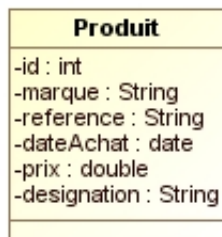


TP 3 : J2EE

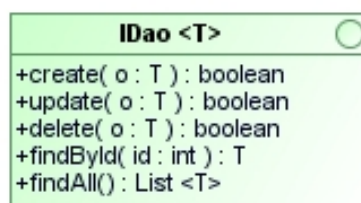
Framework de persistance : Hibernate

Exercice 1 :

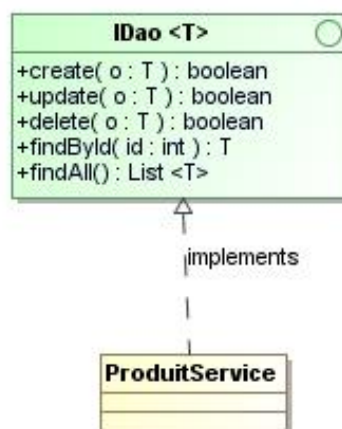
On souhaite créer une application pour la gestion des produits, pour ce faire nous proposons le diagramme de classe suivant :



1. Créer un projet JAVA sous Netbeans nommé « H1 », et rajouter les bibliothèques nécessaires (Hibernate-JPA et le Pilote de la base de données MySql).
2. Créer l'entité Produit dans le package « ma.projet.entity », rajouter les annotations (Entity, Id, GeneratedValue, Column, Table...).
3. Créer une base de données nommée « H1 » sous MySql.
4. Créer le fichier de configuration Hibernate « hibernate.cfg.xml » dans le package « ma.projet.config ».
5. Créer la classe HibernateUtil permettant de créer une instance de la classe « SessionFactory » dans le package « ma.projet.util ».
6. Créer l'interface IDao dans le packahe « ma.projet.dao » :



7. Créer la classe service « ProduitService » qui implémente l'interface IDao dans le package « ma.projet.service » et implémenter les différentes méthodes



8. Dans une classe de teste :
 - Créer cinq produits,
 - Afficher la liste des produits,

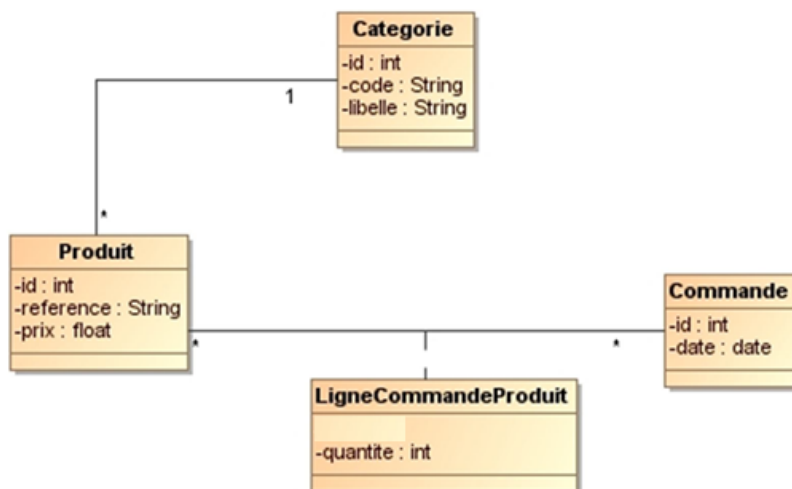
Pr.

- Afficher les informations du produit dont id = 2,
- Supprimer le produit dont id = 3,
- Modifier les informations du produit dont id = 1,
- Afficher la liste des produits dont le prix est supérieur à 100 DH,
- Afficher la liste des produits Commander entre deux dates lus au clavier.

Exercice 2 :

On souhaite développer une l'application de gestion de stock pour un magasin de vente des produits informatiques.

Les travaux de l'équipe chargée de l'analyse et de la conception ont abouti au diagramme de classe suivant :

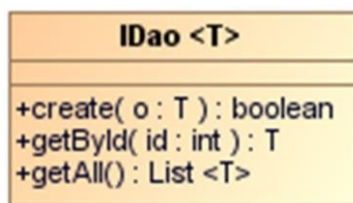


A. Couche persistance

1. Développer les classes entités dans le package « **ma.projet.classes** ».
2. Créer le fichier de configuration **hibernate.cfg.xml** dans le package « **ma.projet.config** ».
3. Créer la classe **HibernateUtil** dans le package « **ma.projet.util** ».

