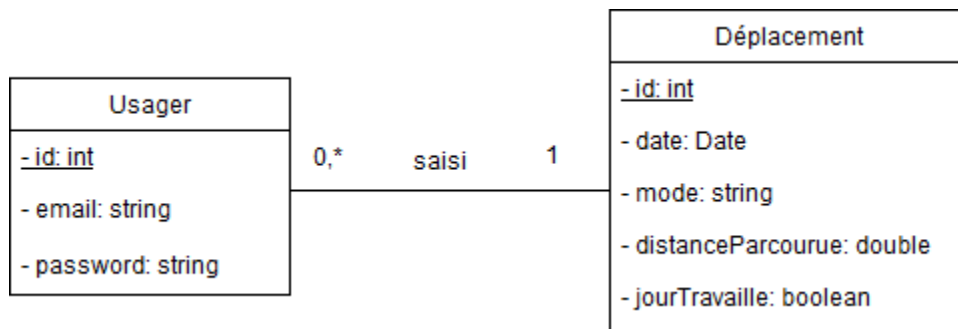


Programmation orientée objet

Question 1 :



Nous avons décidé d'ajouter un identifiant unique de type **int** pour la table **Usager** afin de faciliter son utilisation et améliorer les performances.

Note : Il aurait également été possible de séparer le mode et d'en faire une entité à part entière.

Question 2

Lors de la création de **Test1**, nous avons dans un premier temps créé des usagers :

```

Usager usager1 = new Usager("thibaut.dufour@outlook.com", "thibaut");
Usager usager2 = new Usager("adrien.philippaux@gmail.com", "adrien");
Usager usager3 = new Usager("adrien.philippaux@gmail.com", "toto");
Usager usager4 = new Usager("toto@gmail.com", "toto");
    
```

L'utilisateur 3 a volontairement la même adresse email que le 2 afin de tester le fait que l'email doit être unique.

Puis nous avons créé des déplacements :

```
Deplacement dep1 = new Deplacement(new Date(), "VOI", 54.2, true);
Deplacement dep2 = new Deplacement(new Date(), "TRA", 474.79, false);
Deplacement dep3 = new Deplacement(new Date(), "VEL", 4.0, true);
Deplacement dep4 = new Deplacement(new Date(), "NUL", 0.0, true);
```

Pour chaque usager, nous allons ensuite lui affecter un ou plusieurs déplacement(s).

```
if(usager1.addDeplacement(dep1))
    System.out.println("Déplacement 1 ajouté à l'utilisateur 1");
else
    System.out.println("Problème lors de l'ajout du déplacement 1 à l'utilisateur 1");
```

Lors de l'ajout des usagers à la classe DAO, nous avons également ajouté les déplacements, ce qui nous a posé des problèmes.

Enfin, nous effectuons un affichage sur les différents éléments (usagers + déplacements) puis effectuons des modifications avant de les afficher à nouveau. Voici les éléments que nous avons effectués :

- Affichage de tous les usagers
- Suppression de l'utilisateur 4
- Update de l'utilisateur 1
- Affichage de tous les usagers
- Affichage de tous les déplacements
- Affichage des déplacements de l'utilisateur 1
- Affichage de tous les déplacements
- Suppression du déplacement 4
- Update du déplacement 3
- Affichage de tous les déplacements

Question 3

Afin d'utiliser JDBC, nous avons créé les classes nécessaires, nous avons modifié la **DAO Factory** afin d'ajouter une classe : **DAOFactoryJDBC**.

Nous avons également créé les classes JDBCUsagerDao et JDBCDeplacementDAO qui contiennent toutes les requêtes nécessaires.

Enfin, nous avons testé les différentes classes et requête à l'aide de la classe **Test2**.