DONCAN ALEJANDRO CASQUETE OLIVEROS

PATRONES DE DISEÑO DE SOFTWARE

PATRON FACTORY

Patrón de Diseño: Factory en Java

# ¿Qué es el patrón Factory?

El patrón Factory es un patrón creacional que proporciona una interfaz para crear objetos en una superclase, pero permite a las subclases alterar el tipo de objetos que se crearán. Este patrón es útil cuando el código cliente no necesita saber qué clase concreta se está utilizando, sólo necesita saber que puede usar la interfaz común.

## Ventajas del patrón Factory

- Desacopla el código de creación de objetos del código cliente.

- Permite agregar nuevas clases sin modificar el código cliente.

- Mejora la organización y mantenimiento del código.

# Explicación del ejercicio en Java

Este ejercicio demuestra cómo usar el patrón Factory para crear diferentes tipos de conexiones a bases de datos como MySQL, Oracle, PostgreSQL, SQL y una conexión vacía por defecto.

# Códigos Fuente del Ejercicio

## App.java

package Pack1;  
  
import Pack2.IConexion;  
  
public class App {  
 public static void main(String[] args) {  
 ConexionFabrica fabrica = new ConexionFabrica();  
  
 IConexion cx1 = fabrica.getConexion("ORACLE");  
 cx1.conectar();  
 cx1.desconectar();  
  
 IConexion cx2 = fabrica.getConexion("MYSQL");  
 cx2.conectar();  
 cx2.desconectar();  
  
 IConexion cx3 = fabrica.getConexion("MYSQL");  
 cx3.conectar();  
 cx3.desconectar();  
 }  
}

## ConexionFabrica.java

package Pack1;  
import Pack2.IConexion;  
import Pack3.ConexionMysql;  
import Pack3.ConexionOracle;  
import Pack3.ConexionPostgreSQL;  
import Pack3.ConexionSQL;  
import Pack3.ConexionVacia;  
  
public class ConexionFabrica {  
 public IConexion getConexion(String motor) {  
 if (motor == null) {  
 return new ConexionVacia();  
 }  
 if (motor.equalsIgnoreCase("MYSQL")) {  
 return new ConexionMysql();  
 }  
 else if (motor.equalsIgnoreCase("ORACLE")) {  
 return new ConexionOracle();  
 }  
 else if (motor.equalsIgnoreCase("POSTGRE")) {  
 return new ConexionPostgreSQL();  
 }  
 else if (motor.equalsIgnoreCase("SQL")) {  
 return new ConexionSQL();  
 }  
 return new ConexionVacia();  
 }  
}

## IConexion.java

package Pack2;  
  
public interface IConexion {  
 void conectar();  
 void desconectar();  
}

## ConexionMysql.java

package Pack3;  
  
import Pack2.IConexion;  
  
public class ConexionMysql implements IConexion {  
  
 private String host;  
 private String puerto;  
 private String usuario;  
 private String contrasena;  
  
 public ConexionMysql(){  
 this.host = "localhost";  
 this.puerto = "1234";  
 this.usuario = "admin";  
 this.contrasena = "133";  
 }  
  
 public void conectar(){  
 System.out.println("Se Conecto a MySQL");  
 }  
  
 public void desconectar(){  
 System.out.println("Se Desconecto de MySQL");  
 }  
  
 public String toString(){  
 return "Conexion MySQL [Host= " + host + ", Puerto= "+ puerto + ",Usuario= "+ usuario + ", Contrasena= " + contrasena + "]";  
 }  
}

## ConexionOracle.java

package Pack3;  
  
import Pack2.IConexion;  
  
public class ConexionOracle implements IConexion {  
  
 private String host;  
 private String puerto;  
 private String usuario;  
 private String contrasena;  
  
 public ConexionOracle(){  
 this.host = "localhost";  
 this.puerto = "1234";  
 this.usuario = "admin";  
 this.contrasena = "123";  
 }  
  
 public void conectar(){  
 System.out.println("Se Conecto a Oracle");  
 }  
  
 public void desconectar(){  
 System.out.println("Se Desconecto de Oracle");  
 }  
  
 public String toString(){  
 return "Conexion Oracle [Host= " + host + ", Puerto= "+ puerto + ",Usuario= "+ usuario + ", Contrasena= " + contrasena + "]";  
 }  
}

## ConexionPostgreSQL.java

package Pack3;  
  
import Pack2.IConexion;  
  
public class ConexionPostgreSQL implements IConexion {  
  
 private String host;  
 private String puerto;  
 private String usuario;  
 private String contrasena;  
  
 public ConexionPostgreSQL(){  
 this.host = "localhost";  
 this.puerto = "1234";  
 this.usuario = "admin";  
 this.contrasena = "123";  
 }  
  
 public void conectar(){  
 System.out.println("Se Conecto a PostgreSQL");  
 }  
  
 public void desconectar(){  
 System.out.println("Se Desconecto de PostgreSQL");  
 }  
  
 public String toString(){  
 return "Conexion PostgreSQL [Host= " + host + ", Puerto= "+ puerto + ",Usuario= "+ usuario + ", Contrasena= " + contrasena + "]";  
 }  
}

## ConexionSQL.java

package Pack3;  
  
import Pack2.IConexion;  
  
public class ConexionSQL implements IConexion {  
  
 private String host;  
 private String puerto;  
 private String usuario;  
 private String contrasena;  
  
 public ConexionSQL(){  
 this.host = "localhost";  
 this.puerto = "1234";  
 this.usuario = "admin";  
 this.contrasena = "123";  
 }  
  
 public void conectar(){  
 System.out.println("Se Conecto a SQL");  
 }  
  
 public void desconectar(){  
 System.out.println("Se Desconecto de SQL");  
 }  
  
 public String toString(){  
 return "Conexion SQL [Host= " + host + ", Puerto= "+ puerto + ",Usuario= "+ usuario + ", Contrasena= " + contrasena + "]";  
 }  
}

## ConexionVacia.java

package Pack3;  
  
import Pack2.IConexion;  
  
public class ConexionVacia implements IConexion {  
  
 public void conectar() {  
 System.out.println("Conexion vacia: No se puede conectar");  
 }  
  
 public void desconectar() {  
 System.out.println("Conexion vacia: No se puede desconectar");  
 }  
  
 public String toString(){  
 return "ConexionVacia: No se estableció conexion.";  
 }  
}