



---

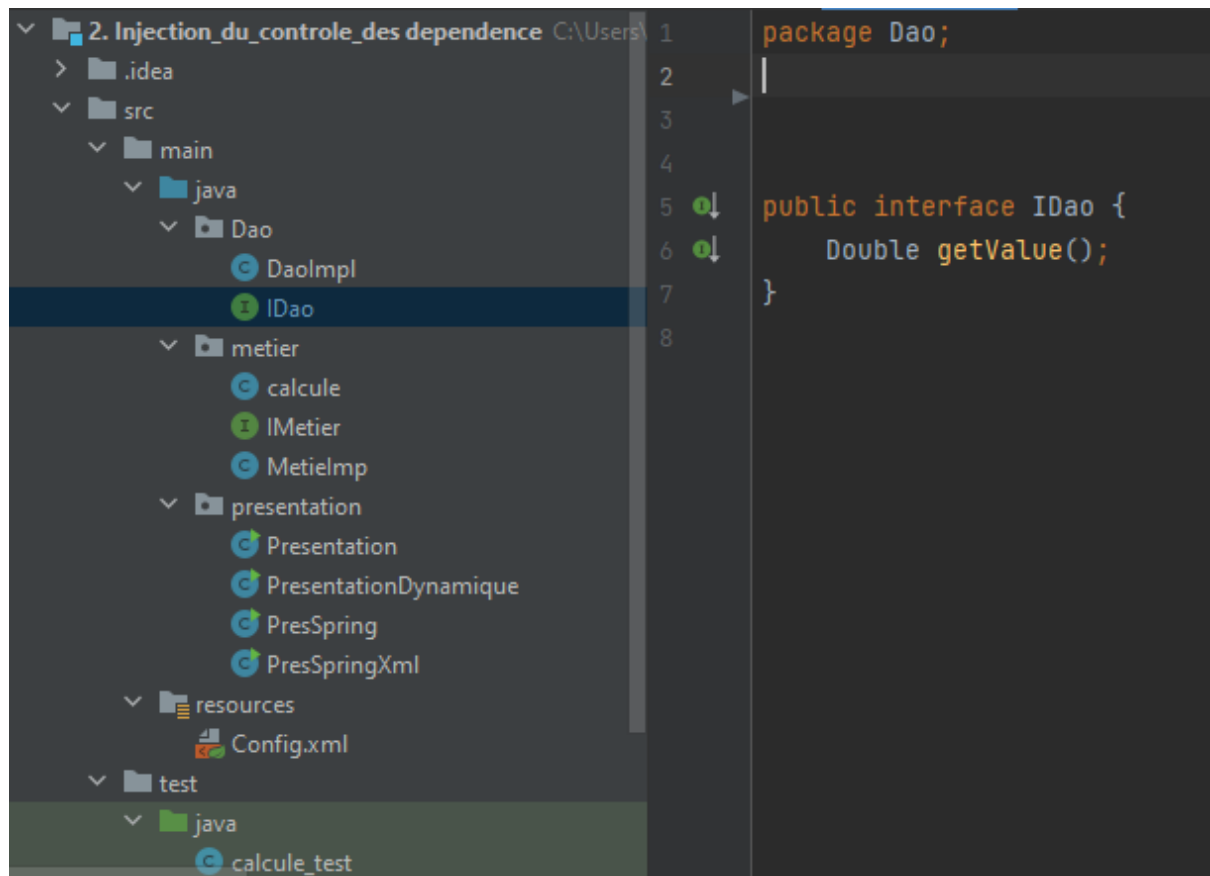
## **Compte rendu du TP**

"L'inversion du contrôlés et l'injection des dépendances"

