LE DEVELOPPEMENT EN COUCHES AVEC JAVA SE

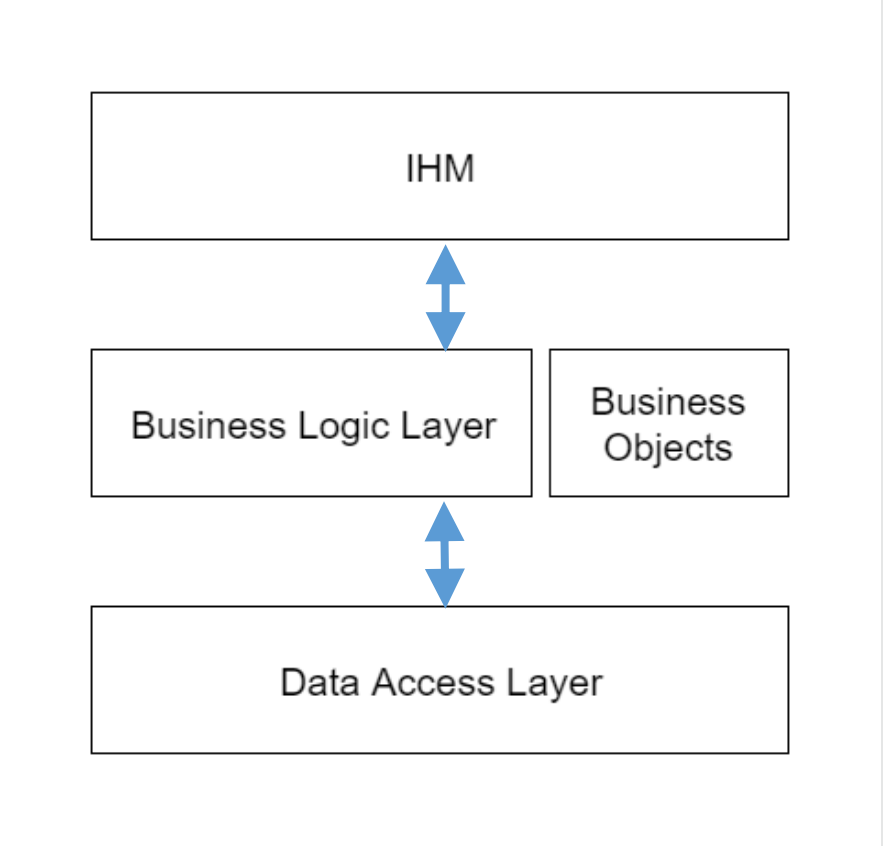
Module 3 – Le développement de la couche Business Objects (BO)

Objectifs

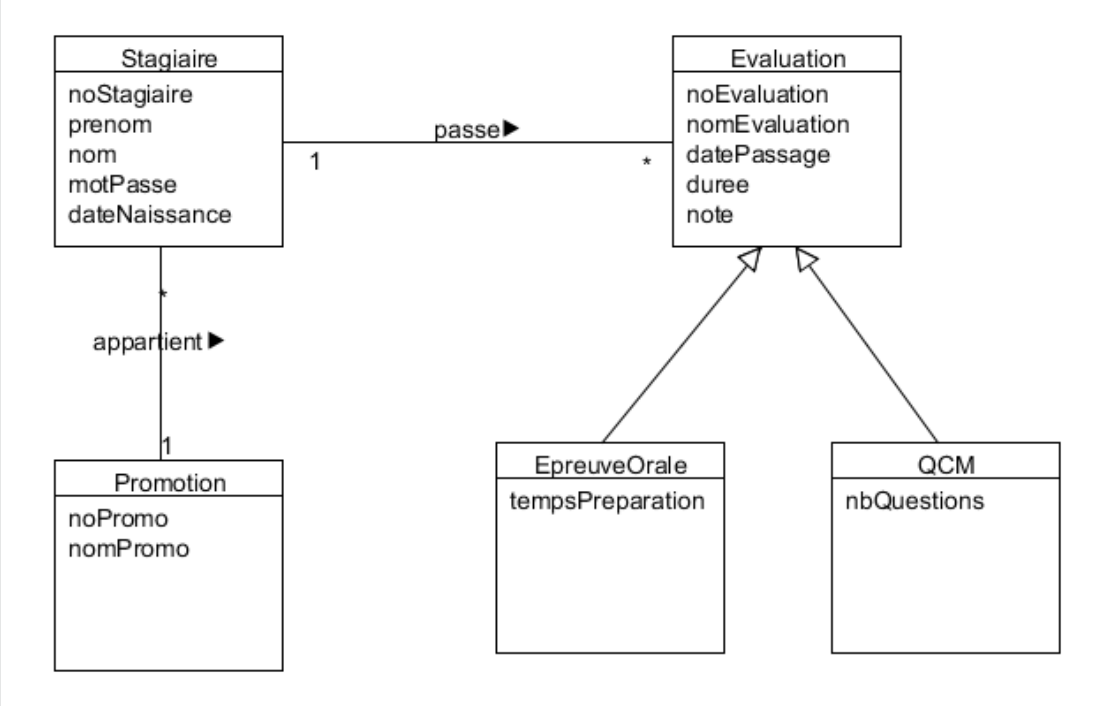
- Situer la couche Business Objects (BO)
- Connaître les responsabilités de la couche BO
- Savoir implémenter la couche BO
  - Notion de POJO
  - Utiliser les constructeurs
  - Utiliser l'encapsulation
  - Utiliser les associations entre classes
  - Utiliser l'héritage

