Clase 6: Patrones de Diseño

Rodrigo Arturo Saffie Kattan

Pontificia Universidad Católica de Chile

rasaffie@ing.puc.cl

16 de agosto de 2016

Contenidos

Repaso Clase Anterior

¿Qué es un componente?

Distintos puntos de vista:

- Vista orientada a objetos
- Vista tradicional
- Vista orientada a procesos

Repaso Clase Anterior

Principios S.O.L.I.D.

- S – Single-responsiblity principle
- O – Open-closed principle
- L – Liskov substitution principle
- I – Interface segregation principle
- D – Dependency Inversion principle

Repaso Clase Anterior

Principios S.O.L.I.D. generan código:

- Menos complejo
- Fácil de mantener
- Fácil de extender

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 formalizado por [GoF, 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]

"[...] allows the software engineering community to capture design knowledge in a way that enables it to be reused." [Pressman, 2009]

Patrones de Diseño

¿Qué caracteriza un patrón de diseño?

Según [Pressman, 2009]:

- Un contexto
- Un problema
- Una solución

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 los 23 patrones definidos por [GoF, 1994]:

"The design patterns in this book are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context."

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

Patrones de Diseño

Existen 3 clasificaciones:

- Creacionales
- Estructurales
- De comportamiento

Patrones de Diseño

Creacionales

- Se centran en la creación, composición y representación de los objetos

Estructurales

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

De comportamiento

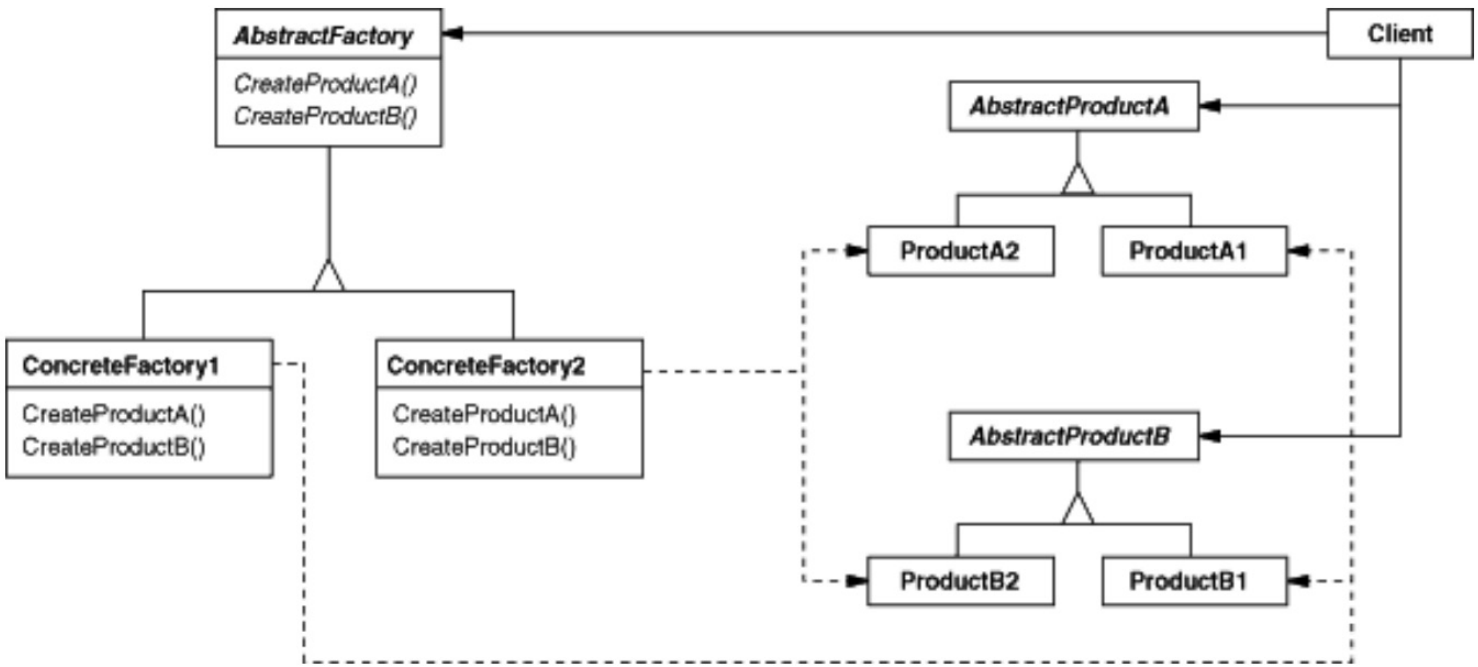
- Se centran en las interacciones y responsabilidades entre objetos

