



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 14

Patrones de diseño

IIC2143 - Ingeniería de Software
Sección 1

Rodrigo Saffie

rasaffie@uc.cl

25 de abril de 2018

Patrones de diseño

¿Qué es un patrón de diseño?

“Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.” [Christopher Alexander](#)

Este concepto fue acuñado por GoF en [Design Patterns: Elements of Reusable Object-Oriented Software](#) en 1994

Patrones de diseño

¿Qué es un patrón de diseño?

“Design patterns make it easier to reuse successful designs and architectures. Expressing proven techniques as design patterns makes them more accessible to developers of new systems. Design patterns help you choose design alternatives that make a system reusable and avoid alternatives that compromise reusability.” [GoF, 1994]

Patrones de diseño

¿Qué caracteriza un patrón de diseño?

Según [GoF, 1994]:

- Un nombre
- Un problema
- Una solución
- Sus consecuencias

Patrones de diseño

En este curso se estudiarán algunos patrones de los 23 definidos en el libro de [GoF, 1994]

Estos patrones son aplicados a OOP, pero existen **muchos** más

Patrones de diseño

¿Por qué estudiarlos?

- Permiten reutilizar soluciones
- Permiten establecer terminología común: mejora la comunicación del equipo
- Nos dan una perspectiva de más alto nivel para el diseño de *software*
- Código es más fácil de modificar
- Permiten aprender estrategias generales de diseño

Patrones de diseño

Existen 3 clasificaciones:

- Creacionales
- Estructurales
- De comportamiento

Patrones de diseño

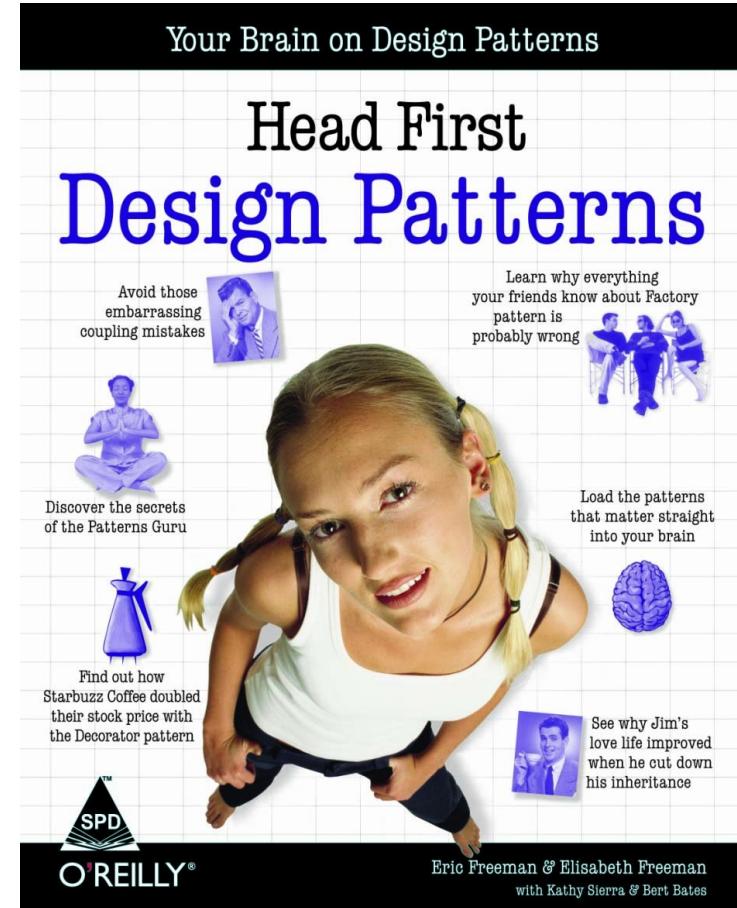
¿Cómo seleccionar un patrón de diseño?

