DONCAN ALEJANDRO CASQUETE OLIVEROS

PATRONES DE DISEÑO DE SOFTWARE

PATRON PROTOTYPE

Patrón de Diseño: Prototype

# Definición

El patrón Prototype es un patrón de diseño creacional que permite crear nuevos objetos copiando instancias existentes. Esto es útil cuando la creación de un objeto es costosa o compleja, o cuando se desea evitar depender de clases concretas.

# Objetivo

Permitir la creación de nuevos objetos a partir de un prototipo existente mediante la clonación, sin acoplar el código a clases concretas.

# Explicación General del Código

El ejemplo proporcionado implementa el patrón Prototype con una clase llamada Usuario que puede clonarse. A continuación, se detalla el propósito de cada clase:

1. Sistema.java: Contiene el método main, donde se crean y clonan instancias de Usuario. Se crea un usuario base con un saldo de 200, luego se clona ese usuario y se imprimen los detalles de cada instancia.

2. Usuario.java: Representa un objeto Usuario con atributos de categoría y saldo. Implementa un método clonar() que utiliza el método clone() de Java. Si el objeto no soporta clonación, se lanza una excepción. El método toString() se sobreescribe para mostrar la información del objeto.

3. IUsuario.java: Es una interfaz que extiende Cloneable e impone la implementación del método clonar(). Esta interfaz ayuda a garantizar que cualquier clase que la implemente sea capaz de clonarse.

# Código Fuente

## Sistema.java

package Modelo;  
  
public class Sistema {  
  
 public static void main(String[] args) {  
 Usuario usuarioBase = new Usuario();  
 usuarioBase.setSaldo(200);  
  
 Usuario usuarioSinSaldo = new Usuario();  
 Usuario usuarioClonado = (Usuario) usuarioBase.clonar();  
  
 System.out.println(usuarioBase);  
 System.out.println(usuarioSinSaldo);  
 System.out.println(usuarioClonado);  
 }  
}

## Usuario.java

package Modelo;  
  
public class Usuario {  
 private String categoria;  
 private double saldo;  
  
 public Usuario(){  
 categoria = "regular";  
 }  
  
 public Usuario clonar(){  
 Usuario usuario = null;  
 try {  
 usuario = (Usuario) clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
 return usuario;  
 }  
  
 public String toString(){  
 return "Usuario [categoria=" + categoria + ", saldo=" + saldo + "]";  
 }  
  
 public void setSaldo(double saldo){  
 this.saldo = saldo;  
 }  
  
 public double getSaldo(){  
 return saldo;  
 }  
}

## IUsuario.java

package inter;  
  
public interface IUsuario extends Cloneable {  
  
 IUsuario clonar();  
   
}

# Funcionamiento del Código

El patrón Prototype se implementa mediante la interfaz IUsuario y el método clonar() en la clase Usuario. En el programa principal, se crea un usuario base con un saldo definido y se clona para generar otro usuario con los mismos datos. Esto demuestra cómo se pueden generar nuevas instancias sin recurrir al constructor directamente.

# Conclusión

Este ejemplo muestra de manera sencilla cómo se puede implementar el patrón Prototype en Java. Al utilizar clonación, se evita la repetición de código y se facilita la creación de objetos complejos. Además, se mantiene una mayor flexibilidad y se promueve el desacoplamiento entre clases.