UF5 - POO. Persistència

:: BBDD Orientades a Objectes ::

Exercici 1. Preparació de l'entorn db4o

- Crea un projecte UF6
- Crear directori /lib a l'arrel
- Copiar jar "db4o-8.0.249.16098-all-java5.jar" al directori
- Configura el Build Path del projecte (Configurar Build Path → Add jar)
- Crear directori /doc a l'arrel
- Mou la documentació (carpeta api)
- Configurar la documentació de la libreria de la base de dades (Build Path --> (Selecciona fitxer jar) → Javadoc location)

Exercici 2. Navegador d'objectes. Instal·lar OME (Object Manager Enterprise)

- Obté el plugin del navegador d'objectes per Eclipse a la carpeta /ome

/ome/ObjectManagerEnterprise-Java-8.0.0.zip

- Descomprimeix-lo, i des de Eclipse

Help > Install New Software > Add > Local > Seleccionar carpeta anterior

- Després d'instal·lar obrir la perspectiva OME i executa alguns dels exemples per veure'ls al navegador

Exercici 3. Template per a les consultes Native Query

- Afegeix un nou Template per l'editor de Java

Window -> Preferences -> Java -> Editor -> Templates -> New

- Les dades del Template són:

Nom: nq

Codi:

```
List <${extent}> list = db.query(new Predicate <${extent}> () {

        public boolean match(${extent} candidate){
                return true;
        }
});
```

- Es pot inserir el template a qualsevol punt del codi fent n+q+control+espai

Els exemples següents es treballaran amb el Model

Concessionaris + Vehicles

Exercici 4. Fes un programa que creï una base de dades en un fitxer anomenat: "DB4OexercicisUF6"

Crea un mètode de consulta de vehicles, per mostrar la informació dels vehicles utilitza el mètode "`infoVehicle()`"

```
public static void consultaVehicles(ObjectSet<Vehicle> vehicles) {     }
```

- Cada vegada que s'executa el programa s'ha de crear una base de dades nova
- Insereix dos cotxe
- Consulta tots els cotxes de la base de dades, utilitza QBE.
- Modifica un dels cotxes.
- Torna a consultar i comprova que s'ha modificat.
- Esborra el primer cotxe.
- Torna a consultar i comprova que s'ha esborrat.

Exercici 5. Consultes QBE

Crea els prototips per a les consultes següents, si alguna consulta no es pot fer amb QBE indica-ho

1. Tots els vehicles amb 10.000 km
2. Tots els cotxes amb 10.000 km
3. Tots els cotxes que no són clàssics
4. Totes les motos de cilindrada 500
5. Tots els vehicles amb matrícula 12345-ABC
6. Tots els cotxes amb matrícula 12345-ABC
7. Totes les motos amb més de 10000 km
8. Totes les motos amb 10000 km i de cilindrada 500
9. Tots els vehicles amb matrícula que comenci per 12
10. Tots els cotxes clàssics o amb més de 140.000 km

Exercici 5. Consultes natives (NQ)

Resol les consultes següents

1. Tots els vehicles amb 10.000 km
2. Tots els cotxes amb 10.000 km
3. Tots els cotxes que no són clàssics
4. Totes les motos de cilindrada 500
5. Tots els vehicles amb matrícula 12345-ABC
6. Tots els cotxes amb matrícula 12345-ABC
7. Totes les motos amb més de 10000 km
8. Totes les motos amb 10000 km i de cilindrada 500
9. Tots els vehicles amb matrícula que comenci per 12
10. Tots els cotxes clàssics o amb més de 140.000 km

Exercici 6. Ordre en les consultes natives (NQ)

Per afegir ordre cal implementar la interface "`QueryComparator`" que inclou un mètode "compare"

Per exemple

```
class NativeQuerySort implements QueryComparator<Vehicle>{

        public int compare(Vehicle v1, Vehicle v2)
        {
            return v1.getMatricula().compareTo(v2.getMatricula());
        }
}
```

La consulta que cal fer seria

```
List <Vehicle> vehicles = db.query(new Predicate<Vehicle>() {

         public boolean match(Vehicle v) {
             return true;
         }
}, new NativeQuerySort());
```

Fes una consulta de tots els vehicles ordenats per kilòmetres descendent i matrícula ascendent

Exercici 7. Consultes SODA

Resol les consultes següents amb l'api de SODA

1. Tots els vehicles amb 10.000 km
2. Tots els cotxes amb 10.000 km
3. Tots els cotxes que no són clàssics (De dues maneres)
4. Totes les motos de cilindrada 500
5. Tots els vehicles amb matrícula 12345-ABC
6. Tots els cotxes amb matrícula 12345-ABC
7. Totes les motos amb més de 10000 km
8. Totes les motos amb 10000 km i de cilindrada 500
9. Tots els vehicles amb matrícula que comenci per 12
10. Tots els cotxes clàssics o amb més de 140.000 km

Exercici 9. Associacions. Sense cascada

Per a fer les comprovacions calen dos programes diferents un que actualitza i l'altre que consulta.

