Clase 11: Patrones de Diseño

Rodrigo Arturo Saffie Kattan

Pontificia Universidad Católica de Chile

rasaffie@ing.puc.cl

6 de septiembre de 2016

Contenidos

Repaso Clase Anterior

Patrones de diseño revisados:

- Chain of Responsibility
- Command
- Interpreter

De comportamiento

- Se centran en las interacciones y responsabilidades entre objetos

De Comportamiento

Iterator

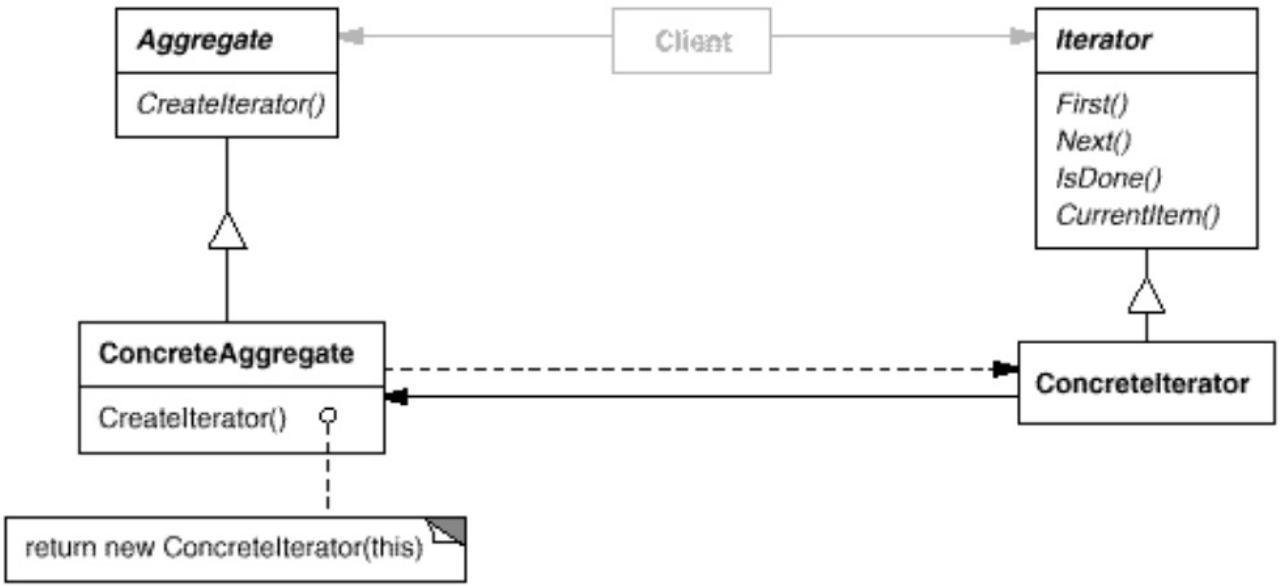
Provee una forma de acceder a los elementos de una agregación de objetos secuenciales sin exponer su representación subyacente.

¿Cuándo se utiliza?

- Para acceder al contenido de un objeto agregado sin exponer su representación interna.
- Para soportar múltiples recorridos de objetos agregados.
- Para proveer una interfaz uniforme para recorrer diferentes estructuras agregadas.

Iterator

Provee una forma de acceder a los elementos de una agregación de objetos secuenciales sin exponer su representación subyacente.



Ejemplo : Iterator

Mediator

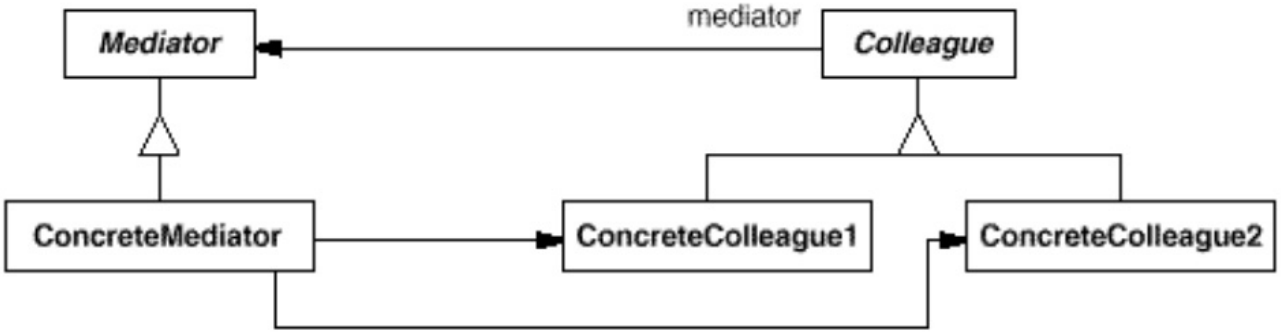
Define un objeto que encapsula como un conjunto de objetos interactúan. Este patrón promueve el bajo acoplamiento al evitar que los objetos se referencien entre ellos explicitamente, y permite variar sus interacciones independientemente.

¿Cuándo se utiliza?

- Un conjunto de objetos se comunican de una forma compleja, pero bien definida.
- Un comportamiento distribuido en múltiples clases debería ser configurable sin generar muchas sub-clases.

Mediator

Define un objeto que encapsula como un conjunto de objetos interactúan. Este patrón promueve el bajo acoplamiento al evitar que los objetos se referencien entre ellos explicitamente, y permite variar sus interacciones independientemente.



