

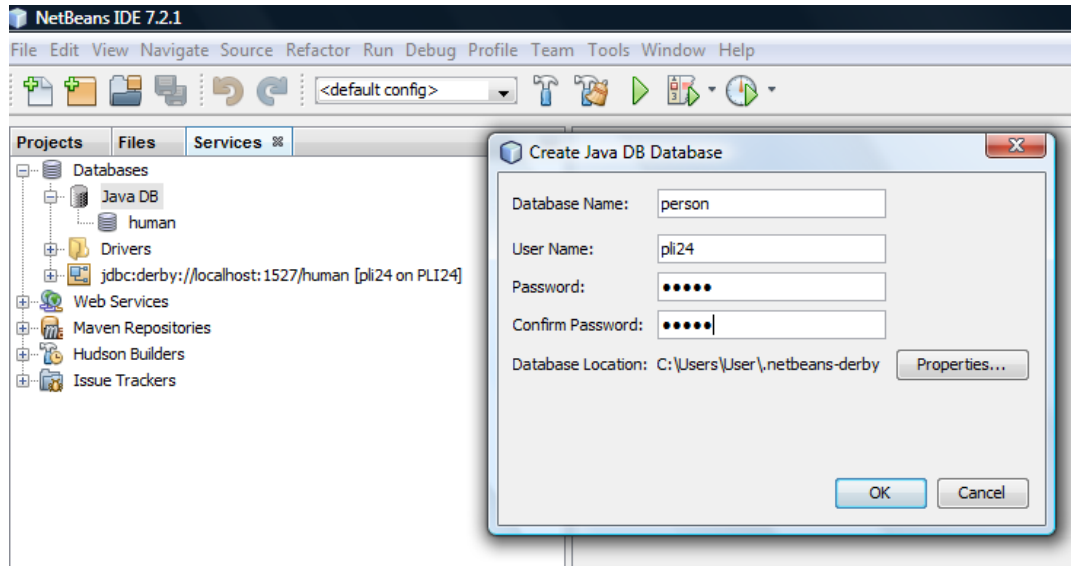
Ακολουθεί ένα σύντομο παράδειγμα δημιουργίας βάσης δεδομένων στο Netbeans, δημιουργία entity class από αυτή, αποθήκευσης αντικειμένων στη βάση και στη συνέχεια ανάκτησής τους και εγγραφής σε XML αρχείο.

