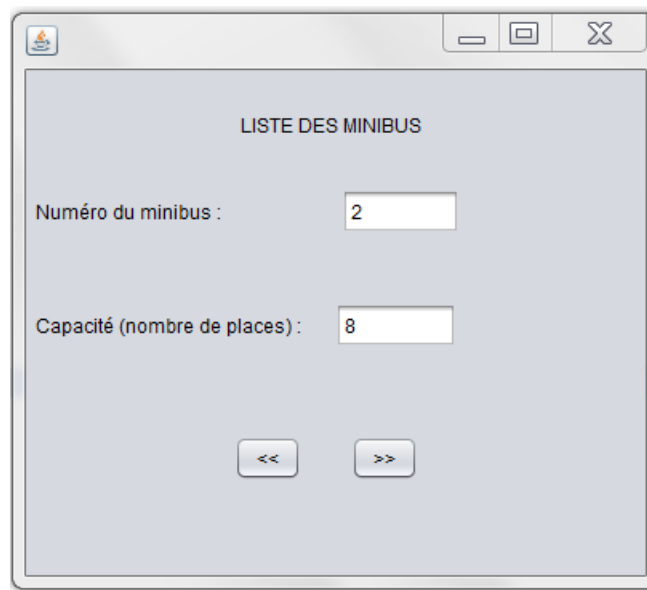


Bases de données
Exercice série 4
Mapping Relationnel-Objets et Design Pattern DAO

Soit l'interface suivante qui ne gère que les Minibus :

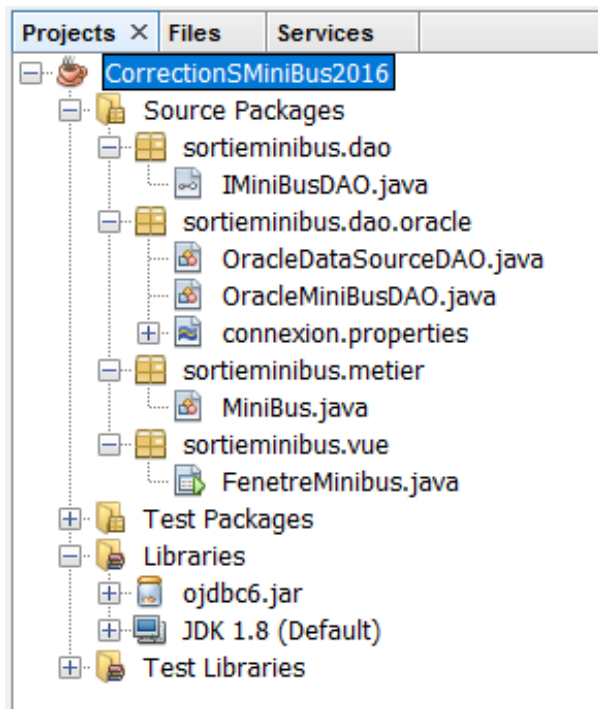


Travail à réaliser :

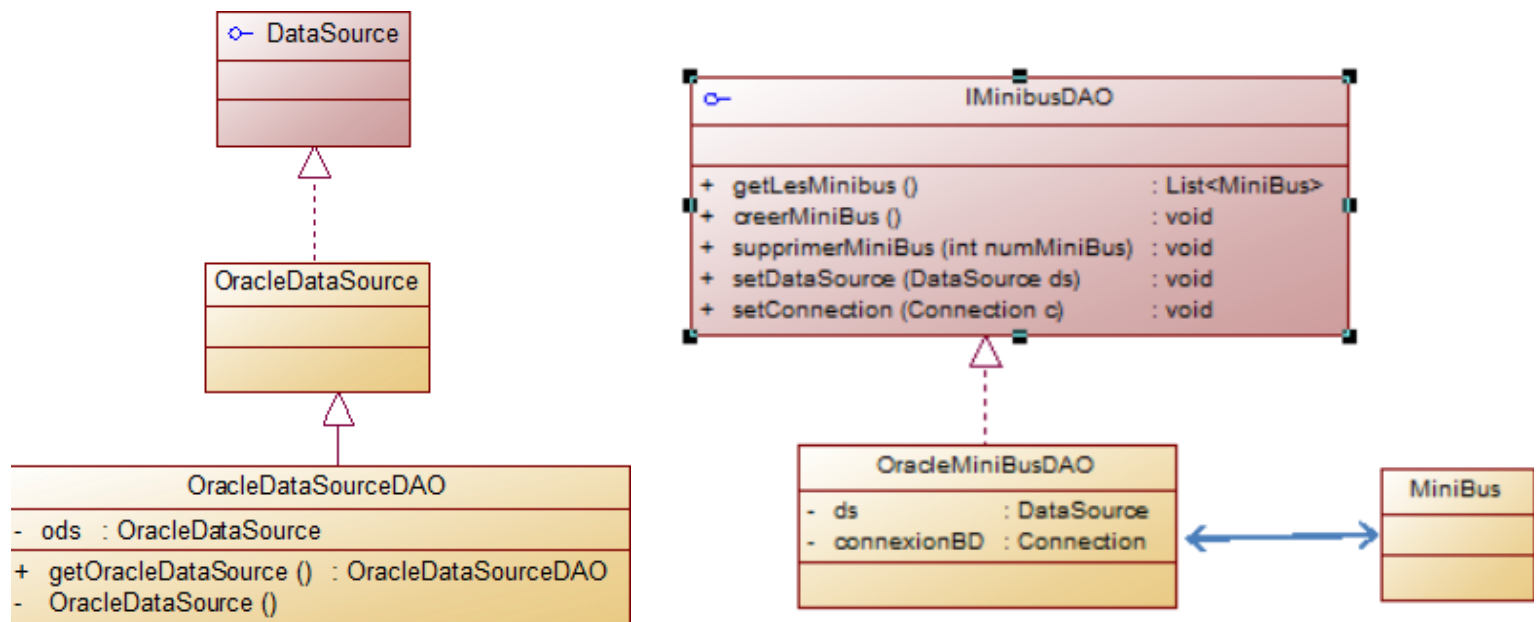
Les données sont stockées dans une base de données. Elles seront accédées via une connexion JDBC.

