

Clase 8: Patrones de Diseño

Rodrigo Arturo Saffie Kattan

Pontificia Universidad Católica de Chile

rasaffie@ing.puc.cl

25 de agosto de 2016

Contenidos

1 Repaso Clase Anterior

2 Patrones de Diseño

- Creacionales
- Estructurales

Patrones de diseño revisados:

- Abstract Factory
- Builder
- Factory Method
- Prototype

Repaso Clase Anterior



Singleton

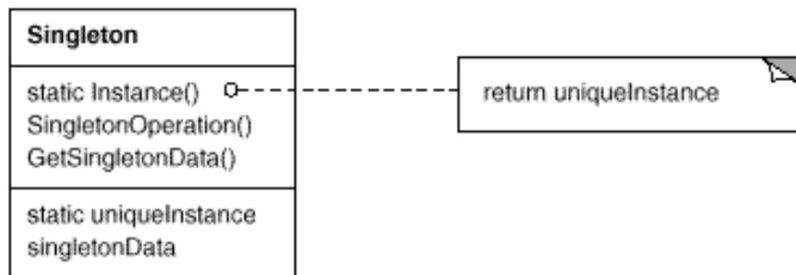
Garantiza que una clase tenga solamente una instancia, y provee un acceso global a la instancia.

¿Cuándo se utiliza?

- Debe haber exactamente una sola instancia de una clase, y puede ser accedida por distintos clientes

Singleton

Garantiza que una clase tenga solamente una instancia, y provee un acceso global a la instancia.



Ejemplo : [Singleton](#)

Estructurales

- Se centran en cómo los objetos se organizan e integran en un sistema

Adapter

Convierte la interfaz de una clase en otra interfaz que los clientes esperan. *Adapter* permite a ciertas clases interactuar, sin lo cual serían incompatibles.

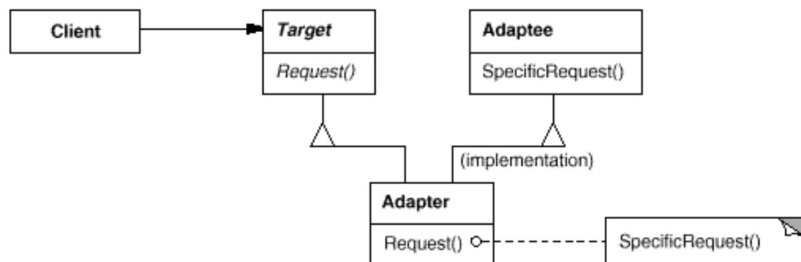
¿Cuándo se utiliza?

- Se necesita utilizar una clase existente, pero su interfaz es incompatible con lo que se necesita
- Se quiere crear una clase que coopere con otras clases que utilizan interfaces incompatibles

Patrones Estructurales

Adapter

Convierte la interfaz de una clase en otra interfaz que los clientes esperan. *Adapter* permite a ciertas clases interactuar, sin lo cual serían incompatibles.



Ejemplo : [Adapter](#)

Bridge

Desacopla una abstracción de su implementación, para que ambas puedan variar independientemente

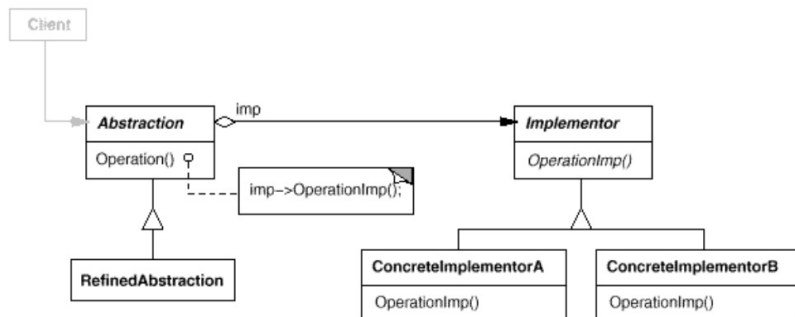
¿Cuándo se utiliza?

- Se quiere evitar una unión permanente entre una abstracción y su implementación
- Una abstracción y su implementación deberían ser extensibles
- Cambios en la implementación de una abstracción no deberían tener impacto en los clientes

Patrones Estructurales

Bridge

Desacopla una abstracción de su implementación, para que ambas puedan variar independientemente



Ejemplo : Bridge

Composite

Compone objetos en estructuras de árboles para representar jerarquías completas. *Composite* permite a los clientes tratar objetos y composiciones de objetos de igual manera.

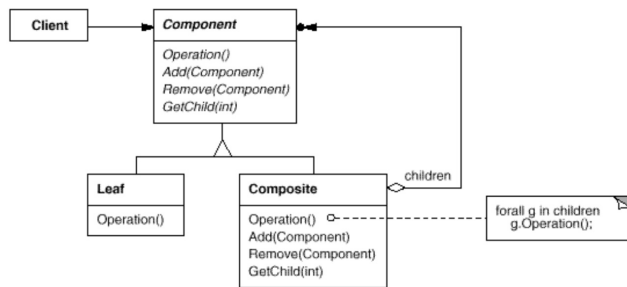
¿Cuándo se utiliza?

- Se quiere representar jerarquías completas de objetos.
- Se quiere que los clientes puedan ignorar las diferencias entre objetos compuestos y objetos individuales. Los clientes tratarán a todos los objetos de la jerarquía de manera uniforme.

Patrones Estructurales

Composite

Compone objetos en estructuras de árboles para representar jerarquías completas. *Composite* permite a los clientes tratar objetos y composiciones de objetos de igual manera.



Ejemplo : Composite

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