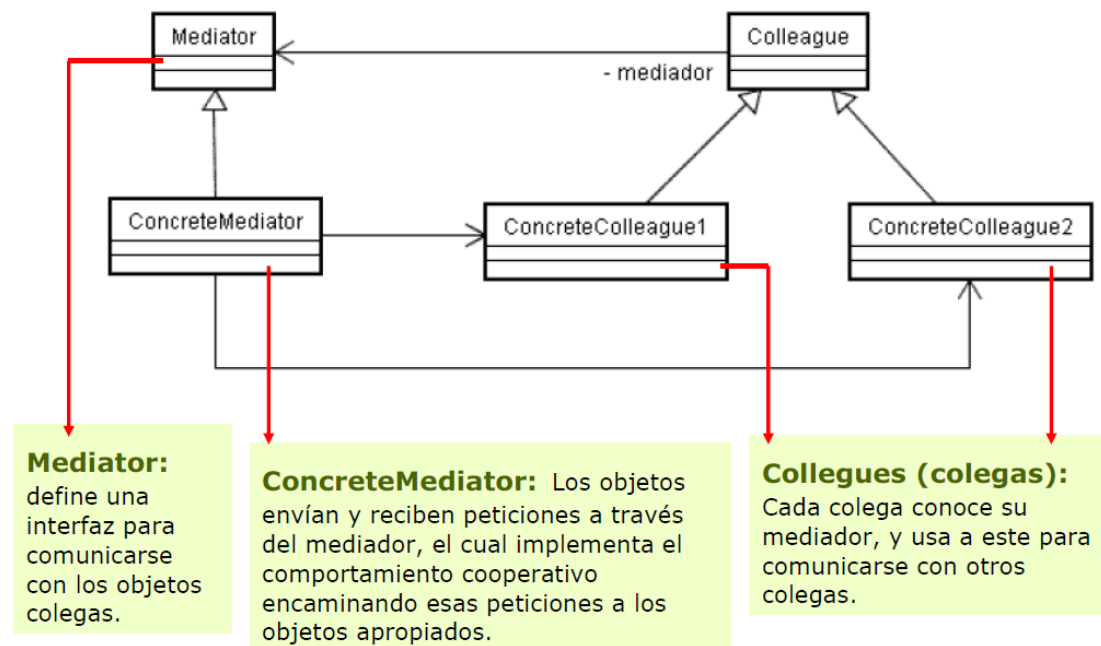
Patron Command

Su propósito es encapsular una solicitud como un objeto.