- Detectar el problema: causa del rediseño
- Analizar el propósito de cada patrón
- Identificar si hay relación entre el problema y un propósito
- Estudiar los patrones de la misma categoría
- Considerar qué podría variar en el diseño

Patrones de diseño



Patrones de diseño



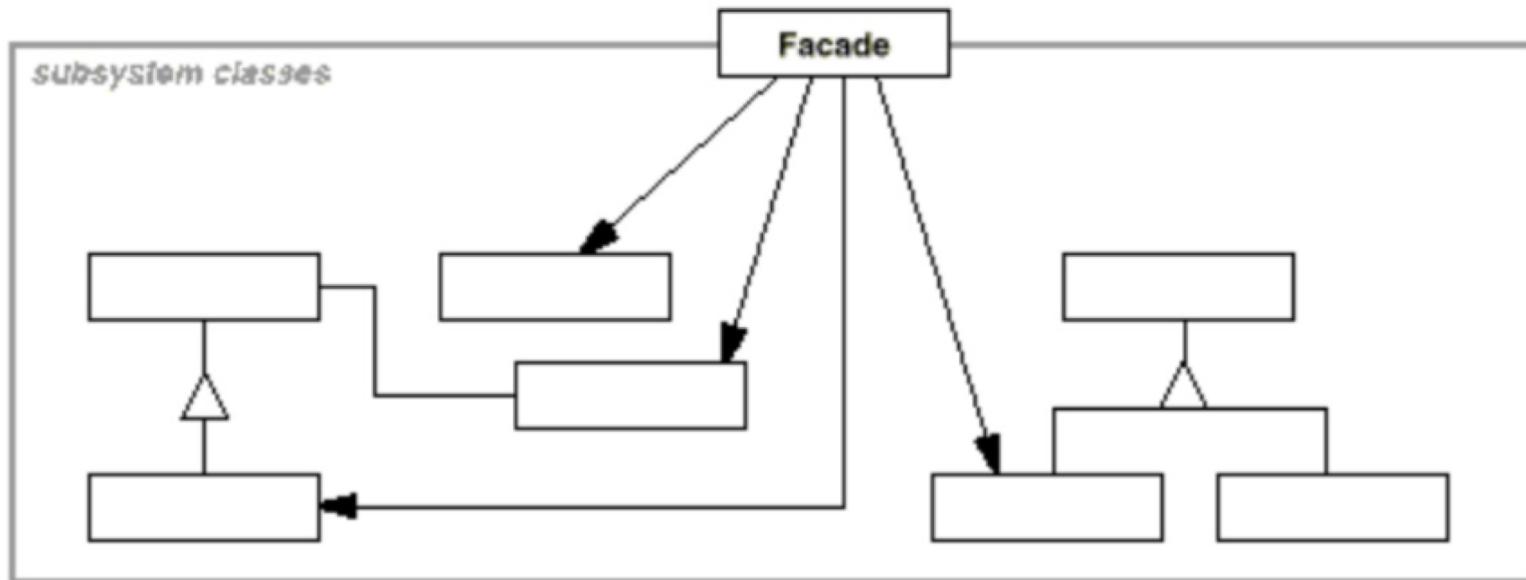
Patrones estructurales

Façade (Fachada)

Provee una interfaz unificada a un conjunto de interfaces en un sub-sistema. *Façade* define un interfaz de alto nivel que hace a un sistema más fácil de utilizar.

Patrones estructurales

Facade (Fachada)