B. Couche service

Soit l'interface suivante :



1. Créer l'interface générique **IDao** dans le package « **ma.projet.dao** ».
2. Créer les classes services : **ProduitService**, **CategorieService**, **CammandeService** et **LigneCommandeService** qui implémentent l'interface **IDao** dans le package « **ma.projet.service** ».
3. Créer une méthode permettant d'afficher la liste des produits par catégorie dans la classe **ProduitService**.
4. Créer une méthode permettant d'afficher la liste des produits commandés entre deux dates.

5. Créer une méthode permettant d’afficher les produits commandés dans une commande donnée :

Exemple :

Commande : 4

Date : 14 Mars 2013

Liste des produits :

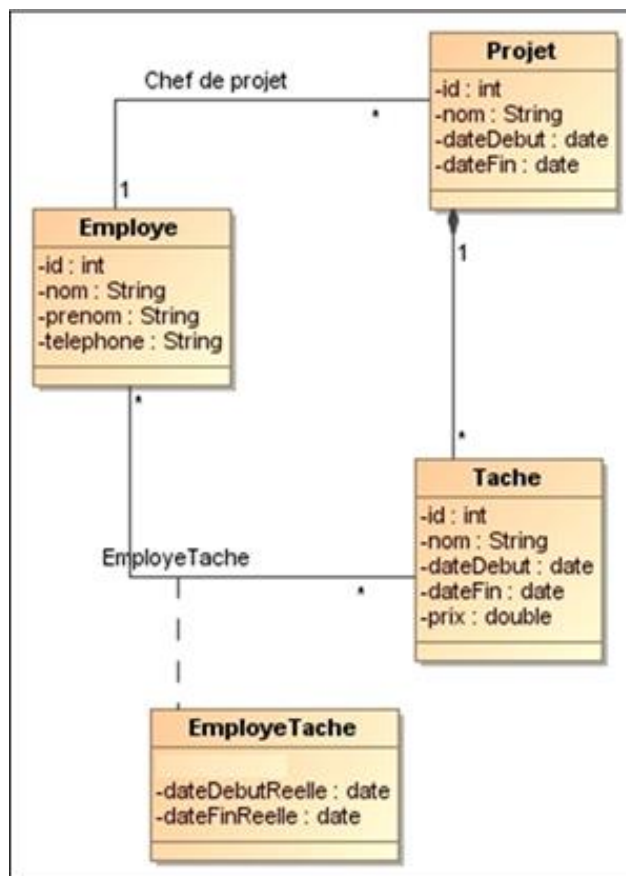
Référence	Prix	Quantité
ES12	120 DH	7
ZR85	100 DH	14
EE85	200 DH	5

6. Créer une méthode permettant d’afficher la liste des produits dont le prix est supérieur à 100 DH dans la classe **ProduitService** en utilisant une requête nommée.
7. Créer des programmes de tester les questions ci-dessus.

Exercice 3 :

Afin d’imputer le temps passé dans un projet sur son coût global, un bureau d’études se propose de mettre en place une application de gestion de projet.

Les travaux de l’équipe chargée de l’analyse et de la conception ont abouti au diagramme de classe suivant :

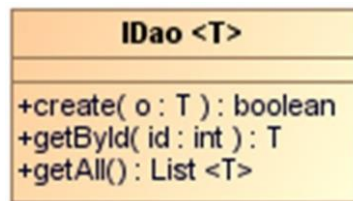


A. Couche persistance (3 pts)

1. Développer les classes entités dans le package « **ma.projet.classes** ».
2. Créer le fichier de configuration **hibernate.cfg.xml** dans le package « **ma.projet.config** ».
3. Créer la classe **HibernateUtil** dans le package « **ma.projet.util** ».

B. Couche service (17 pts)

Soit l'interface suivante :



1. Créer l'interface générique **IDao** dans le package « **ma.projet.dao** ».
2. Créer les classes services : **ProjetService**, **TacheService**, **EmployeService** et **EmployeTacheService** qui implémentent l'interface **IDao**.
3. Créer une méthode permettant d'afficher la liste tâches réalisées par un employé dans la classe **EmployeService**.
4. Créer une méthode permettant d'afficher la liste projets gérées par un employé dans la classe **EmployeService**.
5. Créer une méthode permettant d'afficher la liste des tâches planifiées pour projet dans la classe **ProjetService**.
6. Créer une méthode permettant d'afficher la liste des tâches réalisées dans un projet comme suite :

Exemple :