Le développement de la couche BO

Situer la couche BO



Construire la couche Business Objects



Modèle de données métier Evaluation stagiaire

Utiliser des classes simples (POJOs)

- Ne pas dépendre des autres couches

Utiliser les concepts POO

- Constructeurs
- Encapsulation
- Associations entre classes
  - 1:1
  - 1:n
  - n:m
- Héritage de classe
- Implémentation d'interface

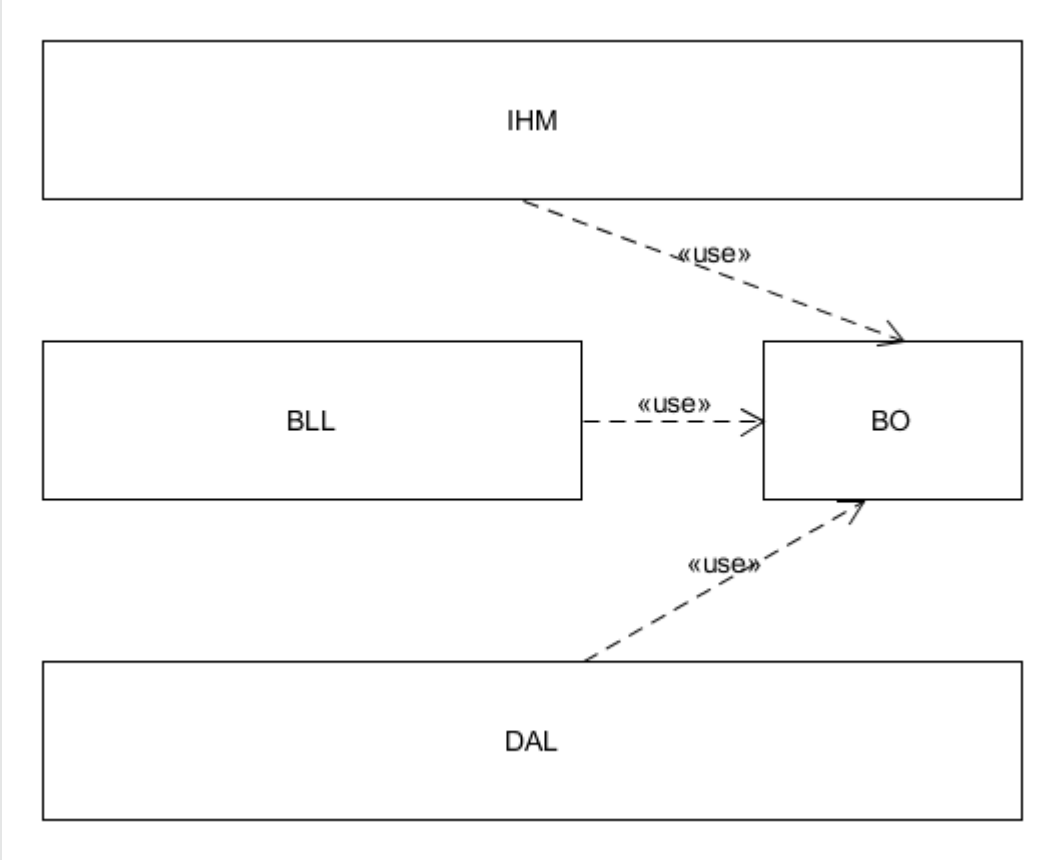
POJO : Plain Old Java Object

Un POJO est une classe java respectant les principes POO (encapsulation notamment), mais sans dépendance avec des frameworks ou librairies tierces qui en réduirait sa ré-utilisabilité.