Αρχικά θα δημιουργηθεί μια βάση δεδομένων με το όνομα **person** και ένα πίνακα με το όνομα **persinf** ο οποίος θα κρατάει 2 δεδομένα για κάθε πρόσωπο (τον κωδικό του και το όνομά του). Στη συνέχεια θα δημιουργηθεί ένα entity class από αυτόν το πίνακα με το όνομα Persinf. Στη main θα δημιουργήσουμε 2 αντικείμενα Persinf και θα τα αρχικοποιήσουμε δίνοντας κωδικό και όνομα. Μετά θα τα αποθηκεύσουμε στη βάση δεδομένων. Στη συνέχεια θα πάμε να διαβάσουμε όλα τα αντικείμενα Persinf που είναι αποθηκευμένα στη βάση και θα αποθηκεύσουμε τις πληροφορίες τους σε ένα αρχείο xml κατά τον ακόλουθο τρόπο:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <persons>
-   <person id="5">
      <name>george</name>
    </person>
-   <person id="6">
      <name>matina</name>
    </person>
  </persons>
```

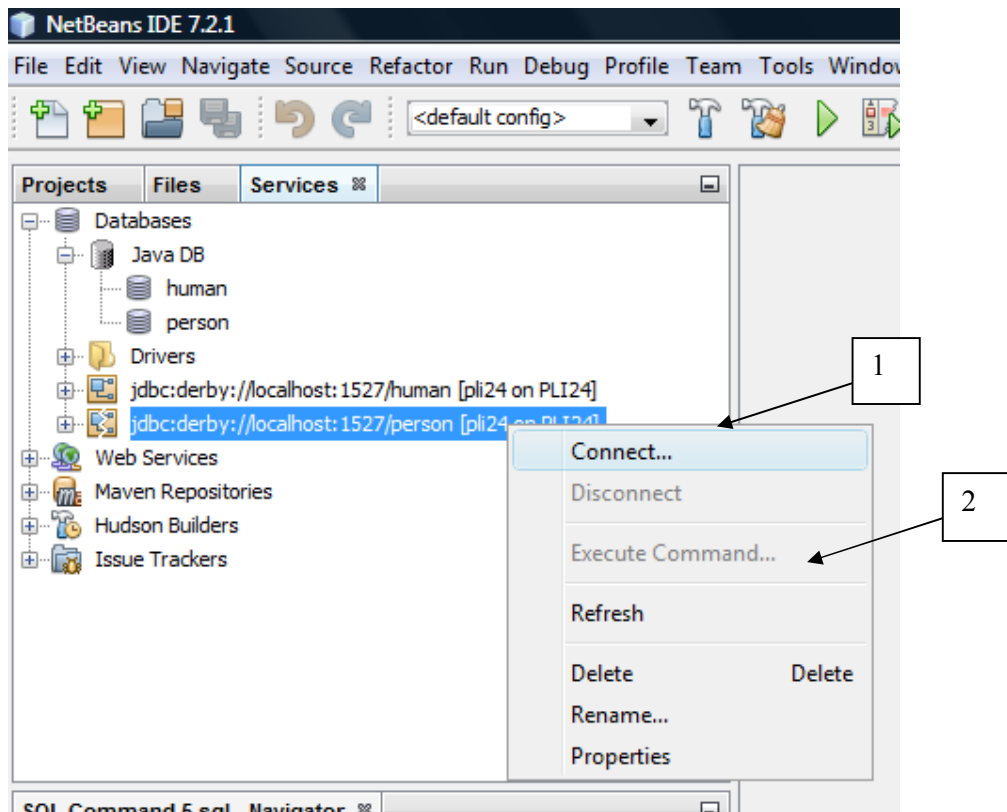
Αρχικά δημιουργούμε μια καινούργια βάση δεδομένων με τα ακόλουθα βήματα:

