Informe Método Abstract Factory

Joshua Abril

Patrones de Diseño

Luis Alberto Reyes

Uninpahu

2025

**¿Qué es?:**El **Abstract Factory** es un patrón que define una **superclase Factory** que contiene métodos para crear diferentes tipos de objetos relacionados. Luego, cada subclase concreta implementa estos métodos para producir una variante específica de los objetos. Es útil cuando una aplicación debe manejar múltiples familias de objetos sin acoplarse a clases específicas.

**Funcionalidad:** El **método Abstract Factory** es un patrón de diseño creacional que permite la creación de familias de objetos relacionados sin especificar sus clases concretas. A diferencia del **Factory Method**, que se centra en la creación de un solo tipo de objeto, Abstract Factory maneja múltiples tipos de objetos que están relacionados entre sí, proporcionando una mayor flexibilidad y separación de responsabilidades.

**Ejemplo breve:** Supongamos que estamos desarrollando una aplicación para diferentes temas de interfaz gráfica (claro y oscuro). Cada tema tiene sus propias versiones de botón y ventana. Con Abstract Factory, podemos tener una fábrica que cree tanto botones como ventanas según el tema seleccionado, sin que el código cliente necesite conocer las clases específicas.

**Código de Ejemplo:**

interface Boton {

void dibujar();

}

class BotonClaro implements Boton {

public void dibujar() {

System.out.println("Dibujando Botón Claro");

}

}

class BotonOscuro implements Boton {

public void dibujar() {

System.out.println("Dibujando Botón Oscuro");

}

}

interface Ventana {

void mostrar();

}

class VentanaClara implements Ventana {

public void mostrar() {

System.out.println("Mostrando Ventana Clara");

}

}

class VentanaOscura implements Ventana {

public void mostrar() {

System.out.println("Mostrando Ventana Oscura");

}

}

interface GUIFactory {

Boton crearBoton();

Ventana crearVentana();

}

class TemaClaroFactory implements GUIFactory {

public Boton crearBoton() {

return new BotonClaro();

}

public Ventana crearVentana() {

return new VentanaClara();

}

}

class TemaOscuroFactory implements GUIFactory {

public Boton crearBoton() {

return new BotonOscuro();

}

public Ventana crearVentana() {

return new VentanaOscura();

}

}

public class Main {

public static void main(String[] args) {

GUIFactory factory = new TemaClaroFactory(); Boton boton = factory.crearBoton();

Ventana ventana = factory.crearVentana();

boton.dibujar();

ventana.mostrar();

}

}

**Conclusión:**

El **Abstract Factory** permite **crear múltiples objetos relacionados sin acoplarse a clases específicas**. En este ejemplo, facilita la gestión de diferentes temas gráficos sin modificar el código cliente, promoviendo la flexibilidad y mantenibilidad del software.

**Explicación del Ejemplo del Repositorio:**

El repositorio implementa el patrón **Abstract Factory** en Java para la creación de figuras geométricas en 2D y 3D, permitiendo generar objetos sin depender de clases concretas. Se define una **fábrica abstracta (AbstractFactory)** que es implementada por dos fábricas concretas: FactoryFigura2D, que crea figuras como Círculo y Cuadrado, y FactoryFigura3D, que genera Esfera y Cubo. Además, FactoryProducer actúa como una **fábrica maestra**, eligiendo cuál fábrica usar según la dimensión solicitada. En Main, el código cliente obtiene la fábrica correspondiente, crea una figura y llama a dibujar (), mostrando su tipo en consola. Este diseño permite una creación organizada y escalable de objetos, facilitando la gestión de nuevas figuras sin modificar el código existente.