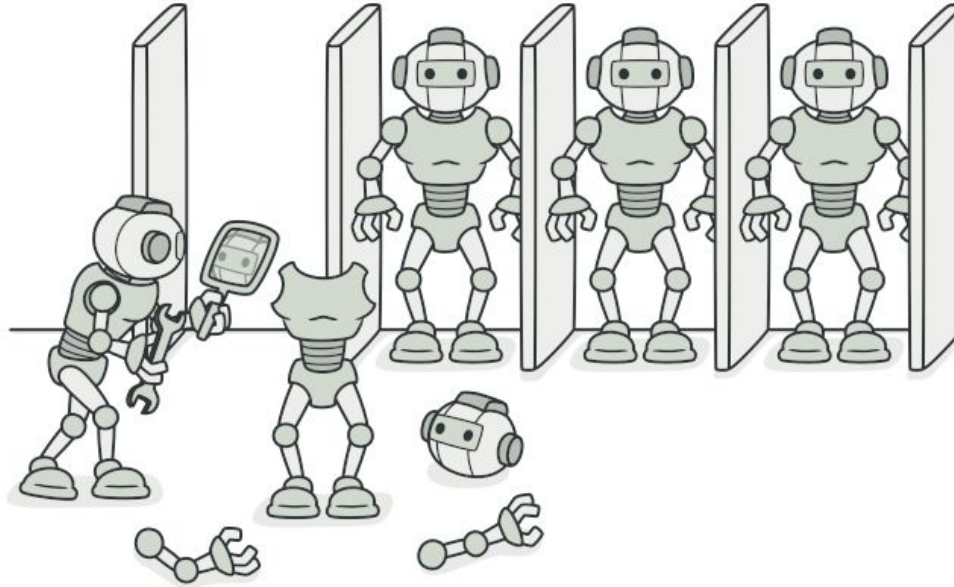


Prototype

Patrón de diseño creacional



Definición.

El patrón de diseño prototipo tiene como finalidad crear nuevos objetos clonando una instancia creada previamente.

Problema.



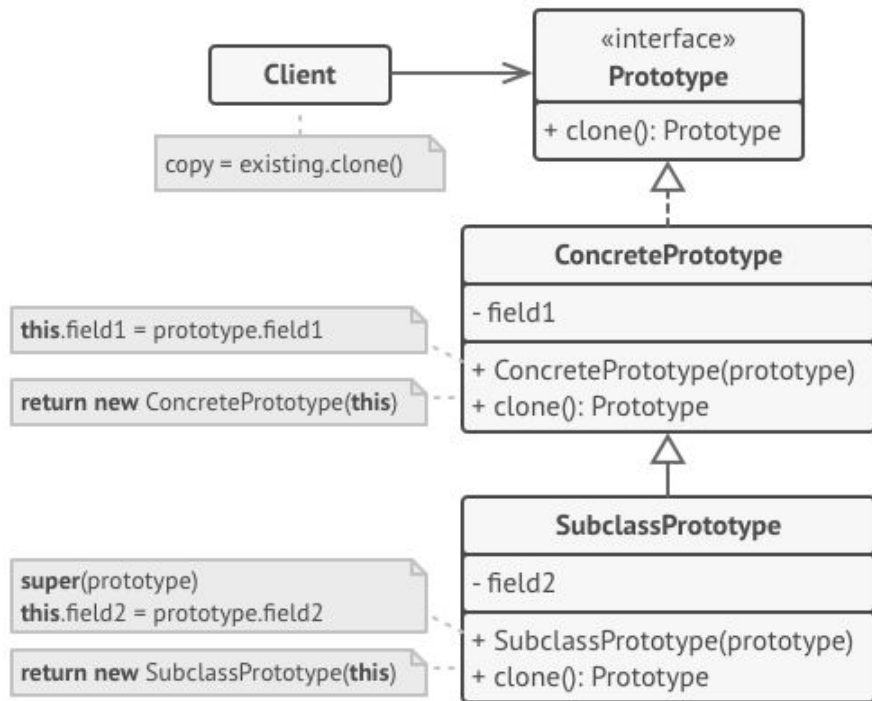
No siempre es posible copiar un objeto "desde fuera".

Se aplica cuando: La creación de nuevos objetos acarrea un coste computacional elevado. Los objetos a crear tienen o suelen tener atributos que repiten su valor.

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.

3 El **Cliente** puede producir una copia de cualquier objeto que siga la interfaz del prototipo.

1 La interfaz **Prototipo** declara los métodos de clonación. En la mayoría de los casos, se trata de un único método `clonar`.



2 La clase **Prototipo Concreto** implementa el método de clonación. Además de copiar la información del objeto original al clon, este método también puede gestionar algunos casos extremos del proceso de clonación, como, por ejemplo, clonar objetos vinculados, deshacer dependencias recursivas, etc.

Ventajas

- 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.

Desventaja.

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

Ejemplos de uso.

El patrón Prototype está disponible en Python listo para usarse con una módulo **copy**.

El módulo estándar copy permite crear copias de distintos objetos de Python, generalmente colecciones mutables (como las listas y los diccionarios) e instancias de clases, también mutables

Biografía.

[https://es.wikipedia.org/wiki/Prototipo_\(patr%C3%B3n_de_dise%C3%B1o\)](https://es.wikipedia.org/wiki/Prototipo_(patr%C3%B3n_de_dise%C3%B1o))

<https://refactoring.guru/es/design-patterns/prototype>

<https://jaimesendraberenguer.medium.com/patrones-de-dise%C3%B1o-en-python-ii-patrones-de-creaci%C3%B3n-b6d643edcead>