

```
1  //////////////////////////////////////:
2  // JPA Persistence
3  //////////////////////////////////////
4
5  @Entity
6  public class Todo {
7      @Id
8      @GeneratedValue(strategy = GenerationType.IDENTITY)
9      private Long id;
10     private String summary;
11     private String description;
12
13     public String getSummary() {return summary;}
14     public void setSummary(String summary) {this.summary = summary;}
15     public String getDescription() {return description;}
16     public void setDescription(String description) {this.description = description;}
17
18     @Override
19     public String toString(){
20         return "Todo [summary=" + summary + ", description=" + description + "];"
21     }
22 }
23
24 @Entity
25 public class Task {
26     @Id
27     @GeneratedValue(strategy = GenerationType.IDENTITY)
28     private Long id;
29     private String title;
30
31     @OneToMany(mappedBy = "task")
32     private final List<Task> todos;
33 }
34
35 public class Main {
36     private static final String PERSISTENCE_UNIT_NAME = "todos";
37     private static EntityManagerFactory factory;
38
39     public static void main(){
40         factory = Persistence.createEntityManagerFactory(PERSISTENCE_UNIT_NAME);
41         EntityManager em = factory.createEntityManager();
42         em.getTransaction().begin();
43         Todo todo = new Todo();
44         todo.setSummary("This is a test");
45         todo.setDescription("This is a test");
46         em.persist(todo);
47         em.getTransaction().commit();
48         em.close();
49     }
50 }
51
52 // COURS
53 //////////////////////////////////////
54
55 - Les patterns sont des outils supplémentaires pour réaliser une bonne conception.
56 - Les principes concernant la conception des classes (SOLID), abordent la cohésion ↗
   des modules, traitent du couplage entre modules
```

```
57 --> Les principes et les patterns de conception sont principalement liés aux ↗
    dépendances.
58 - Les idiomes sont une construction utilisée de façon récurrente dans un langage ↗
    de programmation donné pour
59 réaliser une tâche « simple » ex :i++
60
61 ///////////////////////////////////////////////////
62
63 - ORM : Mapping objet-relationnel
64 - Hibernate est un ORM open source et est une implémentation de JPA
65 - JPA fournit un modèle de persistance pour les objetsJava (POJO) basé l'ORM (haut ↗
    niveau)
66 - JDBC fournissent un moyen pour interfacer un programme avec un SGBD (bas niveau)
67 - SGBD : systeme de gestion de base de donnees qui permet de manipuler des infos ↗
    dans une bdd
68
69 ///////////////////////////////////////////////////
70
71 Une unité de persistance (persistance unit) configure l'accès au SGBD pour un ↗
    ensemble d'entités.
72 Elle est décrite dans le fichier XML META-INF/persistence.xml
73
74 ///////////////////////////////////////////////////
75
76 - Marquer une entité : @Entity
77 - marquer une cle primaire : @Id
78 - avec un ORM, on gere les associations n-n avec @ManyToMany (mappedBy = "task")
79
```