Le développement de la couche BO

Les BO sont des POJOS

- La couche BO est utilisée par toutes les autres couches
- La couche BO ne doit pas dépendre des autres couches sinon : dépendances transitives ou cycliques
- La couche BO ne doit être liée à aucune technologie (technologie d'ihm web ou autre, technologie de bases de données)



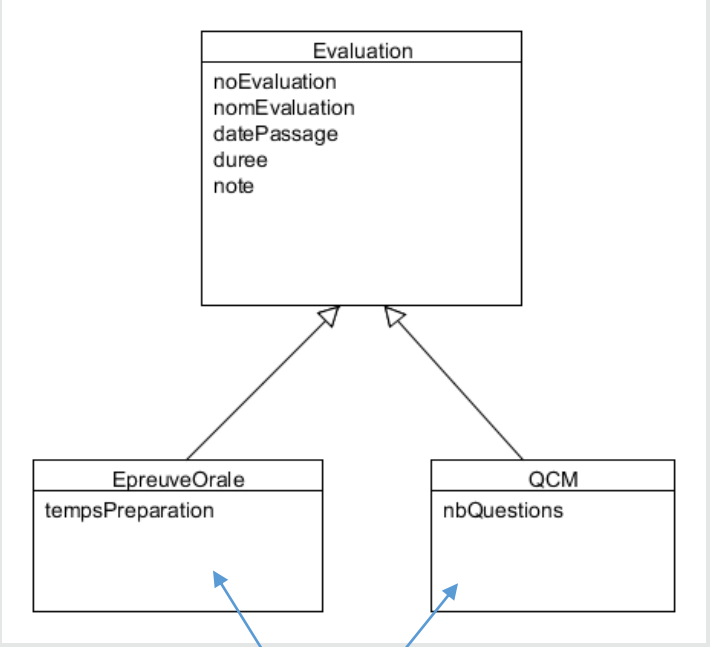
Utiliser des classes simples

- Respecter le principe d'encapsulation
  - Attributs privés
  - Getters et Setters publics
- Utiliser les constructeur(s) publiques nécessaires
- Les BO ne contiennent pas la logique de l'application