¿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

Abstract Factory

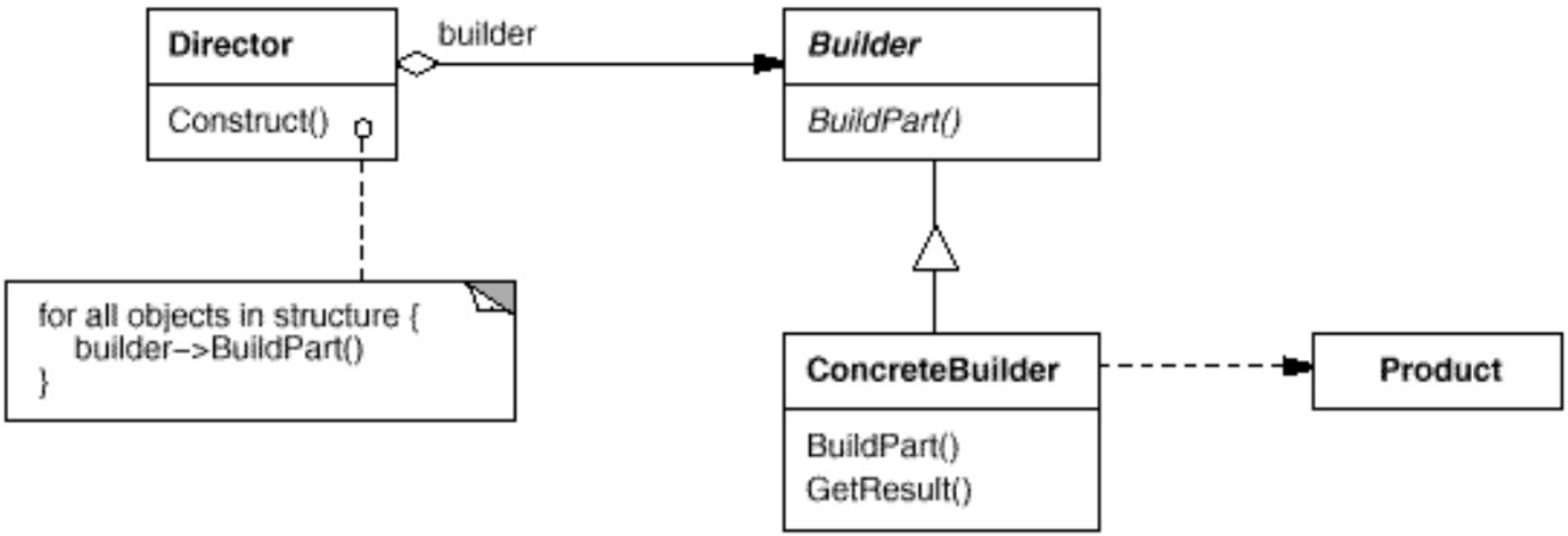
Provee una interfaz para la creación de familias de objetos relacionados o dependientes, sin especificar concretamente sus clases.