Crea un mètode de consulta de concessionaris, per mostrar la informació dels concessionaris utilitza el mètode "`llistaVehicles()`"

```
public static void consultaConcessionaris(List<Concessionari> cons) {  }
```

- Programa 1 - Crea dos concessionaris amb varis vehicles cadascun
- Programa 2 - Consulta tots els concessionaris i tots els vehicles, indica què passa
- Programa 1 - Actualitza un vehicle de cada concessionari, desa els vehicles només
- Programa 2 - Consulta tots els concessionaris i tots els vehicles, observa què passa
- Programa 1 - Actualitza un vehicle de cada concessionari, desa els concessionaris només
- Programa 2 - Consulta tots els concessionaris i tots els vehicles
- Programa 1 - Esborra un vehicle d'un concessionari (mètode "`removeVehicle`"), desa el concessionari
- Programa 2 - Consulta tots els concessionaris i tots els vehicles, observa què passa
- Programa 1 - Esborra un vehicle de la base de dades, desa el vehicle
- Programa 2 - Consulta tots els concessionaris i tots els vehicles, observa què passa
- Com es podria esborrar un dels vehicles del concessionari i fer persistent el canvi?

Exercici 10. Associacions. Amb cascada.

Prepara la base de dades per actualitzar i eliminar en cascada.

- Programa 1 - Crea dos concessionaris amb varis vehicles cadascun
- Programa 2 - Consulta tots els concessionaris i tots els vehicles, indica què passa
- Programa 1 - Esborra un vehicle d'un concessionari (mètode "`removeVehicle`"), desa el concessionari
- Programa 2 - Consulta tots els concessionaris i tots els vehicles, quina de les dues operacions en cascada s'està aplicant?
- Programa 1 - Esborra un un concessionari
- Programa 2 - Consulta tots els concessionaris i tots els vehicles, quina de les dues operacions en cascada s'està aplicant?
- Com podem aconseguir pe exemple que s'esborrin els vehicles de la llista del concessionari però que no s'esborrin tots quan eliminem un concessionari?

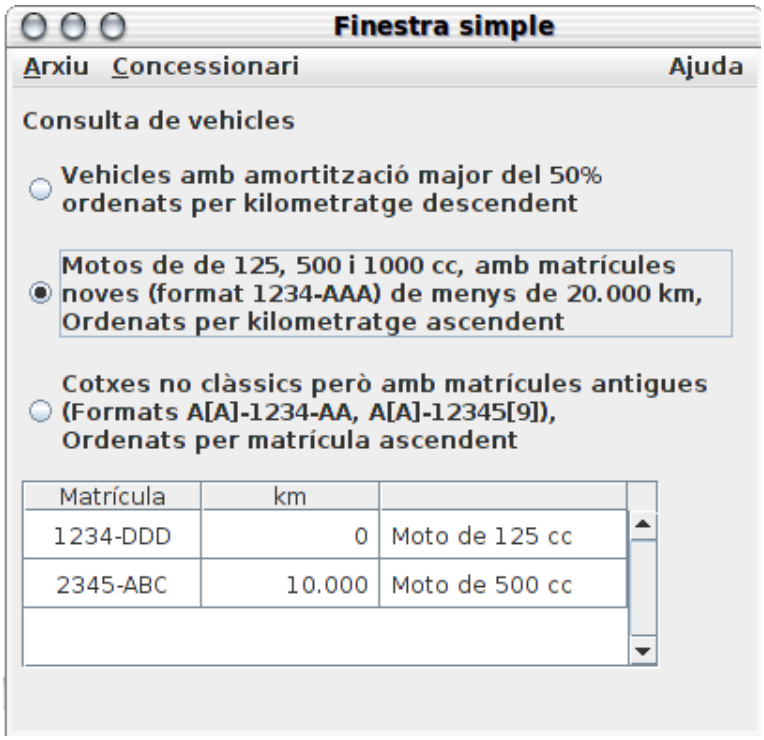
Exercici 11. Associacions. Consultes natives (NQ)

- Obté els concessionaris amb amortitzacions superiors al 30%
- Obté els concessionaris amb 3 o més vehicles
- Obté els concessionaris amb algun cotxe clàssic
    - Afegeix el mètode "isEsClassic() --> false" a Vehicle
    - Afegeix un getter per la llista de vehicles al Concessionari
- Obté els concessionaris amb algun vehicles de més de 10000 km que tinguin menys de 3 vehicles

Exercici 12. Proposta d'Entrega

Modifica l'aplicació gràfica de gestió de vehicles per afegir persistència

- El concessionari i els vehicles es guardaran en una base de dades "db4o" en un fitxer anomenat "cognoms_nom_bbdd"
- Aquest fitxer estarà dins del classpath del projecte (/src/uf6/bbdd/ cognoms_nom_bbdd) per exemple
- El programa detectarà si no existeix el fitxer per crear-lo si cal, però no el crearà nou cada vegada (Molt important)
- S'eliminaran les opcions de desar/obrir
- S'afegirà un botó per esborrar un concessionari, que substituirà el botó de crear quan s'hagi creat un nou concessionari
- S'afegirà la gestió de la persistència en totes les accions d'afegir o esborrar
- S'afegirà un nou panell per a mostrar el resultat de les consultes següents:



S'entregarà un fitxer zip que contindrà

- Projecte comprimit .jar i executable
- Fonts del projecte en un directori /src sense comprimir