Ejemplo

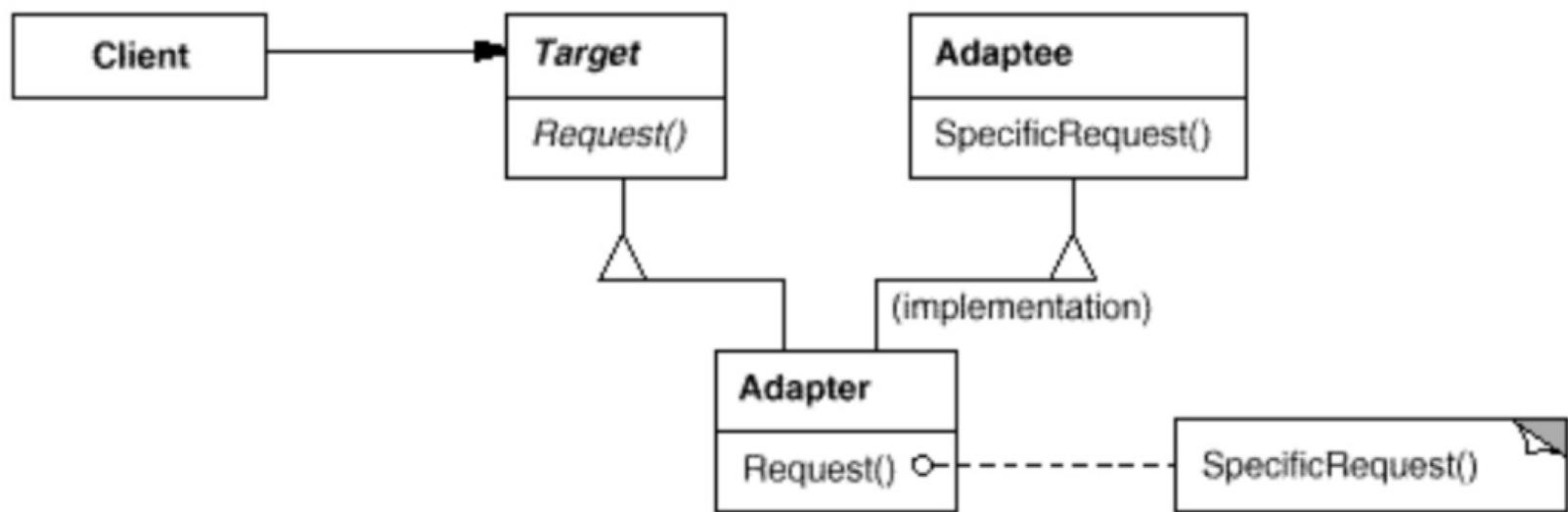
Patrones estructurales

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.

Patrones estructurales

Adapter



Ejemplo

Patrones estructurales

Decorator

Agrega responsabilidad adicional a un objeto de manera dinámica. Decoradores permiten una alternativa flexible para extender funcionalidad de sub-clases.

