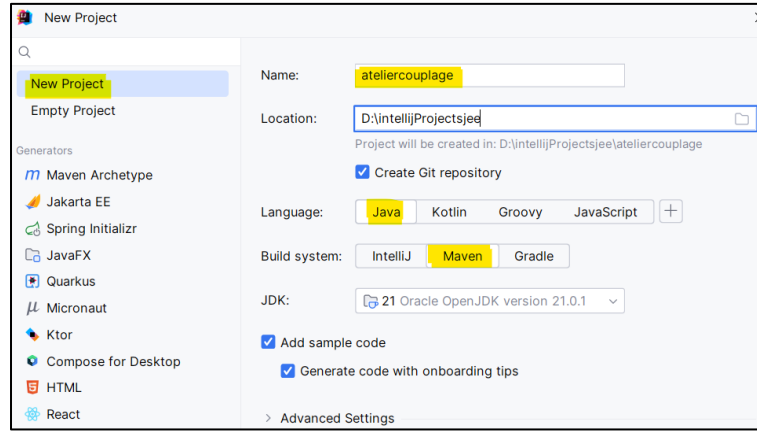


TP 1 - Atelier Couplage Fort Vs Couplage Faible : <https://github.com/badrhr/>

I. Partie 1 : Instanciation statique

1. Créer un nouveau projet (New Project) :



2. Créer les trois packages : dao, metier, et presentation
3. Dans le package dao créer les deux classes : Etudiant et EtudiantDAO.
 - 3.1. Chaque Etudiant est identifié par son id, nom, prenom, et email.
 - 3.2. La classe EtudiantDAO contient :
 - 3.2.1. La liste *etudiants* qui contient tous les étudiants.
 - 3.2.2. Une méthode pour ajouter un nouveau Etudiant :
 - 3.2.2.1. `public Etudiant addEtudiant(Etudiant etudiant) {...}`
 - 3.2.3. Une méthode pour récupérer tous les étudiants :
 - 3.2.3.1. `public List<Etudiant> getAllEtudiants() {...}`
4. Dans le package metier créer la classe EtudiantManager avec les attributs et méthodes suivantes :
 - 4.1. L'attribut EtudiantDAO etudiantDAO.
 - 4.1.1. `public EtudiantDAO etudiantDAO ;`

La classe EtudiantManager utilise les services assurés par la couche dao à travers la classe EtudiantDAO, donc on doit ajouter un attribut de type EtudiantDAO.
 - 4.2. Ajouter la méthode addEtudiant :
 - 4.2.1. `public Etudiant addEtudiant(Etudiant etudiant) {...}`

Avant d'ajouter un nouvel étudiant, on doit vérifier si son email est déjà utilisé par un ancien étudiant ou non.
 - 4.3. Ajouter la méthode getAllEtudiants :
 - 4.3.1. `public List<Etudiant> getAllEtudiants() {...}`
5. Dans le package presentation créer la classe Presentation qui contient la méthode main:
 - 5.1. Créer et ajouter 4 étudiants dans la liste *etudiants*, et tester l'application.

II. Partie 2 : extension de l'application

Après quelques mois de bon fonctionnement de votre application, vous avez décidé de changer l'utilisation de la liste comme moyen de sauvegarde avec un dictionnaire [Map<String, Etudiant> etudiantsMap].

6. Créer une classe EtudiantDAODictionary dans le package dao similaire à la classe EtudiantDAO (question 3.2).
 - 6.1. Au lieu d'enregistrer les étudiants dans la liste, on doit les ajouter au dictionnaire etudiantsMap comme suit : etudiantsMap[id] = etudiant.
7. Citer les différents changements nécessaires pour intégrer la nouvelle classe [EtudiantDAODictionary] au lieu d'ancienne classe [EtudiantDAO].

III. Partie 3 : Instanciation dynamique

8. Proposer une solution solide qui va assurer le bon fonctionnement de l'application dans ses deux versions.
9. Étendre l'application en ajoutant le service de mise à jour des informations des étudiants tout en gardant le même id.
10. Citer les changements nécessaires en justifiant chaque modification ou extension.