Le développement de la couche BO

Utiliser l'héritage

Généralisation

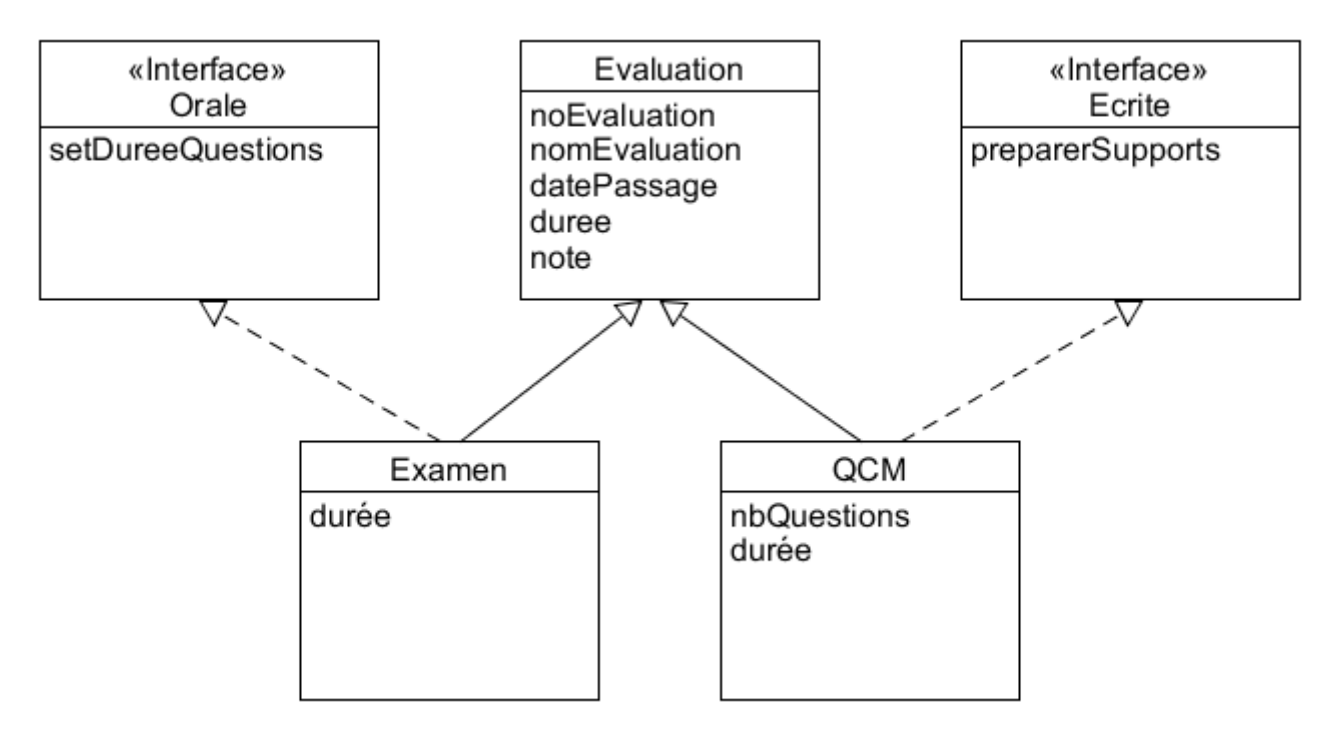


Spécialisation

Catégorisation

LE DEVELOPPEMENT DE LA COUCHE BO

Utiliser les interfaces

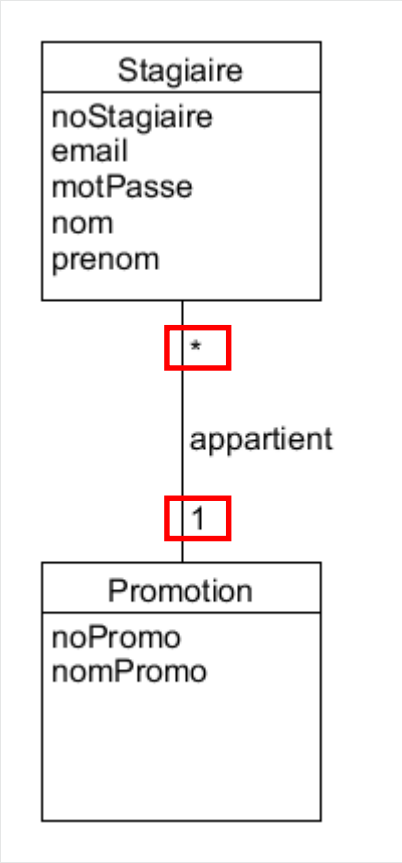



```java
package fr.eni.evaluations.bo;

public interface Ecrite {

    public abstract void setDureeQuestions(int duree);

    public abstract int getDureeQuestions();

}
```

```java
package fr.eni.evaluations.bo;

public class Examen extends Evaluation implements Ecrite{

    private int dureeQuestions;

    @Override
    public void setDureeQuestions(int duree) {
        this.dureeQuestions = duree;
    }

    @Override
    public int getDureeQuestions() {
        return dureeQuestions;
    }

}
```

Le développement de la couche BO

Relier les entités

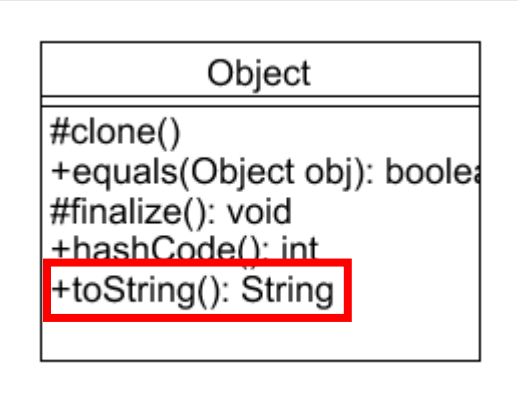



```java
public class Stagiaire {

    private long noStagiaire;
    private String prenom,  nom, email, motDePasse;
    private Promotion promotion;
```

```java
public class Promotion {

    List<Stagiaire> stagiaires;
```

Le développement de la couche BO

Implémenter toString()



- Donner l'état de l'instance.
- Donne de l'information en phase de debug
- Exemple d'implémentation :

```java
@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append("Login [login=");
    sb.append(login);
    sb.append(", password=");
    sb.append(password);
    sb.append("]");
    return   sb.toString() ;
}
```

Le développement de la couche BO

Gestion d'une papeterie - partie 1

TP