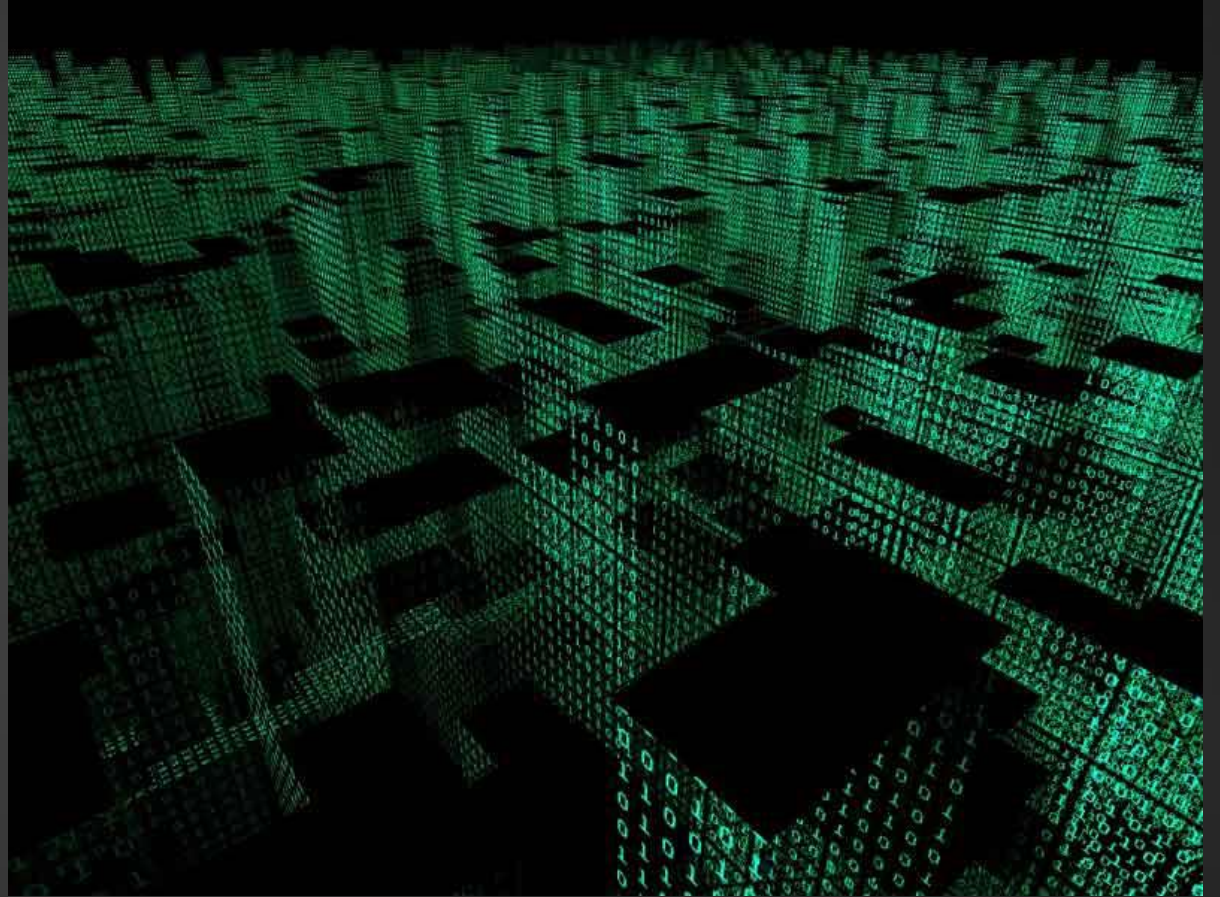


Prototype

PATRONES DE DISEÑO



Prototype

*El patrón **prototype** tiene como finalidad crear nuevos objetos duplicándolos, clonando una instancia creada previamente.*



Problema y Contexto:

En algunos casos, el coste de crear un objeto nuevo desde 0 es muy elevado, y más aún si luego hay que establecer una gran colección de atributos.

En éste contexto sería más conveniente clonar un objeto predeterminado que actúe de prototipo y modificar los valores necesarios para que se ajuste a su nuevo propósito.

Cuando utilizar este patrón :

Aplica donde sea necesario la creación de objetos parametrizados como "recién salidos de fábrica" ya listos para utilizarse, con la gran ventaja de la mejora de la performance: clonar objetos es más rápido que crearlos y luego setear cada valor en particular.