Pour réaliser cette application vous veillerez à **séparer nettement les logiques** (couche métier, couche présentation, couche données) par l'utilisation de packages (voir ci-dessous l'architecture) et à **mettre en œuvre le pattern DAO** pour gérer le mapping Relationnel-objet.

- Architecture du projet :

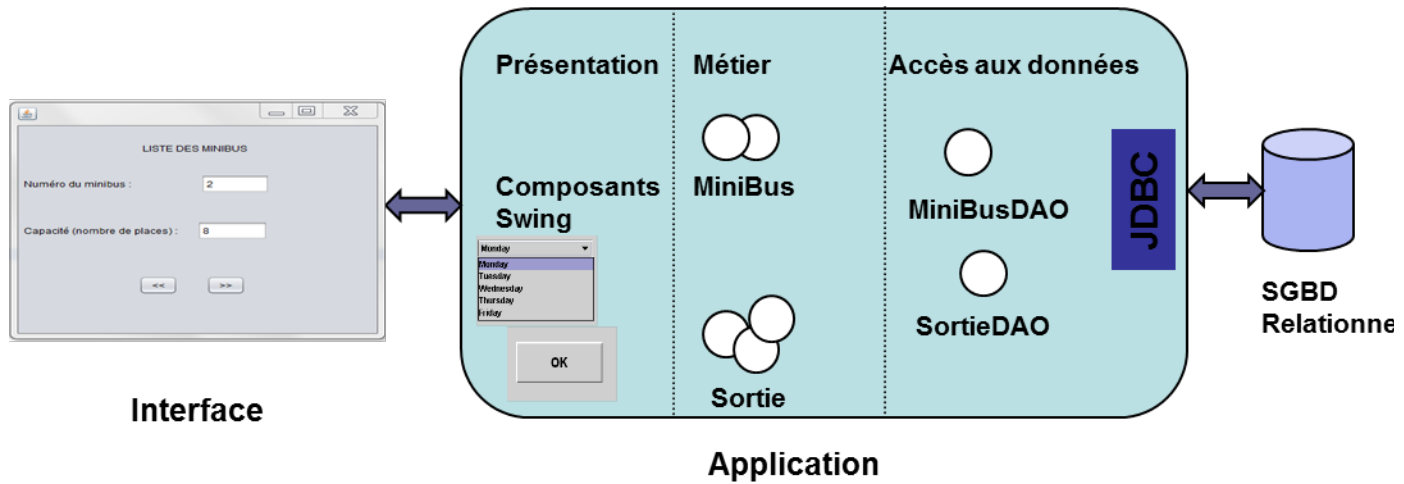


- Hiérarchie pour la persistance :



Compléments

- Composants de l'application :



- ✓ L'interface des objets DAO est indépendante du support de persistance.
- ✓ Le reste de l'application utilise les DAO uniquement au travers de leur interface abstraite. Par exemple si on change d'implémentation de DAO pour passer d'une base Oracle à une base MySql le reste de l'application demeure inchangé.
- ✓ Chaque classe d'objet métier a son propre type de DAO (MiniBusDAO, SortieDAO, etc.). Le même objet DAO peut être utilisé pour les objets d'une même classe d'objets métier.

Objet DAO pour les MiniBus :

- Définir une interface **IMiniBusDAO** → vous pourrez avoir ensuite plusieurs implémentations : **OracleMiniBusDAO** pour le SGBD Oracle, **MySQLMiniBusDAO** pour SGBD MySQL,...
- Le code qui utilise les objets DAO utilise l'interface → il ne dépend pas des implémentations.