Projet : 4 **Nom : Gestion de stock** **Date début : 14 Janvier 2013**

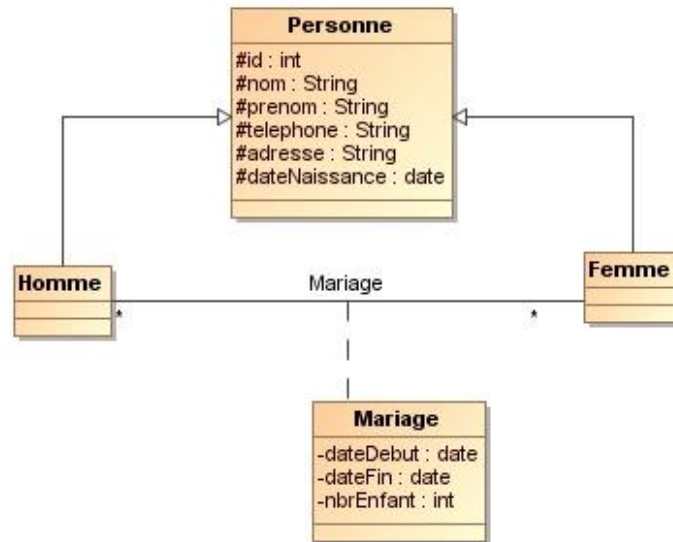
Liste des tachés:

Num	Nom	Date Début Réelle	Date Fin Réelle
12	Analyse	10/02/2013	20/02/2013
13	Conception	10/03/2013	15/03/2013
14	Développement	10/04/2013	25/04/2013

7. Créer une méthode permettant d'afficher la liste des tâches dont le prix supérieur à 1000 DH dans la classe **TacheService** en utilisant une requête nommée.
8. Créer une méthode permettant d'afficher la liste des tâches réalisées entre deux dates dans la classe.
9. Créer des programmes permettant de tester les questions ci-dessus.

Travail à rendre :

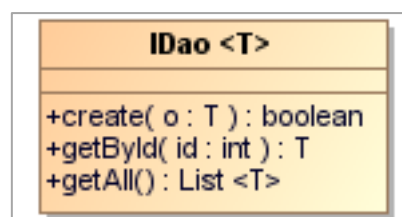
On souhaite développer une application de gestion de l'état civil des citoyens de notre province. Les travaux de l'équipe chargée de l'analyse et de la conception ont abouti au diagramme de classe suivant :

**A. Couche de persistance (6 pts)**

1. Développer les classes entités dans le package « **ma.projet.beans** ». Mentionner toutes les annotations vues en classe. **(3 pts)**
2. Créer le fichier de configuration **hibernate.cfg.xml** dans le package **ma.projet.config**. **(1 pt)**
3. Créer la classe **HibernateUtil** dans le package **ma.projet.util**. **(1 pt)**
4. Générer la base de données sous MySQL. **(1 pt)**

B. Couche service (14 pts)

Soit l'interface suivante :



1. Créer l'interface générique **IDao** dans le package « **ma.projet.dao** ». **(1 pt)**
2. Créer les classes services : **HommeService**, **FemmeService** et **MariageService** qui implémentent l'interface **IDao**. **(1 pt)**
3. Créer une méthode permettant d'afficher les épouses d'un homme passé en paramètre entre deux dates dans la classe **HommeService**. **(1 pt)**
4. Créer une requête native nommée permettant de renvoyer le nombre d'enfants d'une femme donnée entre deux dates. **(1 pt)**
5. Créer une méthode dans la classe **FemmeService** permettant de faire appel à la requête de la question 4. **(1 pt)**
6. Créer une requête nommée permettant de renvoyer les femmes mariées deux fois ou plus. **(1 pt)**

7. Créer une méthode dans la classe **FemmeService** permettant de faire appel à la requête de la question 6. **(1 pt)**
8. Créer une méthode permettant de renvoyer le nombre des hommes qui sont mariés par 4 femmes entre deux dates en utilisant l'API CRITERIA. **(1 pt)**
9. Créer une méthode permettant de renvoyer les mariages d'un homme donné en paramètre comme suit : **(2 pts)**

Nom : SAFI SAID

Mariages En Cours :

1. **Femme** : SALIMA RAMI **Date Début** : 03/09/1990 **Nbr Enfants** : 4
2. **Femme** : AMAL ALI **Date Début** : 03/09/1995 **Nbr Enfants** : 2
3. **Femme** : WAFA ALAOUI **Date Début** : 04/11/2000 **Nbr Enfants** : 3

Mariages échoués :

1. **Femme** : KARIMA ALAMI **Date Début** : 03/09/1989 **Date Fin** : 03/09/1990 **Nbr Enfants** : 0

10. Créer un programme de teste : **(4 pts)**

- Créer 10 femmes et 5 hommes,
- Afficher la liste des femmes,
- Afficher la femme la plus âgée,
- Afficher les épouses d'un homme passé en paramètre,
- Afficher le nombre d'enfants d'une femme passée en paramètre entre deux dates,
- Afficher la liste des femmes mariées deux fois ou plus,
- Afficher qui sont mariés par quatre femmes entre deux dates,
- Afficher les mariages d'un homme passé en paramètre.