Ejemplo : Abstract Factory

Builder

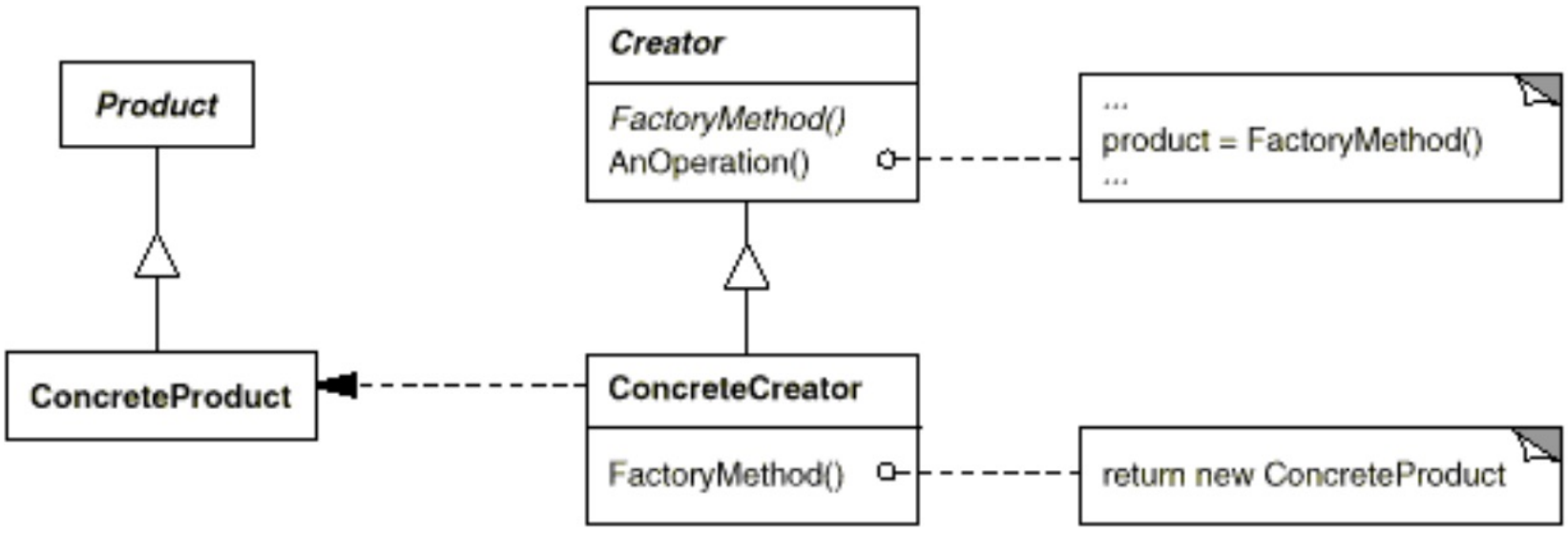
Separa la construcción de un objeto complejo de su representación, para que así el mismo proceso de construcción pueda crear diferentes representaciones.



Ejemplo : Builder

Factory Method

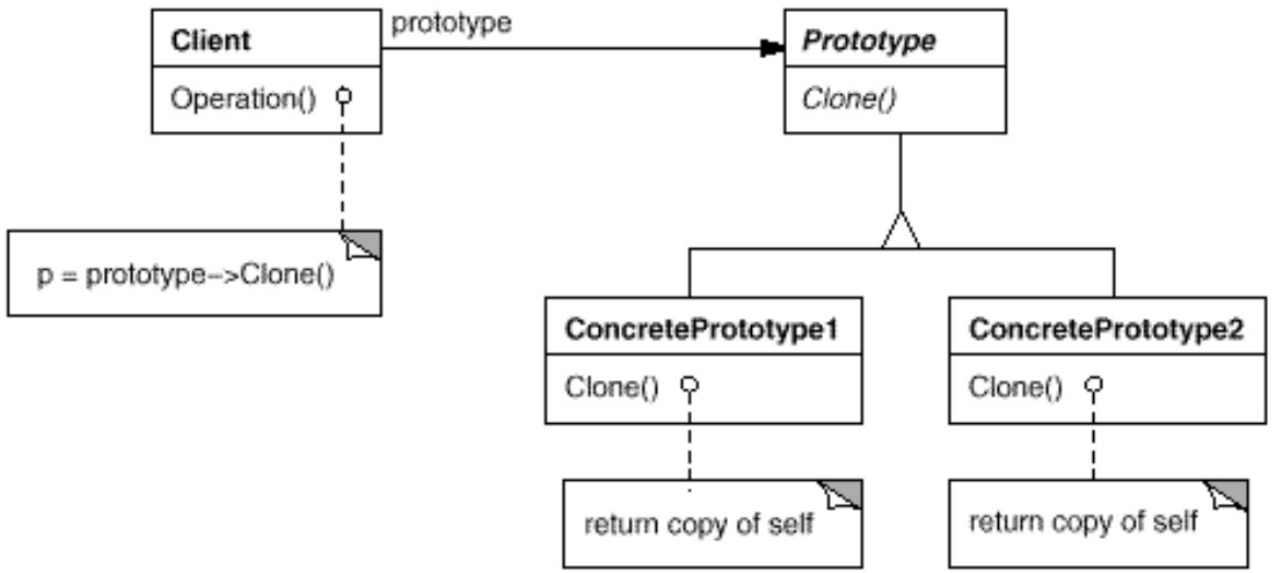
Define una interfaz para la creación de un objeto, pero delega a las subclases que decidan qué clase instanciar.



