DATA ACCESS OBJECT

Patrón DAO

- El patrón DAO nos permite acceder a datos que pueden estar localizados en distintas fuentes.
- La idea se basa en tres elementos:
  - **Objeto DAO**: es un simple POJO (*Plain Old Java Object*). Contiene la estructura de nuestros datos.
  - **Interfaz**: define todas las operaciones que se pueden realizar con un objeto DAO: insertar, modificar, eliminar, consultar, etc. En esta interfaz no habrá ninguna referencia a ninguna base de datos del tipo que sea.
  - **Clase que implemente la interfaz**: será la clase responsable de obtener los datos de un origen (base de datos, fichero …)

Ejemplo de aplicación

- Vamos a desarrollar los 3 elementos: objetoDAO, interfaz y una clase que implemente la interfaz
- Utilizaremos la base de datos Neodatis como fuente de datos
- Haremos una aplicación por componentes siguiendo el patrón DAO para gestionar departamentos.
- Los tres componentes del patrón formarán un paquete que podemos llamar Dep.

ObjetoDAO

- Vamos a crear una clase pública que llamaremos **Departamento** (Departamento.java)
- Implementará la interfaz **Serializable** que nos permitirá almacenar el estado de un objeto Departamento en un momento dado.
- Contendrá tres atributos:
  - int deptno → Número identificativo del departamento
  - String dnombre → Nombre del departamento
  - String loc → Nombre de la localidad donde está ubicado el departamento
- Tendrá dos constructores: Departamento() y Departamento(int deptno, String dnombre, String loc)
- Tendrá todos los “setters” y “getters” necesarios

Interfaz

- Añadimos a nuestra librería la interfaz DepartamentoDAO.java. La nueva interfaz tendrá las siguientes operaciones:
  - **public boolean InsertarDep (Departamento dep)** → recibe un objeto Departamento para insertarlo en la fuente de datos que sea. Devuelve *true* si la operación se ha realizado correctamente, en caso contrario devuelve *false*
  - **public boolean EliminarDep (int deptno)** → elimina el departamento de la fuente de datos. Recibe el número de departamento a eliminar. Devuelve *true* o *false* según el resultado de la operación
  - **public boolean ModificarDep (int deptno, Departamento dep)** → modifica el departamento indicado en deptno, los datos a modificar están en el objeto dep.
  - **public Departamento ConsultarDep (int deptno)** → recibe un número de departamento y devuelve el objeto departamento cuyo número coincida con deptno

Implementación de DepartamentoDAO

- Añadimos la librería JAR de Neodatis.
- Llamaremos a la clase **DepartamentoImpl** e implementará la interfaz DepartamentoDAO
- Tendrá una propiedad estática: *static ODB bd;*
- El constructor de la clase abrirá la base de datos (“Departamento.BD”)
- En esta clase se sobrescriben todos los métodos definidos en la interfaz anterior para la base de datos “Departamento.BD” gestionada por Neodatis. También se añade un método para obtener la conexión a la BD que previamente crea el constructor.

Clase que probará el DAO

- Una vez tenemos las 3 clases, generamos el JAR.
- Ahora, creamos en un nuevo proyecto (“dao_prueba”) añadimos el JAR creado y el JAR de Neodatis.
- Añadimos al nuevo proyecto una nueva clase que tendrá más o menos la siguiente estructura:

```
/* Empezará con una línea como la que sigue: */
DepartamentoDAO depDAO =  new DepartamentoImpl( );

/* 1-- Inserta un nuevo departamento */
/* 2-- Consulta el nuevo departamento */
/* 3-- Modifica algunos valores del nuevo departamento*/
/* 4-- Elimina el departamento creado */
```