

Introduction – Application N-tier

Les interfaces : pierre angulaire de la modularité

Exemple avec la couche DAO (TD2/TD3)

Une **interface** décrit un ensemble de méthodes qu'une classe doit posséder, au minimum.

```
public interface IDaoConcours{...}
public class DaoConcoursMySQL implements IDaoConcours{...}
public class DaoConcoursJSON implements IDaoConcours{...}
```

Dès lors, une **instance** d'une classe qui implémente l'interface peut être désignée par une référence **IDaoConcours**. *On n'a pas besoin de connaître le type réel de l'objet pour invoquer les services de l'interface.* Le type réel n'est mentionné qu'à l'instanciation.

```
IDaoConcours dao=null; // référence sur la couche DAO
// Instanciation: au choix, implémentation 1) ou 2)
// 1) dao=new DaoConcoursJSON("dataJ3.json");
// 2) dao=new DaoConcoursMySQL();
...
```

```
// Dans le reste du logiciel, la couche DAO est désignée par dao, peu importe l'implémentation
Candidat[] tab=dao.getCandidats(2); // DaoConcoursJSON ? DaoConcoursMySQL ? Peu importe
```

Application N-tier : on voit l'application comme un empilement de couches, chacune décrite par une interface. Dès lors, chaque couche est un module facilement remplaçable. C'est aussi une sorte de mini liste chaînée où chaque noeud est une couche du logiciel (cf dessin ci-dessous).

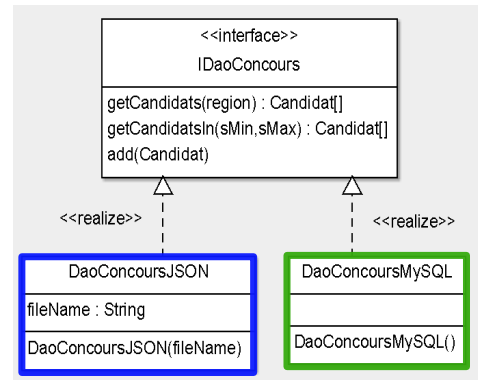
Exemple d'architecture classique : couche UI (User Interface) / couche Métier / couche DAO

L'utilisateur interagit avec l'implémentation de la couche UI (Console, GUI, service Web)

La couche UI s'adresse à la couche Métier (logique métier)

La couche Métier lit/écrit des données en s'adressant à la couche DAO

Bénéfice : facilité de remplacement de la couche DAO (changement de SGBD), ou de la couche UI



dao(référence à un objet)