Ejemplo : Mediator

Memento

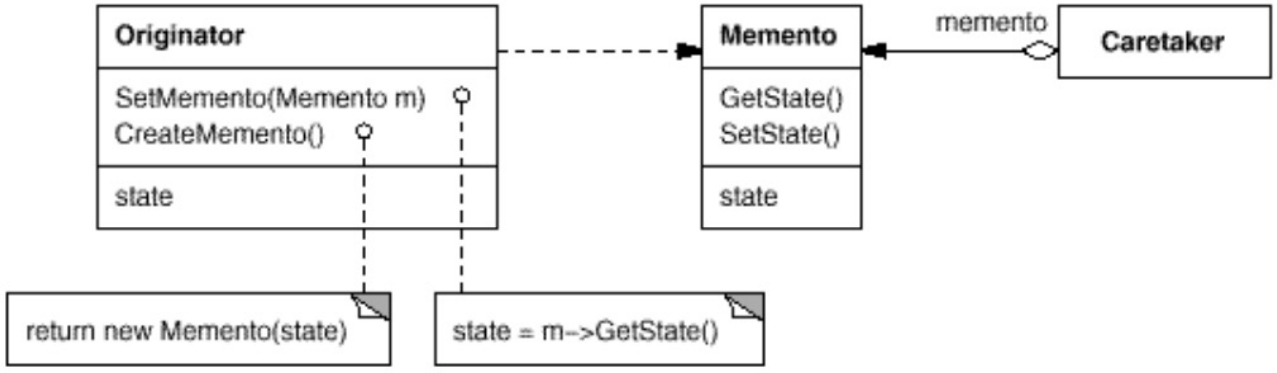
Sin violar el principio de ocultamiento, captura y expone el estado interno de un objeto para que este pueda volver a ese estado en otro momento.

¿Cuándo se utiliza?

- Una "fotografía" de una parte del estado de un objeto debe ser grabada para poder restaurarla.
- Una interfaz directa para obtener el estado de un objeto expondría detalles de implementación, violando el principio de ocultamiento.

Memento

Sin violar el principio de ocultamiento, captura y expone el estado interno de un objeto para que este pueda volver a este estado en otro momento.



Ejemplo : Memento

De Comportamiento

Observer

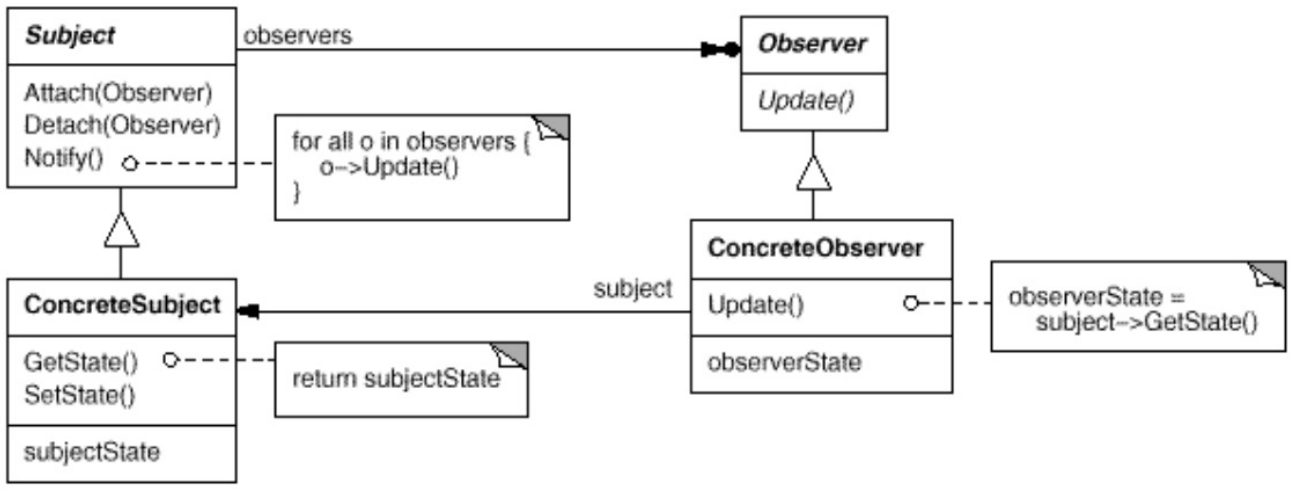
Define una relación de dependencia entre uno y N objetos, para que cuando ese objeto cambie su estado, todos los N sean notificados y se actualicen automáticamente.

¿Cuándo se utiliza?

- Cuando una abstracción tiene dos aspectos, uno dependiente del otro. Encapsular estos aspectos en objetos separados permite variarlos y reutilizarlos independientemente.
- Cuando un cambio en un objeto genera cambios en otros, y no se sabe cuántos objetos deben cambiar.
- Cuando un objeto debería ser capaz de notificar a otros objetos sin realizar supuestos de quiénes son estos objetos. En otras palabras, se quiere reducir el acoplamiento entre objetos dependientes.

Observer

Define una relación de dependencia entre uno y N objetos, para que cuando ese objeto cambie su estado, todos los N sean notificados y se actualicen automáticamente.



Ejemplo : Observer

Referencias

Pressman, R. S. (2009)

Software Engineering: A Practitioner's Approach

7th ed., *McGraw-Hill Education*

Gamma, E., Helm. R., Johnson R., Vlissides, J. (1994)

Design Patterns: Elements of Reusable Object-Oriented Software

1st ed., *Addison-Wesley Professional*

dofactory.com

http://www.dofactory.com/net/design-patterns

Fin