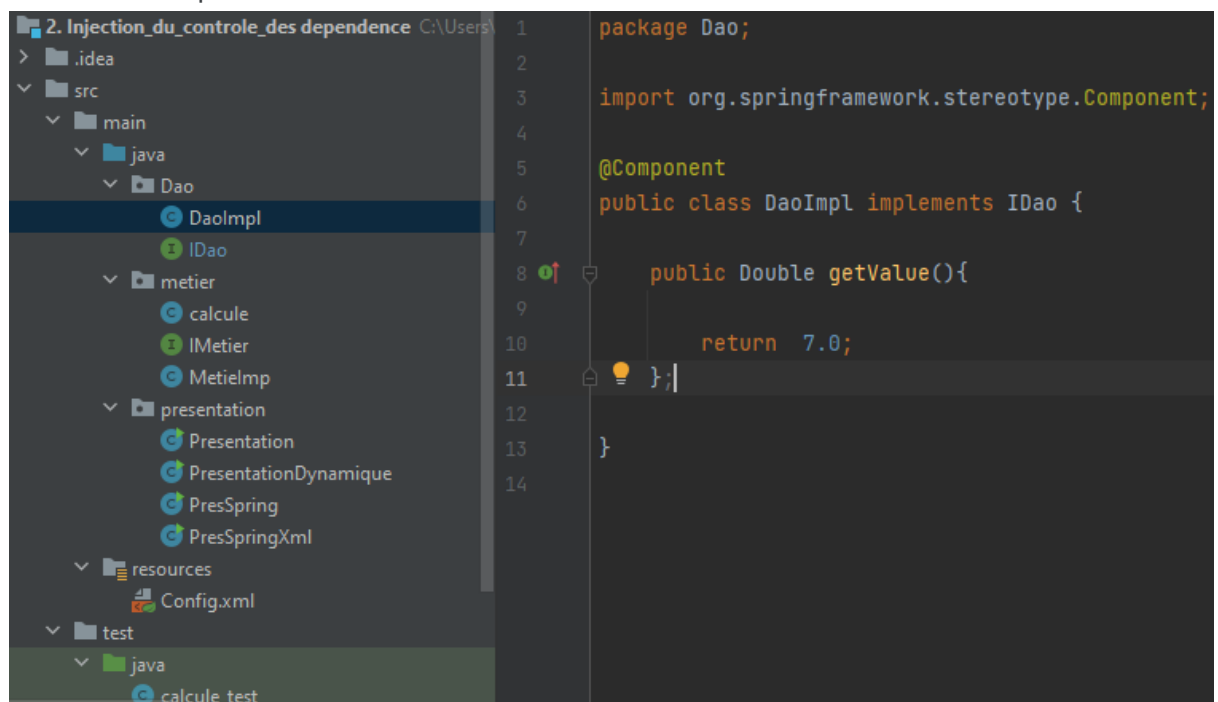
---

Compte rendu rédigé par Hajar Zarguan,  
Étudiante en cycle d'ingénieur: Génie des logiciels et systèmes informatiques distribuées  
École Normale Supérieure d'Enseignement Technique .

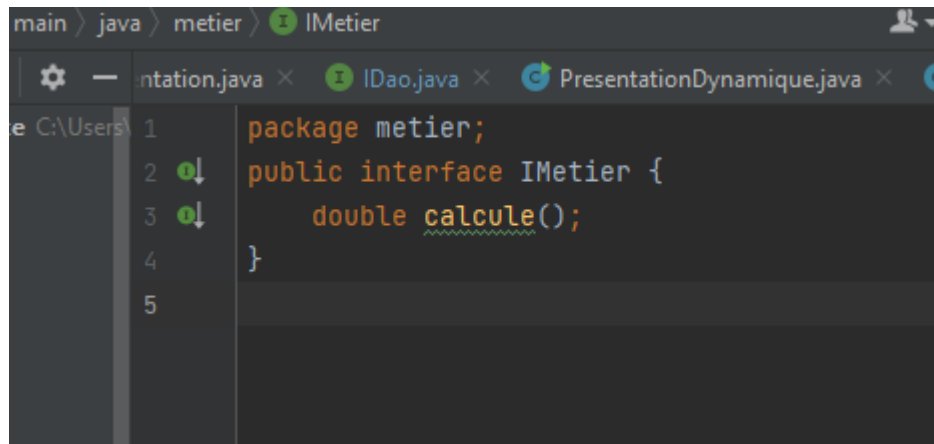
## 1. Créer l'interface IDao



## 2. Créer une implémentation de cette interface



### 3. Créer l'interface IMetier



The screenshot shows an IDE with a project named '2. Injection\_du\_controle\_des\_dependence'. The 'src/main/java' directory is expanded, showing a package 'metier'. The 'IMetier.java' file is open, showing the following code:

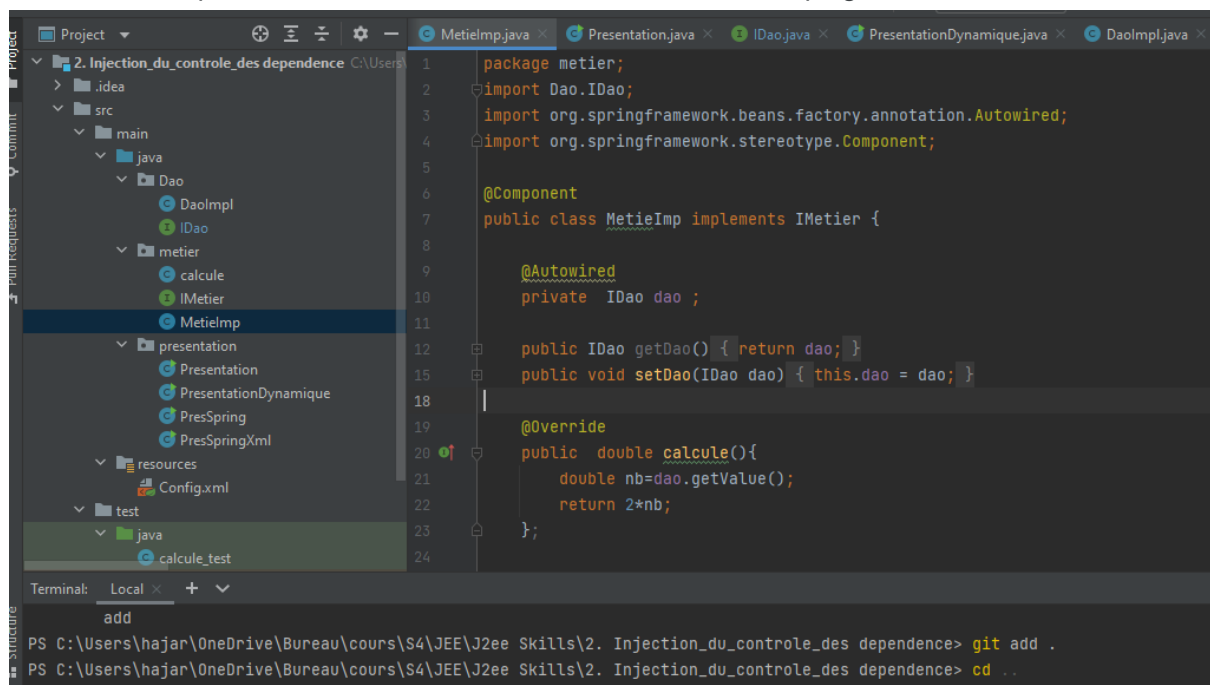
```
package metier;

public interface IMetier {

    double calcule();

}
```