Debe ser utilizado cuando un sistema posea objetos con datos repetitivos, en cuanto a sus atributos: por ejemplo, si una biblioteca posee una gran cantidad de libros de una misma editorial, mismo idioma, etc. Hay que pensar en este patrón como si fuese un fábrica que tiene ciertas plantillas de ejemplos de sus productos y, a partir de estos prototipos, puede crear una gran cantidad de productos con esas características



Solución y Estructura:

La solución consistirá en definir una interfaz que expone el método necesario para realizar la clonación del objeto.

Las clases que pueden ser clonadas implementarán esta interfaz, mientras que las clases que deseen clonar deberán utilizar el método definido en la interfaz.

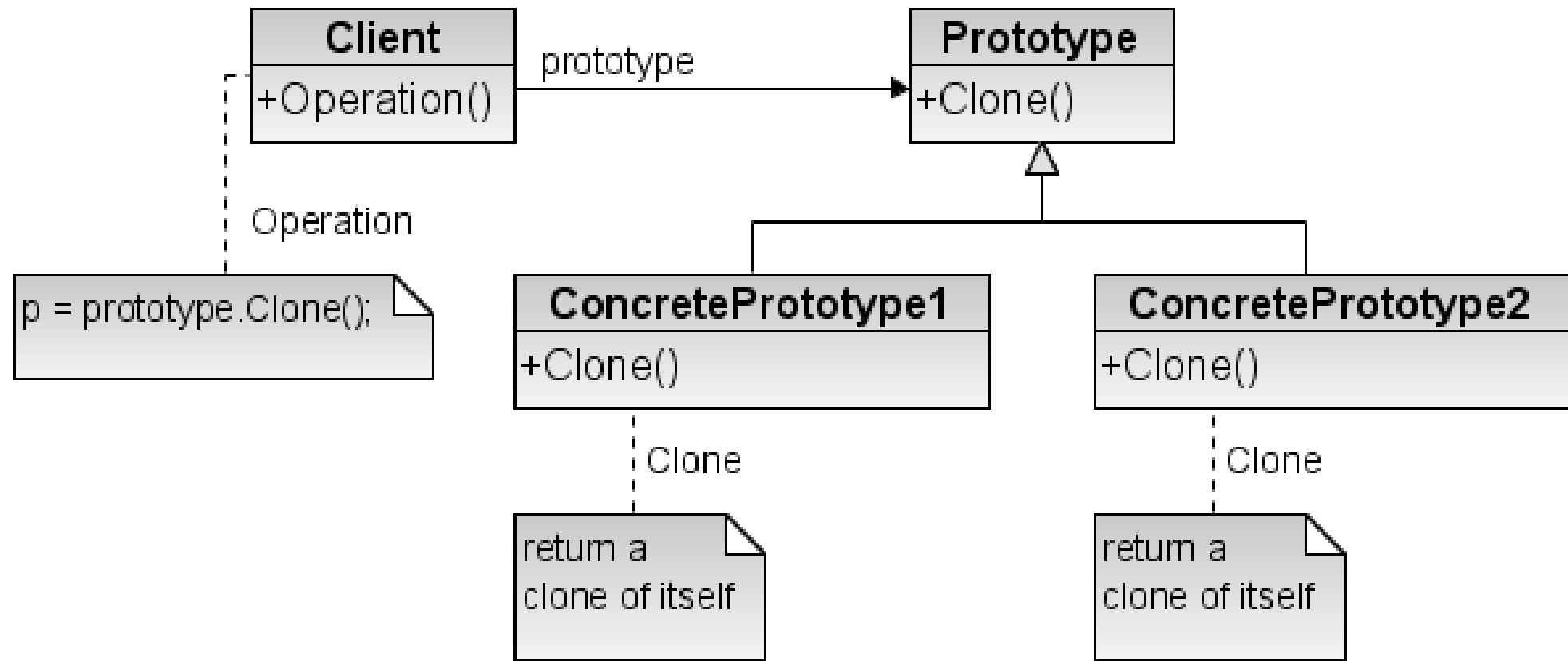
Consecuencias:

▶ POSITIVAS:

- ▶ Clonar un objeto es mucho más rápido que crearlo.
- ▶ Un programa puede añadir y borrar dinámicamente objetos prototipo en tiempo de ejecución.
- ▶ El cliente no debe conocer los detalles de cómo construir los objetos prototipo.

▶ NEGATIVAS

- ▶ En objetos muy complejos, implementar la interfaz Prototype puede ser muy complicada.



Prototype: Interfaz que define la operación de clonado. Será implementada por todos los objetos que puedan ser clonados. En ocasiones es implementado como una clase abstracta.

Client: Crea nuevos objetos mediante la clonación de los mismos.

ConcretePrototype1: Clase que puede ser instanciada mediante la clonación de un prototipo. Implementa la interfaz **Prototype**.