Ejemplo : Factory Method

Prototype

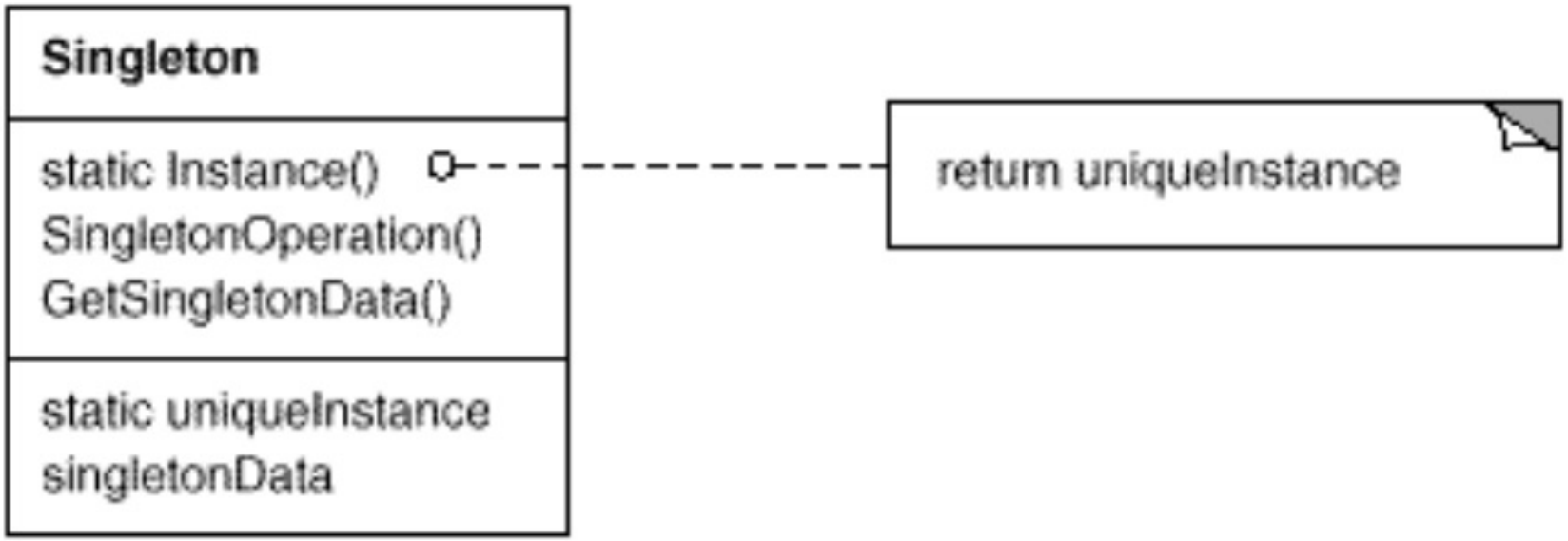
Especifica los tipos de objetos a crear utilizando prototipos, y crea objetos nuevos a través de copias de estos prototipos.



Ejemplo : Prototype

Singleton

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



Ejemplo : Singleton

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