



db4o datu basea

Sarrera

Laborategi honetan, db40 objetuetan oinarritutako datu basearen instalazioa ikusiko aztertuko dugu.

Helburuak

Laborategi honen helburuak hurrengoak dira:

- Db4o datu basearen instalazioa
- Datu base baten kudeaketa (ireki, itxi)
- Objektuak datu base batean gorde
- Objektuak datu base batetik berreskuratu

Jarraitzeko pausoak

1. Db4o datu basearen instalazioa

Erabiliko dugun bertsioa 7.4 izango da. Datu basea instalatzeko hurrengo pausoak jaraitu:

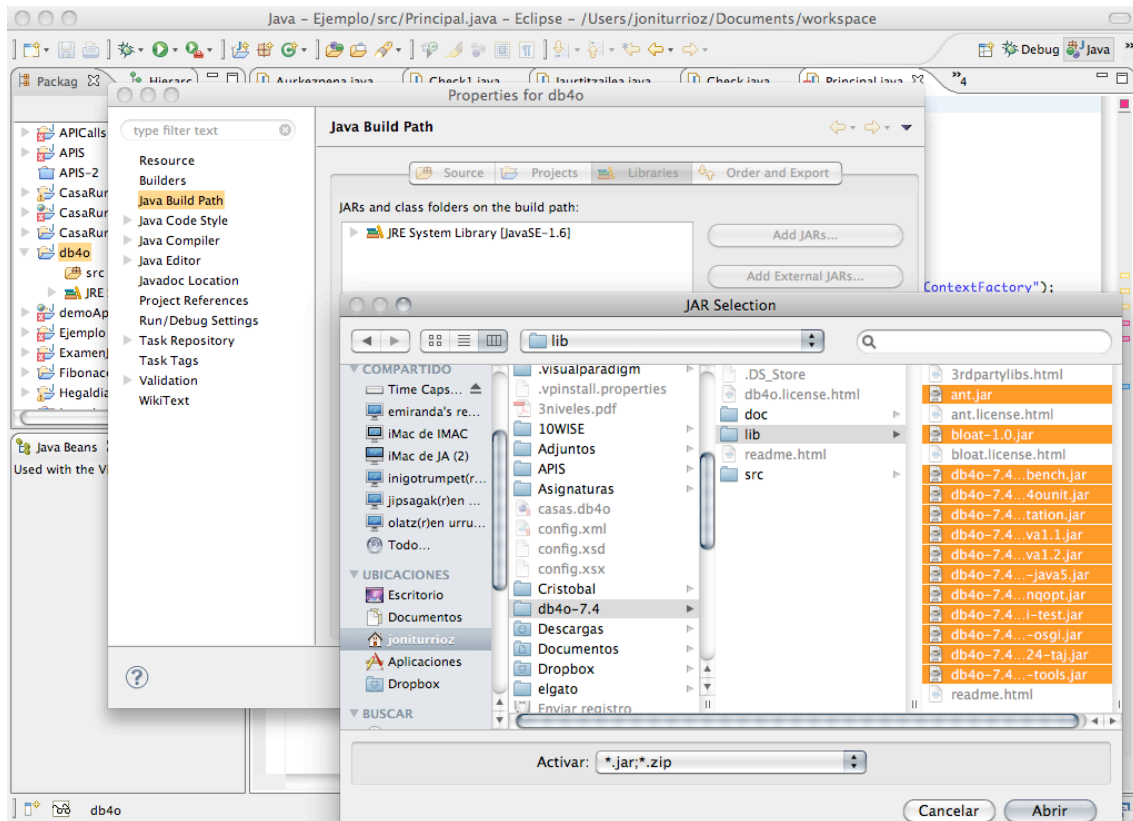
a) Kodea deskargatu. Erabiliko dugun bertsioa 7.4 izango da. Hurrengo helbidean aurkitu daiteke:

http://download.cnet.com/3001-10254_4-10338175.html?spi=dabf4e06c3cdd490e1e04d134cb7a0bf

b) Karpeta batean zip fitxategia deskonprimitu.

c)Java proiektu bat sortu Eclipse-n.

d) db4o datu basearen .jar liburutegiak kargatu sortutako proiektuan. Horretarako proiektuaren izenaren gainean ipini, eta eskuineko botoiarekin Properties aukeratu. Ageztzen den pantailan, ezkerreko partean "Java Build Path" sakatu eta eskuineko partean goiko partean "Libraries" aukeratu eta jarraian "Add external jar" botoia sakatu hurrengo irudian agertzen den bezala. Bukatzeko db4o deskonprimitu duzuen karpetatik "lib" azpikarpetan dauden .jar fitxategi guztiak aukeratu eta kargatu.



2. Java klase bat sortu "main" metodo batekin.

Java klasea sortu eta datu bat ireki eta itxi metodoak definitu, hurrengo kodean agertzen den bezala.

```
public static void main(String[] args) {
    String DB4OFILENAME="c:\\formula1.db4o";
    ObjectContainer db = Db4o.openFile(Db4o.newConfiguration(),
DB4OFILENAME);
    try {
        //do something
    } finally {
        db.close();
    }
}
```

3. Objektuak sartu datu base batean

Pilot klaseko objektuak gordeko ditugu datubasean. Pilot klasearen kodea hurrengo da:

```
public class Pilot {
    String name;
    int points;
    public Pilot(String name, int x){
        this.name=name;
        this.points=x;
    }
    public String toString(){
        return name+" "+Integer.toString(points);
    }
    public void addPoints(int x){
        this.points=x;
    }
}
```

Horretarako Pilotoak gordetzen dituen metodoa definituko dugu klase nagusian hurrengo kodearekin:

```
public static void storePilot(ObjectContainer db,String
name,int points) {
    Pilot pilot = new Pilot(name, points);
    db.store(pilot);
    System.out.println("Stored " + pilot);
}
```

eta hurrengo kodea ipiniko dugu `try { }` barnean bi piloto sortu eta gordetzeko.

```
storePilot(db,"Michael Schumacher", 100);
storePilot(db,"Rubens Barrichello", 99);
```

4. Galderak egin datu baseari

Piloto guztiak berreskuratzeko hurrengo metodoa definitu dezakegu.

```
public static void retrieveAllPilotQBE(ObjectContainer db) {  
    Pilot proto = new Pilot(null, 0);  
    ObjectSet result = db.queryByExample(proto);  
    listResult(result);  
}
```

eta hurrengo "listResult" metodoa emaitzak inprimatzeko.

```
public static void listResult(ObjectSet s){  
    Pilot p;  
    while (s.hasNext()) {  
        p=(Pilot)s.next();  
        System.out.println(p);  
    }  
}
```

Jarraian `try { }` kodean ezabatu aurreko instrukzioak (bestela beste bi Piloto sortuko ditu) eta deitu aurreko metodoari Pilotoak berreskuratzeko.



5. Objektuak ezabatu

Hurrengo kodea izena bat emanda, Piloto bat ezabatzen du.

```
public static void deletePilotByName(ObjectContainer db,  
String name) {  
    ObjectSet result = db.queryByExample(new Pilot(name,  
0));  
    Pilot found = (Pilot) result.next();  
    db.delete(found);  
    System.out.println("Deleted " + found);  
}
```

Ariketa bezala, "Michael Schumacher" pilotoa ezabatu, eta galdetu datu baseari Piloto guztien informazioari buruz.

Ariketa:

Labotegi 8-an login eta password-a frogatzen zuen aplikazio bat garatu genuen. Nola abertastu dezakezu aplikazioa, frogatzeko datuak db4o datu base batean gordeta badaude.