### 4. Créer une implémentation de cette interface en utilisant le couplage faible



The screenshot shows the same IDE with the 'MetieImp.java' file open. The code is as follows:

```
package metier;

import Dao.IDao;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

@Component
public class MetieImp implements IMetier {

    @Autowired
    private IDao dao;

    public IDao getDao() { return dao; }

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

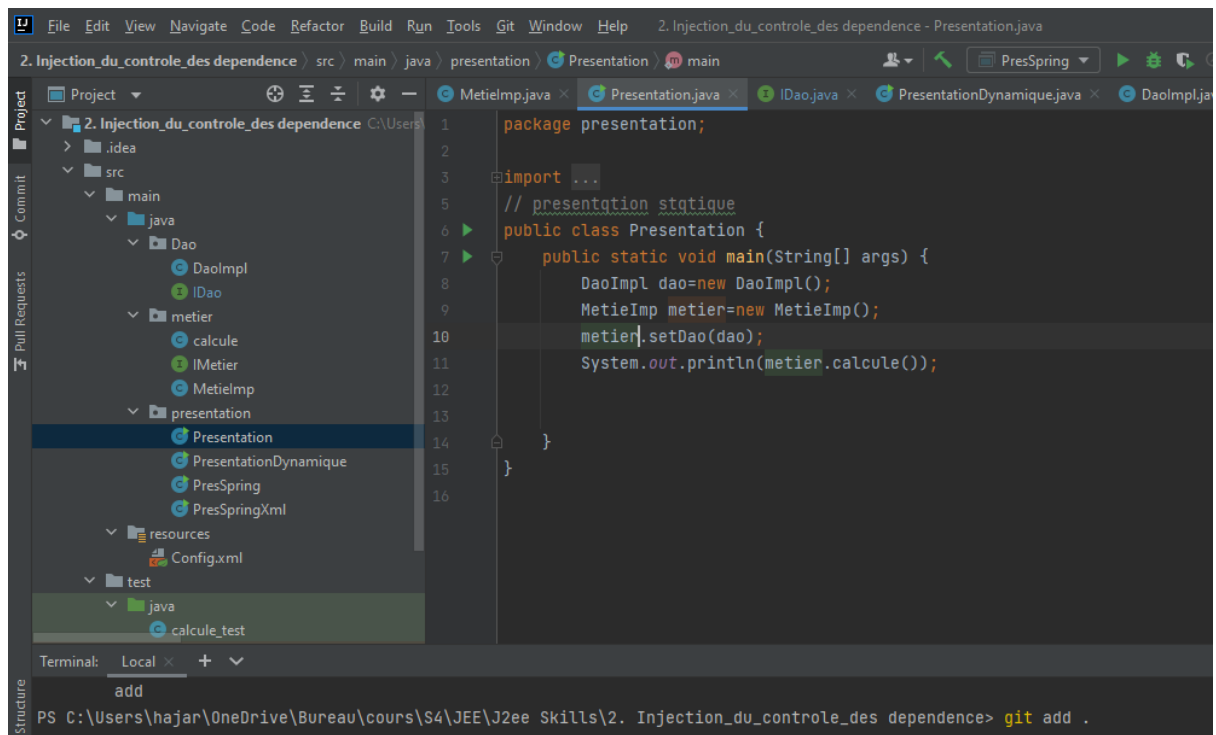
    @Override
    public double calcule(){
        double nb=dao.getValue();
        return 2*nb;
    };
}
```

The left sidebar shows the project structure, and the bottom terminal shows the following commands:

```
add
PS C:\Users\hajar\OneDrive\Bureau\cours\S4\JEE\J2ee Skills\2. Injection_du_controle_des dependance> git add .
PS C:\Users\hajar\OneDrive\Bureau\cours\S4\JEE\J2ee Skills\2. Injection_du_controle_des dependance> cd ..
```

### 5. Faire l'injection des dépendances :

#### a. Par instantiation statique



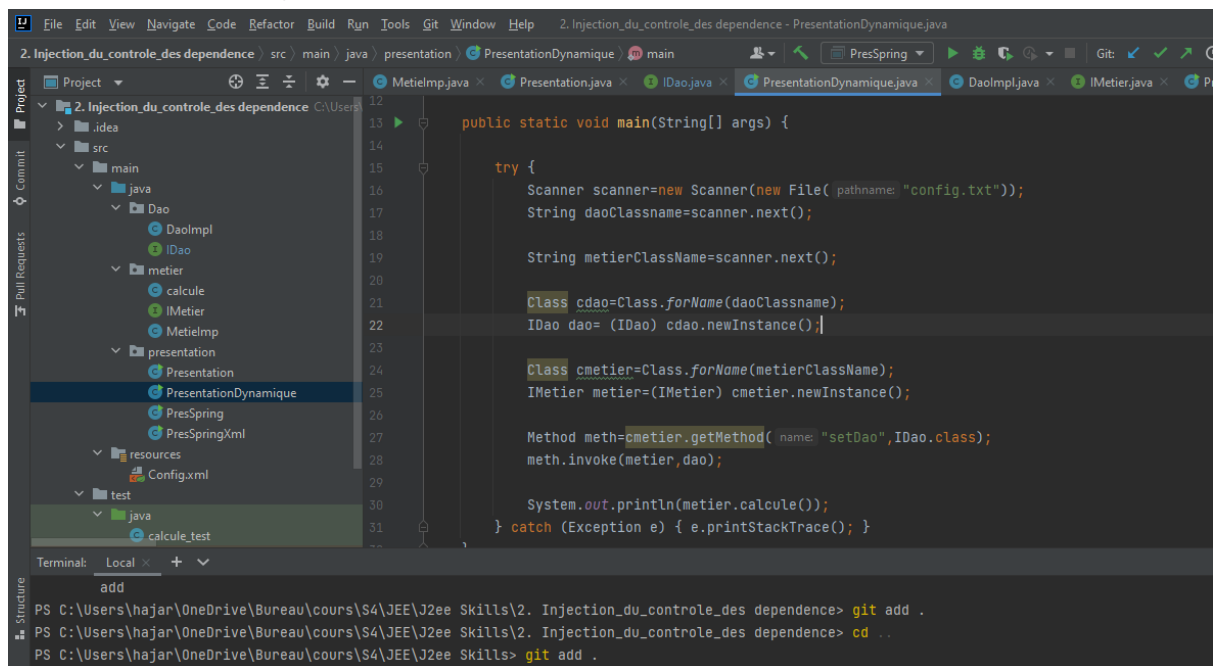
```
1 package presentation;
2
3 import ...
4
5 // presentation statique
6 public class Presentation {
7     public static void main(String[] args) {
8         DaoImpl dao=new DaoImpl();
9         MetieImp metier=new MetieImp();
10        metier.setDao(dao);
11        System.out.println(metier.calcul());
12    }
13 }
14
15 }
16 }
```

