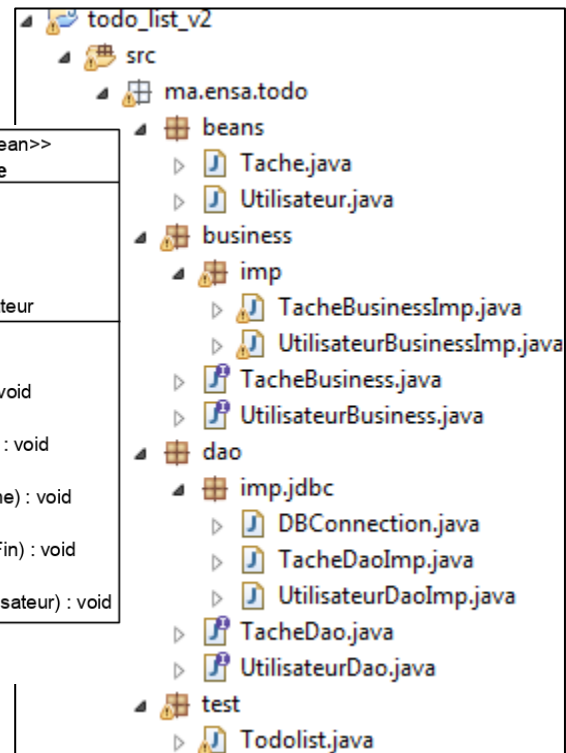
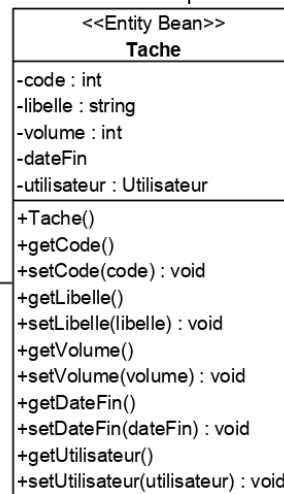
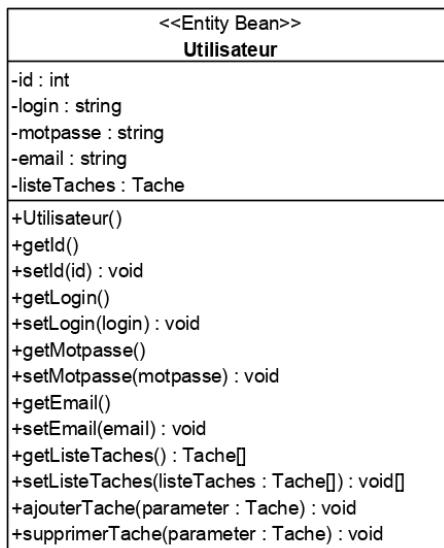


Atelier 3 : « Todo list v2.0 »

Nous allons refaire l'application en appliquant cette fois-ci le désigne pattern DAO et en garantissant un mapping entre les objets beans (Tache et Utilisateur) et les données des tables de la BD.

Travail à faire :

1. Créer un nouveau projet Java nommé « **todo_list_v2** »
2. Créer les packages nécessaires en respectant le design pattern DAO (couche test, couche business, Couche beans, couche dao, ...)
3. Créer les Javabeans (**Tache et Utilisateur**) qui font le mapping aux tables relationnelles (voir ci-dessous).
4. Créer les interfaces **EtudiantDao** et **TacheDao** et leurs implémentations JDBC (**TacheDaoImp** et **UtilisateurDaoImp**).
5. Créer les interfaces **EtudiantBusiness** et **TacheBusiness** et leurs implémentations (**TacheBusinessImpl** et **UtilisateurBusinessImpl**).
6. Créer la classe **TodoList** qui permet de faire les injections nécessaires et lancer le programme principal.



```

public interface TacheBusiness {
    public void afficherTaches(Utilisateur utilisateur);
    public void ajouterTache(Utilisateur utilisateur);
    public void supprimerTache(Utilisateur utilisateur);
    public void modifierTache(Utilisateur utilisateur);
}

public interface UtilisateurBusiness {
    public void ajouterUtilisateur();
    public Utilisateur connecterUtilisateur();
}

public interface TacheDao {
    public List<Tache> findByUtilisateur(Utilisateur utilisateur);

    public Tache createTache(String libelle, int volume, Date dateFin, int id);
    public void deleteTache(int code);
    public void updateTache(int code, String libelle, int volume, Date dateFin, int id);
}

public interface UtilisateurDao {
    public Utilisateur findByLogin(String login, String motpasse);
    public void createUtilisateur(String login, String motpasse, String email);
}
    
```