- Πάτε στο tab Services-> Databases -> Java DB -> Create Database

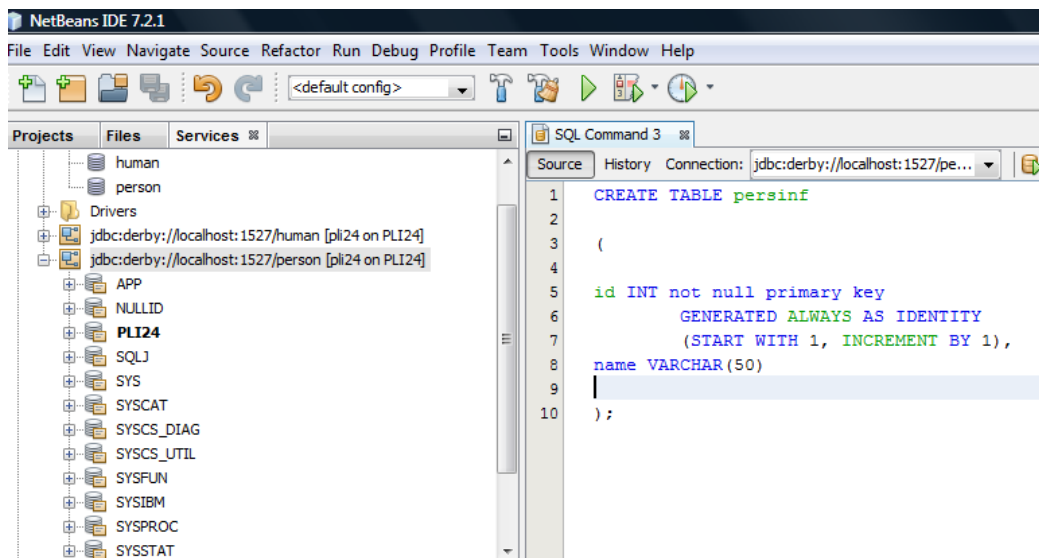


Στο παράθυρο διαλόγου που εμφανίζεται θα πρέπει να εισάγετε το όνομα της βάσης (person) username και password (εγώ έβαλα και στα 2 pli24). Πατάτε OK

Στη συνέχεια πάτε πάνω στην αντίστοιχη σύνδεση της βάσης και την κάνετε ενεργή με δεξί κλικ και 1. connect. Μετά πάλι δεξί κλικ πάνω στην σύνδεση της βάσης και επιλέγετε 2. Execute Command

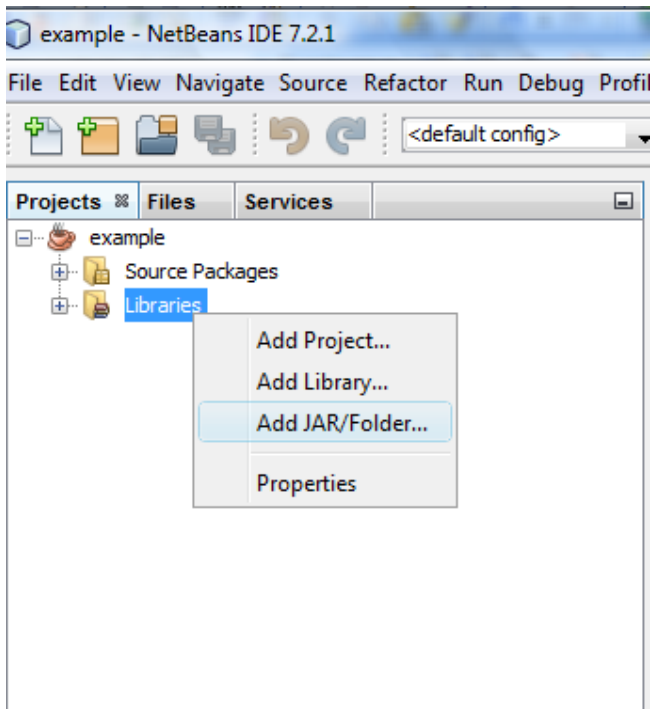


Στη συνέχεια στο παράθυρο που εμφανίζεται γράφεται τον κώδικα sql για την δημιουργία των πινάκων της βάσης
Εγώ δημιούργησα ένα πίνακα με το όνομα persinf και 2 πεδία το id(πρωτεύον κλειδί που αυξάνεται αυτόματα) και το name το όνομα του ατόμου.

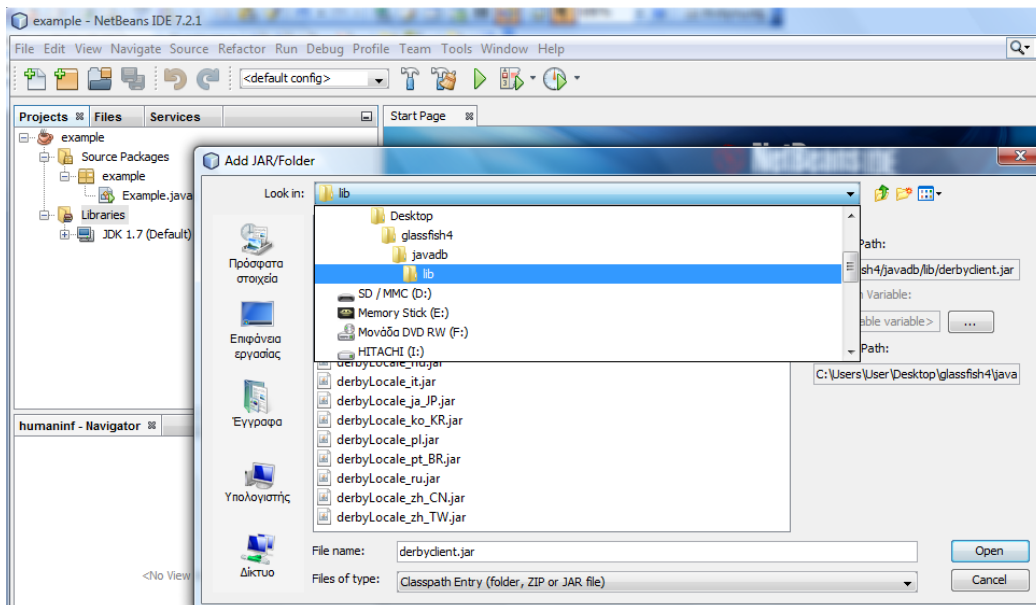


Τελείωσα με τη δημιουργία της βάσης μου.

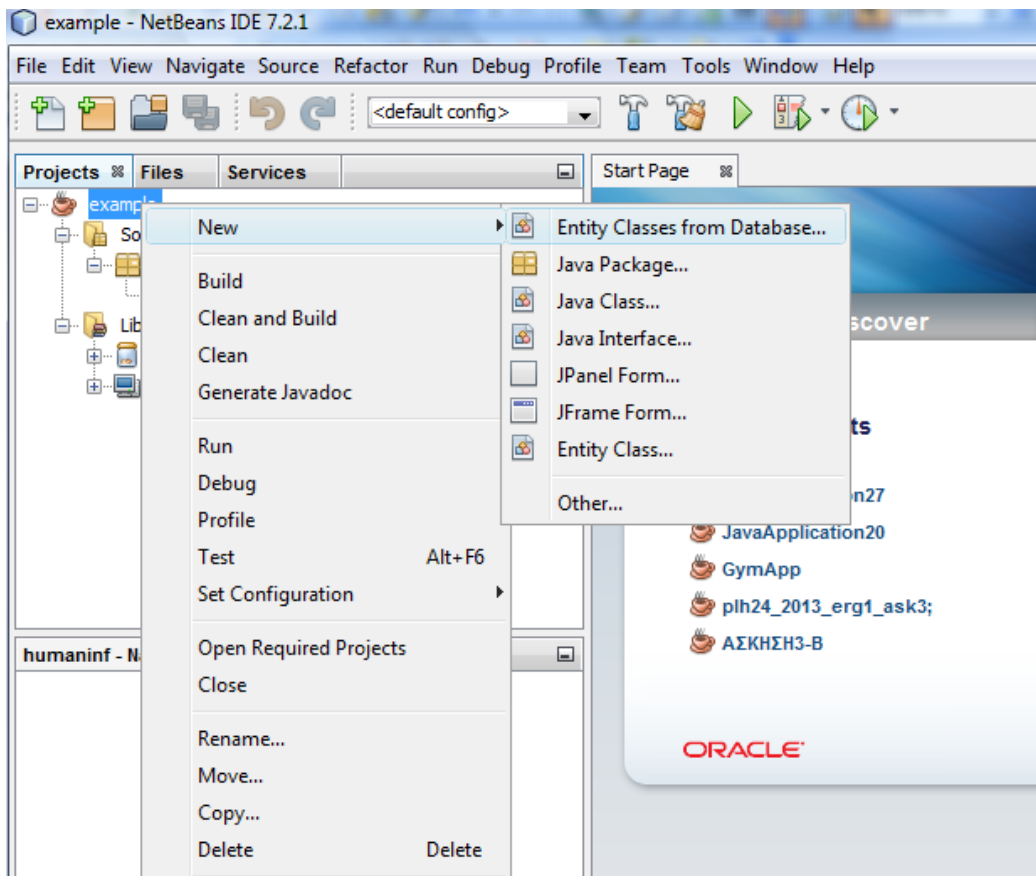
Τώρα πάω να δημιουργήσω ένα καινούργιο project με το γνωστό τρόπο File-> New Project-> Java Application -> example.