Patrones estructurales

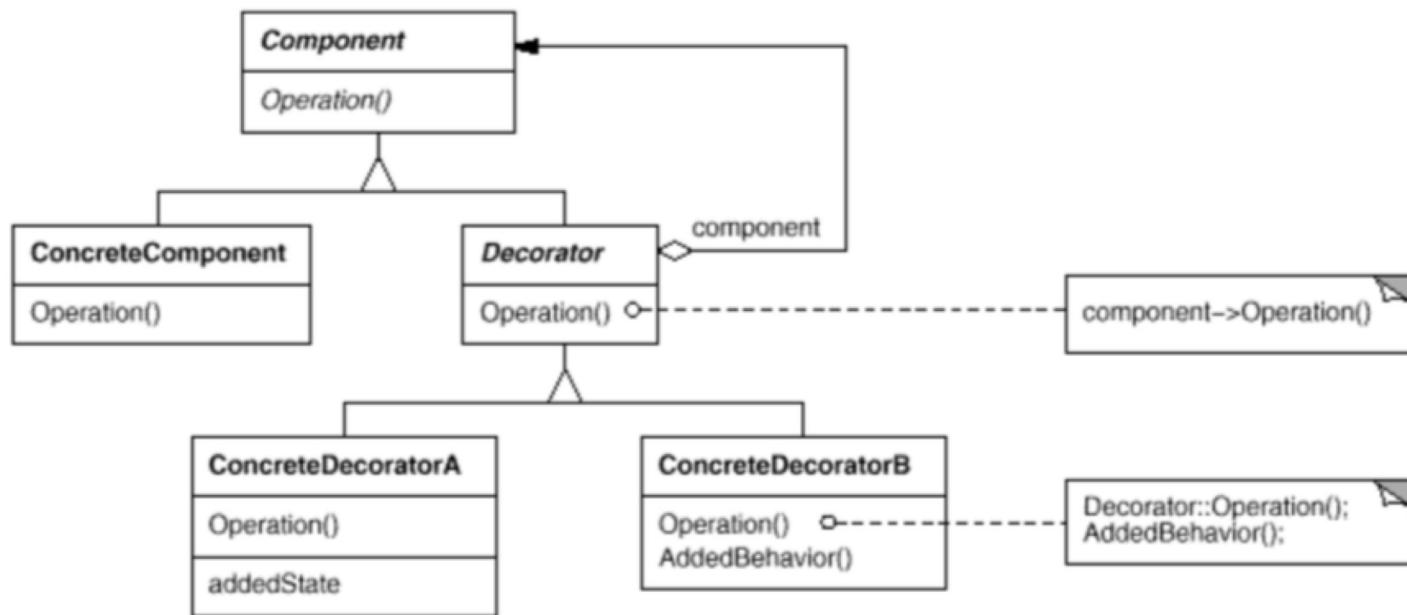
Decorator

¿Cuándo se utiliza?

- Para agregar responsabilidad a objetos individuales de manera dinámica y transparente (sin afectar otros objetos)
- Para responsabilidades que pueden ser retiradas
- Cuando extender con sub-clases es impráctico.
Puede ser por un gran número de extensiones independientes, o que no se pueden desprender sub-clases a partir de una clase

Patrones estructurales

Decorator



[Ejemplo1](#) y [Ejemplo2](#)

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.

Patrones estructurales

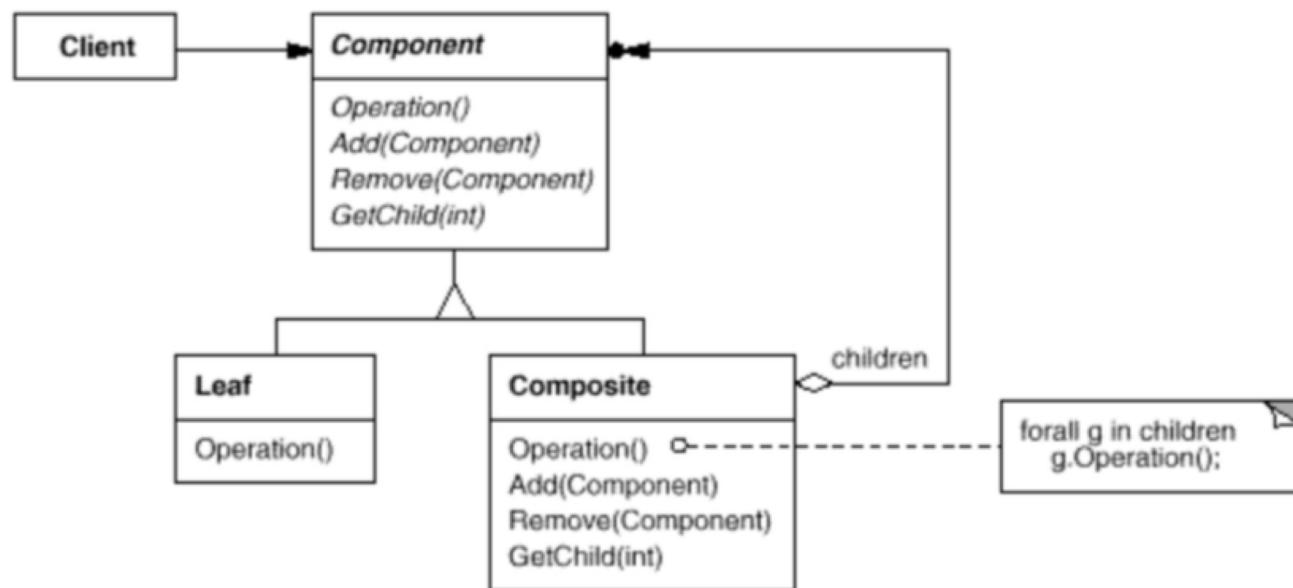
Composite

¿Cuándo se utiliza?

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

Patrones estructurales

Composite



Ejemplo

Patrones creacionales

Singleton

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

Patrones creacionales

Singleton

¿Cuándo se utiliza?

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

Patrones creacionales

Singleton



Ejemplo



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 14

Patrones de diseño

IIC2143 - Ingeniería de Software
Sección 1

Rodrigo Saffie

rasaffie@uc.cl

25 de abril de 2018