

Clase 10: Patrones de Diseño

Rodrigo Arturo Saffie Kattan

Pontificia Universidad Católica de Chile

rasaffie@ing.puc.cl

1 de septiembre de 2016

Contenidos

- 1 Repaso Clase Anterior
- 2 Patrones de Diseño
 - De Comportamiento

Patrones de diseño revisados:

- Decorator
- Façade
- Flyweight
- Proxy

De comportamiento

- Se centran en las interacciones y responsabilidades entre objetos

Chain of Responsibility

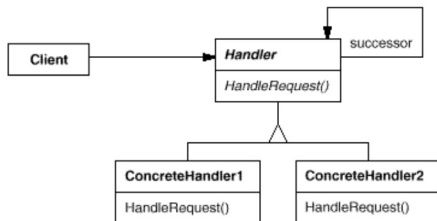
Evita el acoplamiento entre el objeto que envía una solicitud y el objeto que la recibe, al dar la oportunidad de que más de un objeto procese la solicitud. Encadena los objetos solicitados y delega la solicitud a lo largo de la cadena hasta que un objeto se encargue de ella.

¿Cuándo se utiliza?

- Más de un objeto puede encargarse de una solicitud, y el que se hará cargo es desconocido.
- Se quiere generar una solicitud a uno de varios objetos sin especificar el receptor.
- El conjunto de objetos que puede hacerse cargo de la solicitud se define dinámicamente.

Chain of Responsibility

Evita el acoplamiento entre el objeto que envía una solicitud y el objeto que la recibe, al dar la oportunidad de que más de un objeto procese la solicitud. Encadena los objetos solicitados y delega la solicitud a lo largo de la cadena hasta que un objeto se encargue de ella.



Ejemplo : Chain of Responsibility

Command

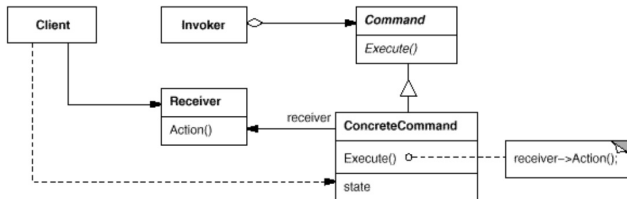
Encapsula una solicitud como un objeto, para así permitir la parametrización de clientes con solicitudes distintas, encolar o registrar solicitudes, y soportar operaciones que se pueden deshacer.

¿Cuándo se utiliza?

- Se necesita parametrizar objetos con una acción a realizar. *Command* es un reemplazo para *callbacks* en OOP.
- Especificar, encolar y ejecutar solicitudes en momentos diferentes.
- Soportar que la solicitud se deshaga.
- Estructurar un sistema alrededor de operaciones complejas, formadas a partir de operaciones primitivas.

Command

Encapsula una solicitud como un objeto, para así permitir la parametrización de clientes con solicitudes distintas, encolar o registrar solicitudes, y soportar operaciones que se pueden deshacer.



Ejemplo : **Command**

Interpreter

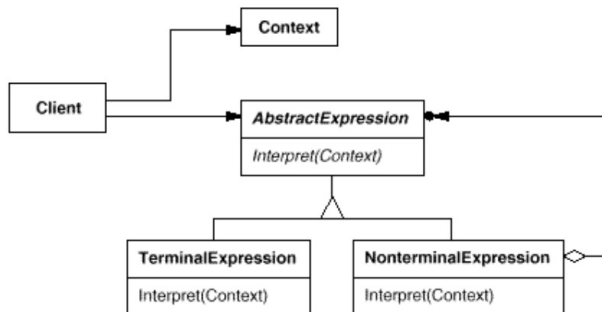
Dado un lenguaje, define una representación de su gramática junto con un intérprete que utiliza la representación para interpretar frases en el idioma.

¿Cuándo se utiliza?

- La gramática del lenguaje es simple
- La eficiencia no es un requisito principal

Interpreter

Dado un lenguaje, define una representación de su gramática junto con un intérprete que utiliza la representación para interpretar frases en el idioma.



Ejemplo : [Interpreter](#)

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