Στη συνέχεια εισάγω το .jar για τη βάση derby. Επιλέγω το Libraries->δεξί κλικ->Add JAR/Folder



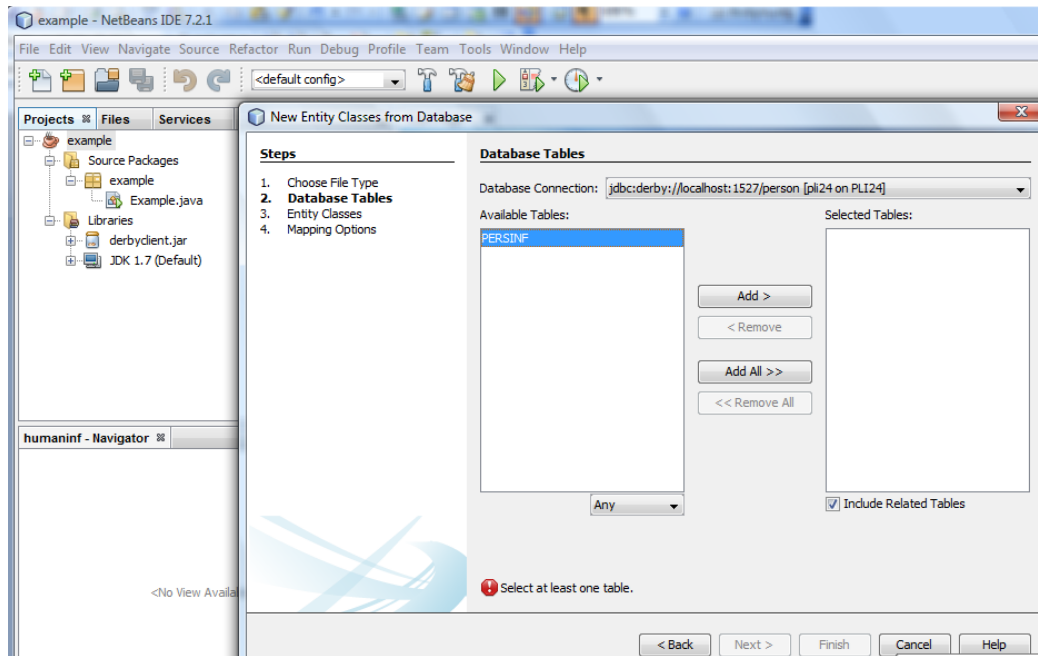
Επιλέγω τον αντίστοιχο φάκελο του glassfish (glassfish->javadb->lib->derbyclient.jar) και πατάω open. Αν δεν έχετε κατεβάσει το glassfish μπορείτε να το βρείτε εδώ <https://glassfish.java.net/download.html>



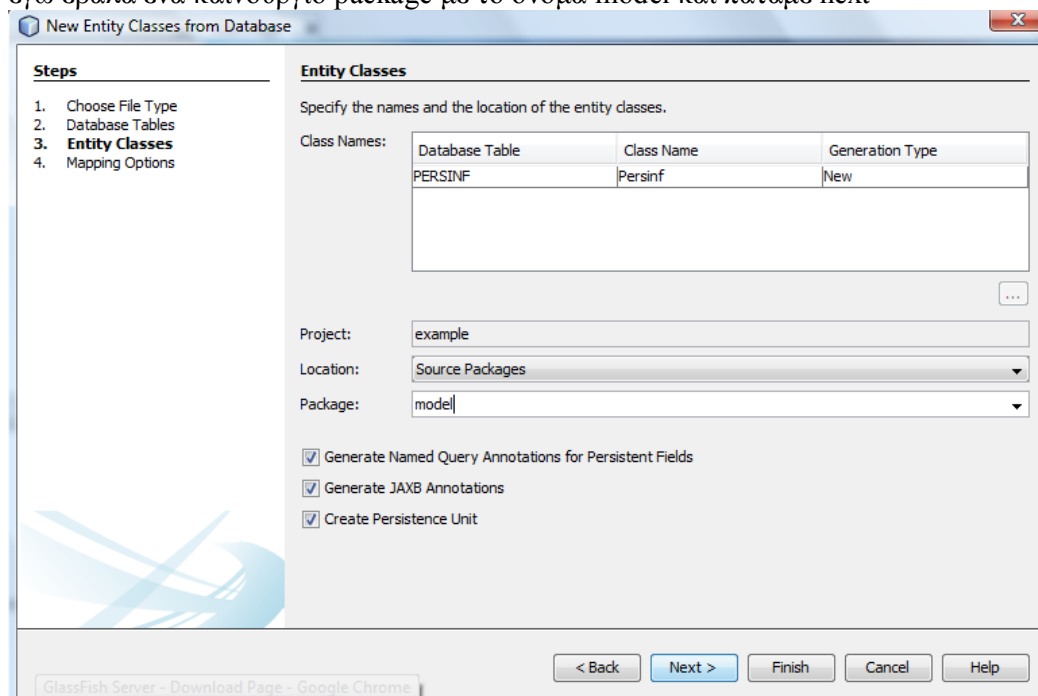
Στη συνέχεια πάω να δημιουργήσω τις entity classes. Στην προκειμένη περίπτωση είναι μόνο μια η Persinf. Πάω πάνω στο project μου (example) και με δεξί κλικ επιλέγω το Entity Classes from Database.



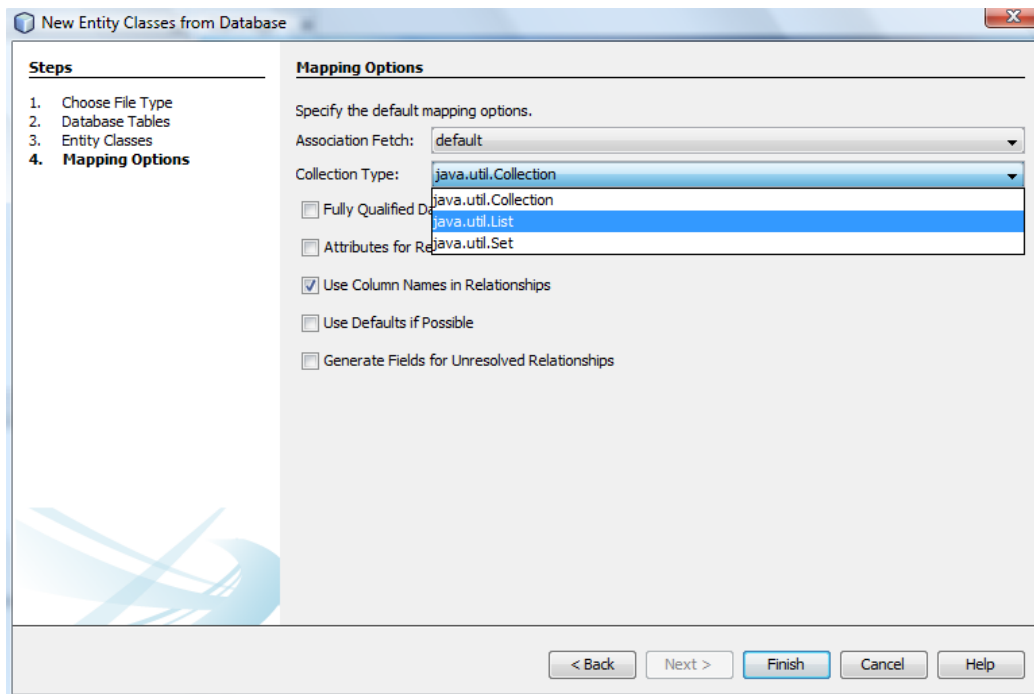
Στο παράθυρο που βγαίνει 1. διαλέγετε την κατάλληλη σύνδεση, 2. επιλέγετε πίνακες και κάνετε 3. add τους πίνακες που σας ενδιαφέρουν, πατάτε 4. next



Στη συνέχεια επιλέγεται το package στο οποίο θα δημιουργηθούν τα entity classes, εγώ έβαλα ένα καινούργιο package με το όνομα model και πατάμε next



