Prototype

Nombre:
Prototype

Clasificación del patrón:
Creacional

Intención:
Especifica el tipo de objetos a crear usando una instancia de tipo prototipo y crea nuevas instancias copiando ese prototipo.

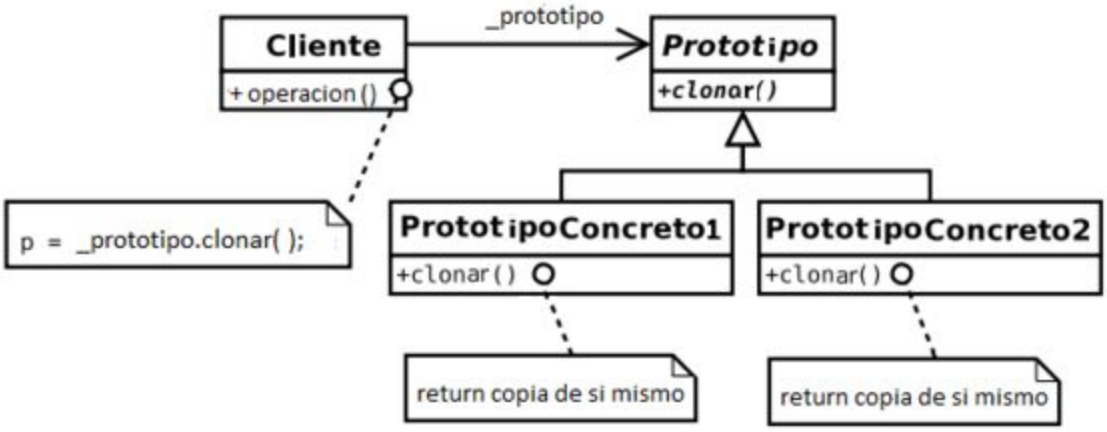
Otros nombres:
Prototipo

Motivación:
Es útil cuando se requiere abstraer la lógica que decide qué tipos de objetos serán utilizados por la aplicación y la lógica que van a usar estos durante la ejecución. Los motivos de hacer la separación pueden ser varios, como la dependencia de un parámetro obtenido en tiempo de ejecución. Ese prototipo será clonado y el nuevo objeto será una copia exacta con el mismo estado.

Aplicabilidad:
Se debe usar cuando:

- Se quiera lograr un acoplamiento bajo entre clases
- Se carguen las clases de manera dinámica en la aplicación
- Evitar la jerarquía paralela de fábricas para las clases de productos
- Evitar tener muchas clases que pueden tener muy pocas variaciones de estado

Estructura:



Participantes:

- Prototipo: Declara una interfaz para clonarse
- Prototipo Concreto: Implementa la operación para clonarse
- Cliente: Crea un objeto pidiendole a prototipo para que se clone

Colaboraciones:

Cliente crea un objeto prototipo para que se clone

Ventajas:

- Las clases concretas son ocultadas a los clientes. Esto promueve el bajo acoplamiento entre clases.
- Nuevas clases pueden ser añadidas en tiempo de ejecución.
- Nuevos objetos pueden ser especificados por sus diversos estados.
- Ahorra tiempo perdido durante la inicialización del objeto.

Desventajas:

- Todas las clases deben heredar de una clase base en común para que Clone() pueda ser implementada. Puede ser difícil para clases ya existentes.
- Algunos objetos no soportan la copia y por lo tanto éste modelo es inservible.

Implementación:

- Prototipo manager: La creación y destrucción constante de los prototipos puede ser almacenada en un archivo de registro, el cual será llamado Prototipo Manager; éste devolverá el prototipo asociado a una clave.
- Implementar la operación clonar. Se debe realizar una clonación profunda del objeto.

- Inicializar los clones.

Código de ejemplo:

- Bicicleta:

```java
package ar.com.patronesdisenio.prototype;

public abstract class Bicicleta implements Cloneable {

 private String color;
 private String rodado;

 /**
  * Metodo clonador
  */
 public Bicicleta clone() throws CloneNotSupportedException {
  return (Bicicleta) super.clone();
 }

 public abstract String verBicleta();

    //Getters and Setters
 public String getColor() {
  return color;
 }

 public void setColor(String color) {
  this.color = color;
 }

 public String getRodado() {
  return rodado;
 }

 public void setRodado(String rodado) {
  this.rodado = rodado;
 }

}
```

- Bicicleta modificada:

```java
package ar.com.patronesdisenio.prototype.impl;

import ar.com.patronesdisenio.prototype.Bicicleta;

/**
  * @author usuario
 */
public class BicicletaModificada extends Bicicleta {

 @Override
 public String verBicleta() {

  return "Este es el color: " + this.getColor() + " El rodado es: " + this.getRodado();
 }


}
```

- Cliente:

```java
package ar.com.patronesdisenio.prototype.impl;
import ar.com.patronesdisenio.prototype.Bicicleta;

/**
 * * @author usuario
 *
 */
public class Cliente {

 /**
  * @param args
  * @throws CloneNotSupportedException
  */
 public static void main(String[] args) throws CloneNotSupportedException {
  Bicicleta bc = new BicicletaModificada();
  bc.setColor("Roja");
  bc.setRodado("22");
  System.out.println(bc.verBicleta());

  Bicicleta bc2 = bc.clone();
  bc2.setColor("Negro");
  bc2.setRodado("30");

  System.out.println(bc2.verBicleta());

 }
}
```

Usos conocidos:

- Unidraw drawing
- Sketch pad
- ThingLab

Patrones relacionados:

- Abstract Factory
- Composite

Bibliografía:

No específico. (No específico). GoF Design Patterns (Versión 2.1.0) [Aplicación móvil]. Descargado de: https://drive.google.com/file/d/0BywiVyFlIabXcVhGZlJBcnhWTkU/view.

Patrones de Diseño - Patrón Prototype - ¿Que es el Patrón de Diseño Prototype/Prototipo? [Página web]. (28 de octubre de 2014.). Ubicación http://java-white-box.blogspot.com/2014/10/patrones-de-diseno-patron-prototype-que.html.

Farias Pinto, M, J. (10 de septiembre de 2018). Presentación del tema: "Patrones de diseño: Prototype"— Transcripción de la presentación:. Patrones de diseño: Prototype. https://slideplayer.es/slide/13728617/.

Junta de Andalucía. (s.f). Prototipo. Marco de Desarrollo de la Junta de Andalucía. http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/199.