

```
1  Persistence
2  En P00, la persistance est la propriété permettant à un objet de continuer à
3  exister après la des-
4  truction de son créateur.
5  – C’est la capacité de sauvegarder l’état des objets, i.e. les données finales de
6  l’application
7  – La persistance est dite orthogonale ou transparente si la propriété est
8  intrinsèque à l’environnement d’exécution
9
10 Serialisation
11 La sérialisation permet la transformation d’un objet en un flux d’octets.
12 – Permet le stockage des objets sur disque, leur transmission par le réseau, . . .
13 – L’opération inverse se nomme désérialisation
14 – Marshalling/unmarshalling sont des concepts équivalents
15
16 public class Personne {
17     private String nom;
18     private int age;
19
20     public Personne(String nom, int age) {/**/}
21
22     @Override
23     public String toString(){/**/}
24 }
25
26 public abstract class DAO<T>{
27     protected Connection connect = /* ... */;
28
29     public abstract T create(T obj);
30     public abstract T find(String id);
31     public abstract T update(T obj);
32     public abstract void delete (T obj);
33 }
34
35 public class PersonneDAO extends DAO<Personne>{
36     @Override
37     public Personne create(Personne obj){
38         try{
39             PreparedStatement prepare = connect.prepareStatement(
40                 "INSERT INTO personnes (nom, age) VALUES (?, ?)");
41             prepare.setString(1, obj.getNom());
42             prepare.setLong(2, obj.getAge());
43             int result = prepare.executeUpdate();
44             assert result == 1;
45         }
46         catch(SQLException e){
47             e.printStackTrace();
48         }
49         return obj;
50     }
51
52     @Override
53     public Personne find(String id){
54         Personne p = new Personne();
55
56         try{
```

```
55         PreparedStatement prepare = connect.prepareStatement(  
56             "SELECT * FROM personnes WHERE nom = ? ");  
57         prepare.setString(1, id);  
58         int result = prepare.executeQuery();  
59         if(result.first()){  
60             p=new Personne(result.getString("nom"), result.getString("age"));  
61         }  
62     }  
63     catch(SQLException e){  
64         e.printStackTrace();  
65     }  
66     return obj;  
67 }  
68 }  
69  
70 public class DAO Factory {  
71     public static DAO<Personne> getPersonneDAO(){  
72         return new PersonneDAO();  
73     }  
74 }  
75  
76 public class Main {  
77     public static void main(String[] args){  
78         DAO<Personne> personneDao = DAOFactory.getPersonneDAO();  
79         System.out.println(personneDao.find("Dupond"));  
80     }  
81 }  
82  
83 //////////////////////////////////////:  
84  
85 Exemple classe immuable :  
86 public final class Personnel{  
87     private final String nom;  
88     private final String prenom;  
89 }  
90
```