Στο επόμενο παράθυρο επιλέγουμε το java.util.List και πατάμε Finish



Τώρα έχουμε δημιουργήσει αυτόματα την ακόλουθη entity class

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package model;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author User
 */
@Entity
@Table(name = "PERSINF")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Persinf.findAll", query = "SELECT p FROM Persinf p"),
```

```

    @NamedQuery(name = "Persinf.findById", query = "SELECT p FROM Persinf p
    WHERE p.id = :id"),
    @NamedQuery(name = "Persinf.findByName", query = "SELECT p FROM
    Persinf p WHERE p.name = :name"))
    public class Persinf implements Serializable {
        private static final long serialVersionUID = 1L;
        @Id
        @GeneratedValue(strategy = GenerationType.IDENTITY)
        @Basic(optional = false)
        @Column(name = "ID")
        private Integer id;
        @Column(name = "NAME")
        private String name;

        public Persinf() {
        }

        public Persinf(Integer id) {
            this.id = id;
        }

        public Integer getId() {
            return id;
        }

        public void setId(Integer id) {
            this.id = id;
        }

        public String getName() {
            return name;
        }

        public void setName(String name) {
            this.name = name;
        }

        @Override
        public int hashCode() {
            int hash = 0;
            hash += (id != null ? id.hashCode() : 0);
            return hash;
        }

        @Override
        public boolean equals(Object object) {
            // TODO: Warning - this method won't work in the case the id fields are not set
            if (!(object instanceof Persinf)) {
                return false;
            }