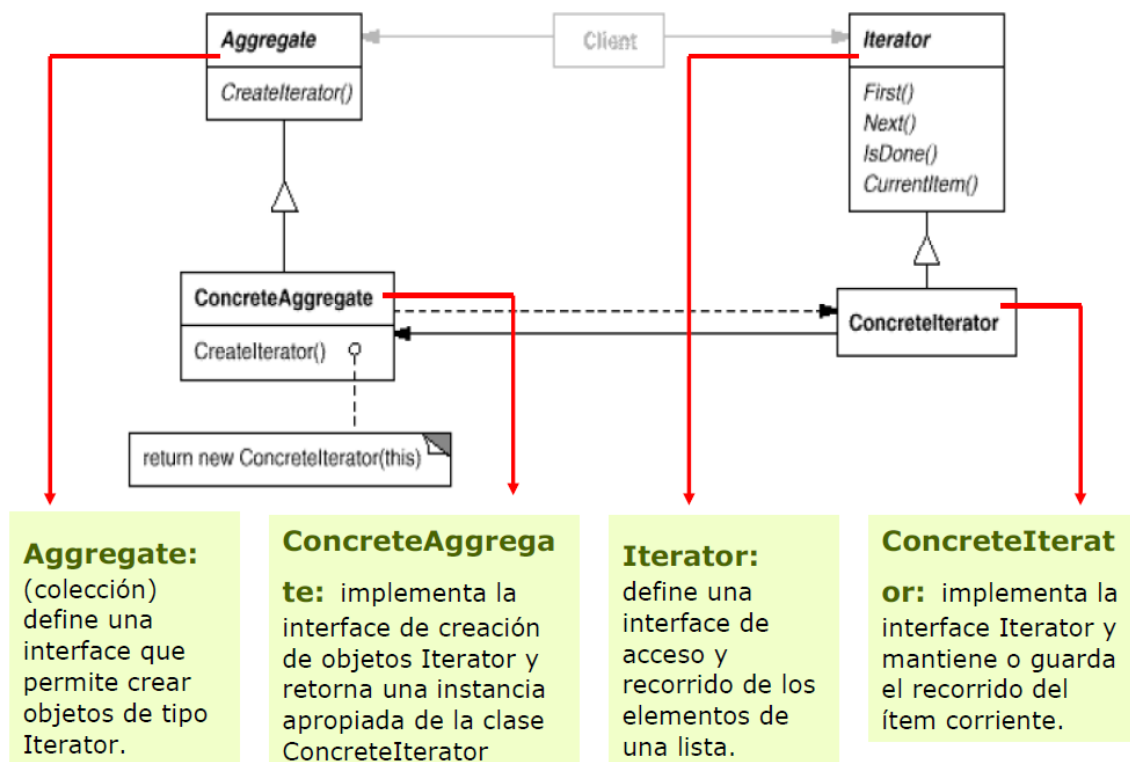
Patron Mediator

Define un objeto que encapsula cómo interactúan un conjunto de objetos. Promueve un bajo acoplamiento al evitar que los objetos se refieran unos a otros explícitamente, y permite variar la interacción entre ellos de forma independiente.



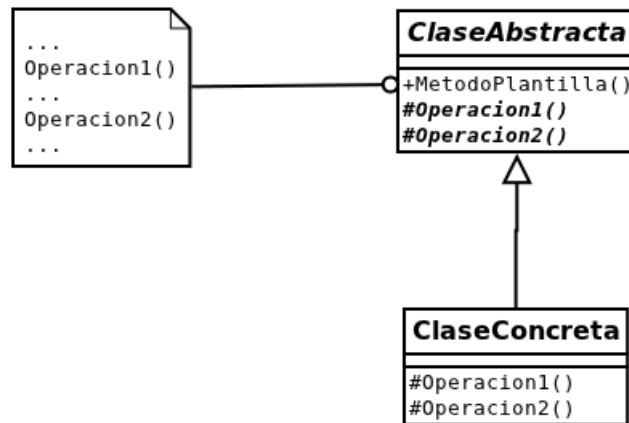
Patron Iterator

Provee un camino para acceder a los elementos de una colección secuencial de objetos sin exponer la representación interna de dicho recorrido.



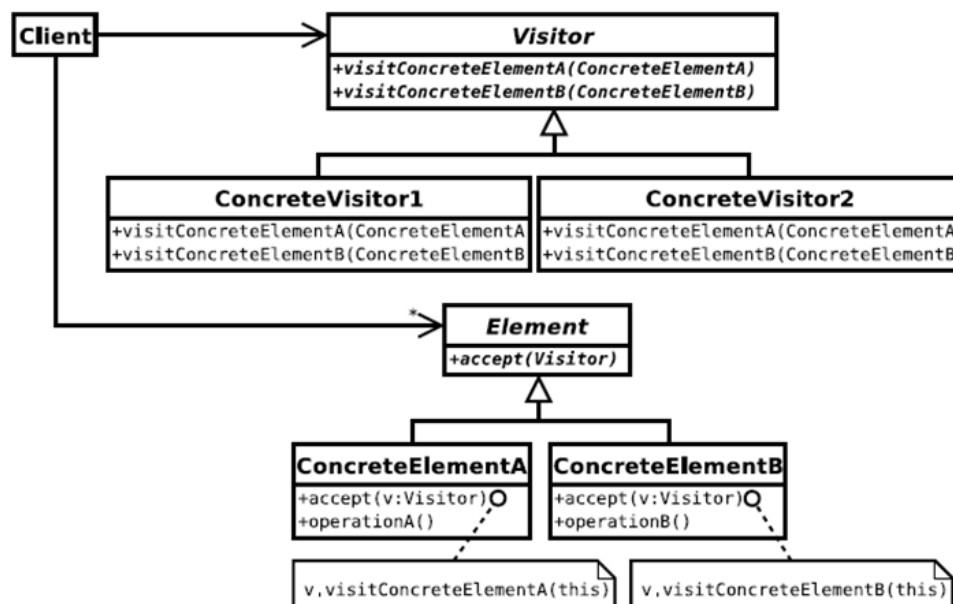
Patron Template Method

Permitir que ciertos pasos de un algoritmo definido en una operación de una superclase, sean redefinidos en las subclases sin necesidad de tener que sobrescribir la operación entera.



Patron Visitor

Es un patrón de comportamiento, que permite definir una operación sobre objetos de una jerarquía de clases sin modificar las clases sobre las que opera.



Patron Observer

Define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes.

