# Rapport  IOC // Couplage Faible : TP compte rendu

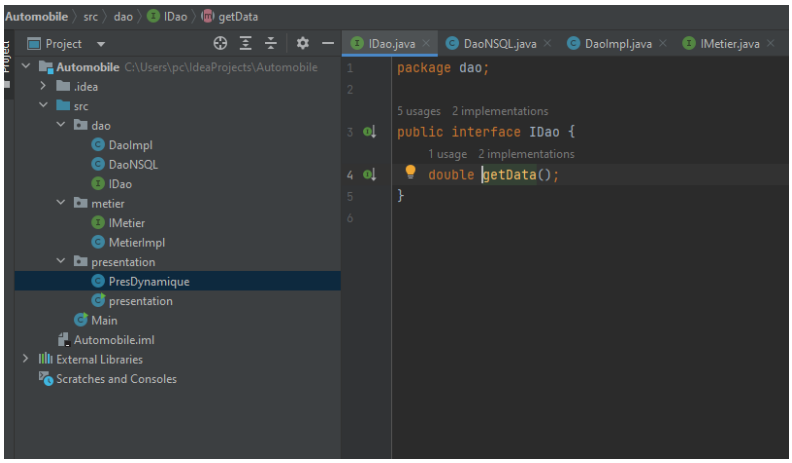
Réalisé par : Salma Nacimi et Oussama Zouitni

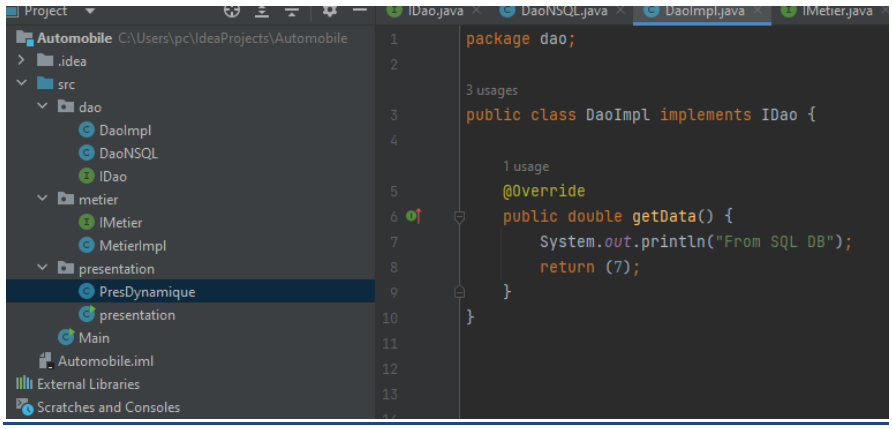
Il existe 4 méthodes pour faire l'injection des dépendances :