```



```

        Persinf other = (Persinf) object;
        if (((this.id == null && other.id != null) || (this.id != null
&& !this.id.equals(other.id))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "model.Persinf[ id=" + id + " ]";
    }
}

```

Στο παράδειγμά μου δεν χρειάζεται να προσθέσω κάτι στη κλάση Persinf. Προχωρώ στη main.

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package example;
import java.io.File;
import model.*; // κάνω import το πακέτο με τις entity classes
import java.util.List; // κάνω import το πακέτο για την δημιουργία λίστας θα
// χρειαστεί όταν θα διαβάσω από τη βάση όλα τα αντικείμενα Persinf

// στη συνέχεια κάνω import τις βιβλιοθήκες JPA για τη διαχείριση entities
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.persistence.TypedQuery;

// στη συνέχεια κάνω import τις βιβλιοθήκες για τη διαχείριση xml αρχείων
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.Attr;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
/**
 *

```

```

* @author User
*/
public class Example {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Persinf p1= new Persinf(); // δημιουργώ ένα νέο αντικείμενο, δεν χρειάζεται να
        δώσω id γιατί κατά τη δημιουργία αντικειμένου το id που είναι πρωτεύον κλειδί
        αυξάνεται αυτόματα
        p1.setName("matina"); // αρχικοποιώ το όνομά του αντικειμένου
        Persinf p2= new Persinf(); // αρχικοποιώ 2ο αντικείμενο
        p2.setName("george");

        EntityManager em; // δημιουργώ μια μεταβλητή entity manager
        EntityManagerFactory emf =
        Persistence.createEntityManagerFactory("examplePU"); // κάνω σύνδεση με το
        persistence unit (θα το βρείτε στο αρχείο persistence.xml μέσα στο φάκελο META-
        INF πεδίο Persistence Unit Name)
        em = emf.createEntityManager(); //αρχικοποιώ τη μεταβλητή em
        em.getTransaction().begin(); //ξεκινάω μια καινούργια συναλλαγή για να
        αποθηκεύσω στη βάση δεδομένων τα αντικείμενα p1, p2.

        em.persist(p1); // δημιουργώ το query εισαγωγής για το p1
        em.persist(p2); // δημιουργώ το query εισαγωγής για το p2
        em.getTransaction().commit(); // κάνω commit το query

        // στη συνέχεια πάω να δημιουργήσω ένα αρχείο όπου θα αποθηκεύσω όλα τα άτομα
        (Persinf) που είναι στη βάση.
        try {
            // Δημιουργώ ένα καινούργιο αρχείο

            DocumentBuilderFactory docFactory =
            DocumentBuilderFactory.newInstance(); // Δημιουργία parser που παράγει το
            DOM XML αρχείων
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder(); //
            Δυνατότητα δημιουργίας των στιγμιότυπων του XML αρχείου από το DOM
            Document doc = docBuilder.newDocument(); // αναφέρεται στο XML αρχείο,
            είναι η ιδεατή ρίζα του δέντρου

            // δημιουργώ την ρίζα persons
            Element rootElement = doc.createElement("persons"); // δημιουργία tag persons
            doc.appendChild(rootElement); // προσθήκη tag ως παιδί του κόμβου
            rootElement

            // ανακτώ από τη βάση σε μια λίστα όλα τα αντικείμενα Persinf και τα αποθηκεύω
            στη μεταβλητή results. Χρησιμοποιώ το έτοιμο query Persinf.findAll

            TypedQuery<Persinf> query =

```

```

        em.createNamedQuery("Persinf.findAll", Persinf.class);
        List<Persinf> results = query.getResultList();
        // Στη συνέχεια προσπελάζω τη λίστα με τα αντικείμενα Persinf και για κάθε
        αντικείμενο δημιουργώ ένα καινούργιο κόμβο στο xml δέντρο που έχει τα στοιχεία
        του κάθε ατόμου που είναι αποθηκευμένο στη βάση.
        for (Persinf so:results)

            // System.out.println(so.getName());
            {
                Element person = doc.createElement("person"); // δημιουργώ ένα
                στοιχείο με το όνομα person
                rootElement.appendChild(person); // τον συνδέω με τον τρέχοντα
                κόμβο
                Attr attr = doc.createAttribute("id"); // ο κόμβος αυτός έχει ένα
                χαρακτηριστικό το id (είναι η ιδιότητα id του Persinf)
                attr.setValue(Integer.toString(so.getId())); // σαν τιμή θα πάρει id του
                Persinf, για αυτό καλώ τη συνάρτηση getId()
                person.setAttributeNode(attr);
                Element name = doc.createElement("name");
                name.appendChild(doc.createTextNode(so.getName())); // δημιουργώ
                τον κόμβο με το όνομα του ατόμου
                person.appendChild(name);
            }

            // γράφω όλα τα παραπάνω σε ένα αρχείο xml
            TransformerFactory transformerFactory = TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            DOMSource source = new DOMSource(doc);

            StreamResult result;
            //δημιουργώ το αρχείο xml
            result = new StreamResult(new File("file.xml"));

            transformer.transform(source, result);

            System.out.println("File saved!");

            } catch (ParserConfigurationException pce) {
                pce.printStackTrace();
            } catch (TransformerException tfe) {
                tfe.printStackTrace();
            }
        }
    }
}

```