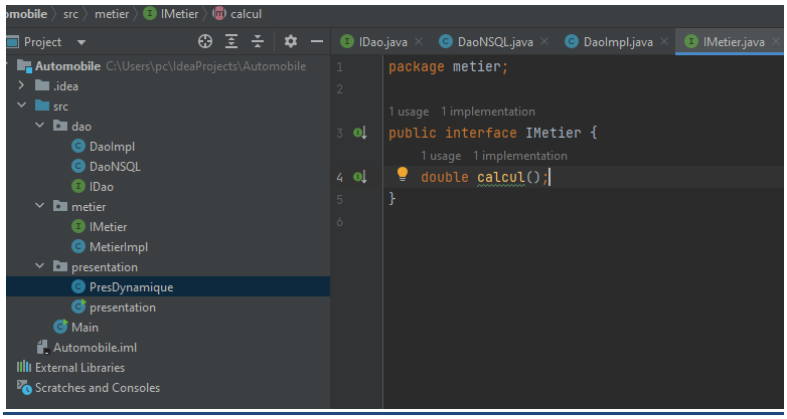
**Sans utiliser Spring :**

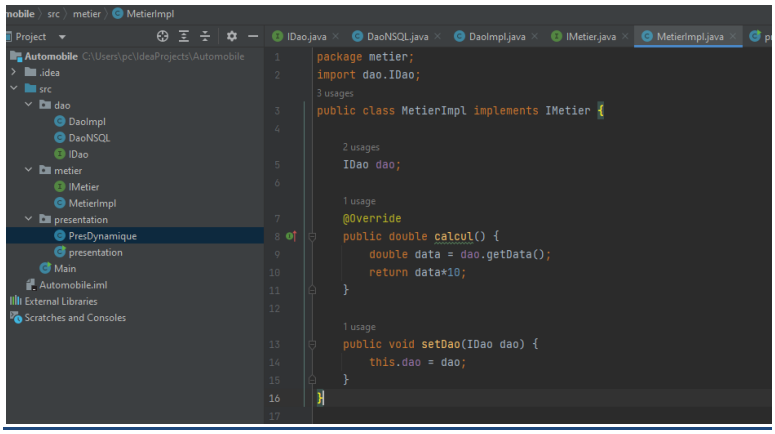
## 1ère méthode :  Par instanciation statique

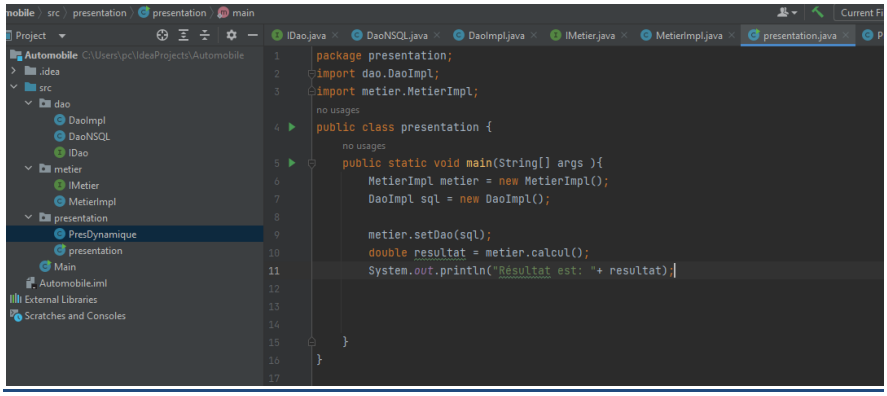
Instancier des classes connues avec new




Automobile › src › metier › IMetier › calcul
Project
Automobile C:\Users\pc\IdeaProjects\Automobile
.idea
src
dao
DaoImpl
DaoNSQL
IDao
metier
IMetier
MetierImpl
presentation
PresDynamique
presentation
Main
Automobile.iml
External Libraries
Scratches and Consoles
IDao.java
DaoNSQL.java
DaoImpl.java
IMetier.java
package metier;
1 usage 1 implementation
public interface IMetier {
1 usage 1 implementation
double calcul();
}


Automobile › src › metier › MetierImpl
IDao.java
DaoNSQL.java
DaoImpl.java
IMetier.java
MetierImpl.java
package metier;
import dao.IDao;
3 usages
public class MetierImpl implements IMetier {
2 usages
IDao dao;
1 usage
@Override
public double calcul() {
double data = dao.getData();
return data*10;
}
1 usage
public void setDao(IDao dao) {
this.dao = dao;
}
}


Automobile › src › presentation › presentation › main
IDao.java
DaoNSQL.java
DaoImpl.java
IMetier.java
MetierImpl.java
presentation.java
package presentation;
import dao.DaoImpl;
import metier.MetierImpl;
no usages
public class presentation {
no usages
public static void main(String[] args ){
MetierImpl metier = new MetierImpl();
DaoImpl sql = new DaoImpl();
metier.setDao(sql);
double resultat = metier.calcul();
System.out.println("Résultat est: "+ resultat);
}
}

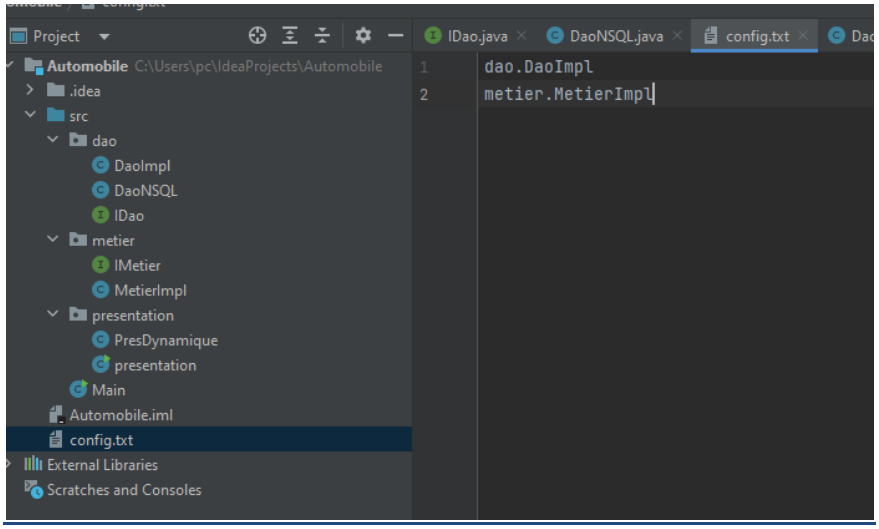
## 2ème méthode : Par instanciation dynamique

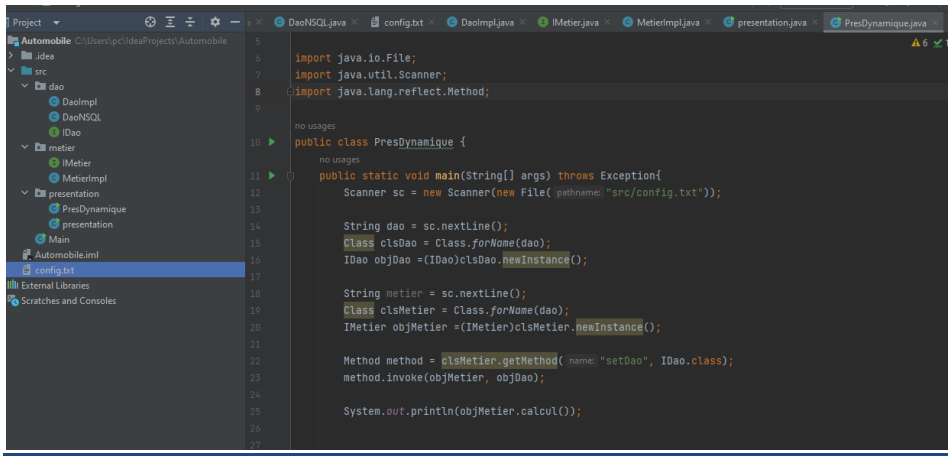
Créer un fichier de configuration qu'on appellera config.txt

1 .Déclaration des noms de classes qu'on utilisera dans l'application

2.On va lire le fichier texte avec la classe scanner(lire a partir d'un fichier)

3.Instancier un objet de type classe c'est l'instanciation dynamique



**Avec Spring :**

## 1ère méthode : En utilisant Beans (Fichier XML)

Télecharger les librairies : spring core, spring context et spring beans avec Java Build Path et les injecter dans le projet dans le nouveau dossier qu'on a créer nommé libs dans pom.xml

Après, on crée un fichier config.xml dans le dossier src et on demarre Spring . Pour démarrer Spring, on doit instancier un objet de type ApplicationContext.

## 2ème méthode : En utilisant les annotations

L'annotation @Autowired permet d'activer l'injection automatique de dépendance. Cette annotation peut être placée sur un constructeur, une méthode setter ou directement sur un attribut (même privé). Le Spring Framework va chercher le *bean* du contexte d'application dont le type est applicable à chaque paramètre du constructeur, aux paramètres de la méthode ou à l'attribut.

Le *framework* va scruter toutes les classes et créer un *bean* dans le conteneur IoC pour celles qui sont identifiées comme des composants. Une classe désigne un composant si elle possède l'annotation @Component ou une annotation de stéréotype de composant.

```
@Component
public class SQL implements IDao {
```

```
@Component
public class MetierV3 implements IMetier {

    @Autowired
    IDao dao;

    @Override
    public double calcul() {
        double d = dao.getData();
        return d*2021;
    }

    public void setDao(IDao d) {
        this.dao = d;
    }
}
```

```java
public class PresentationSpringV2 {

    public static void main(String[] args) {


        ApplicationContext context = new AnnotationConfigApplicationContext("dao","metier");
        IMetier metier = context.getBean(IMetier.class);
        System.out.println("R: " + metier.calcul());

    }

}
```