Terminal: Local x + v

add

PS C:\Users\hajar\OneDrive\Bureau\cours\S4\JEE\J2ee Skills\2. Injection\_du\_controle\_des\_dependence> git add .

## b. Par instanciation dynamique



```
12 public static void main(String[] args) {
13
14     try {
15         Scanner scanner=new Scanner(new File( pathname: "config.txt"));
16         String daoClassName=scanner.next();
17
18         String metierClassName=scanner.next();
19
20         Class cdao=Class.forName(daoClassName);
21         IDao dao= (IDao) cdao.newInstance();
22
23         Class cmetier=Class.forName(metierClassName);
24         IMetier metier=(IMetier) cmetier.newInstance();
25
26         Method meth=cmetier.getMethod( name: "setDao", IDao.class);
27         meth.invoke(metier,dao);
28
29         System.out.println(metier.calcul());
30     } catch (Exception e) { e.printStackTrace(); }
31 }
```

Terminal: Local x + v

add

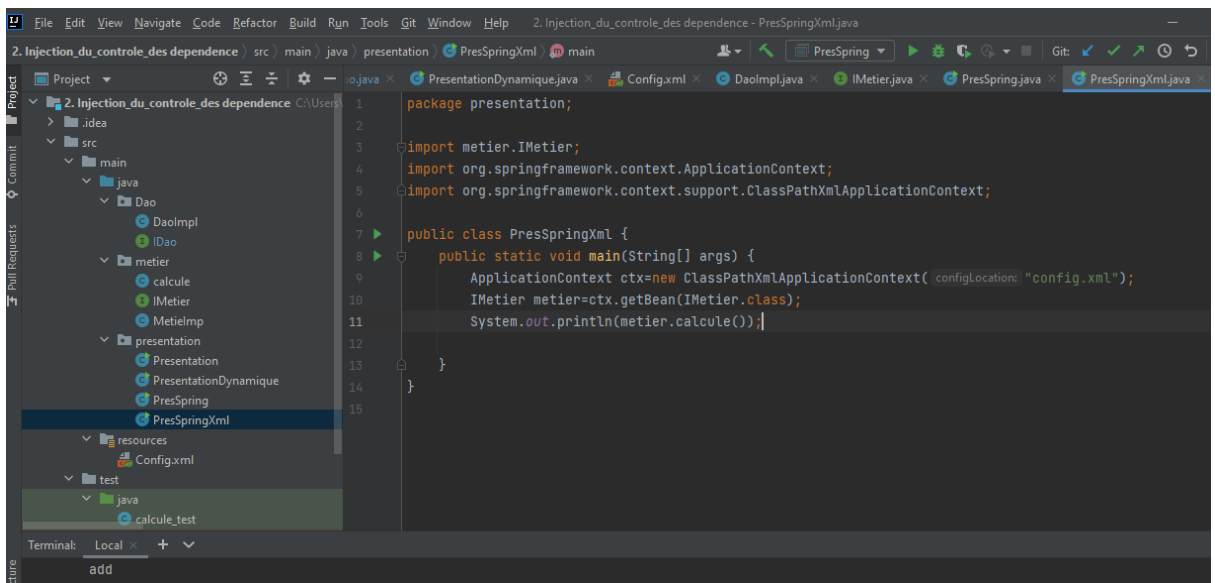
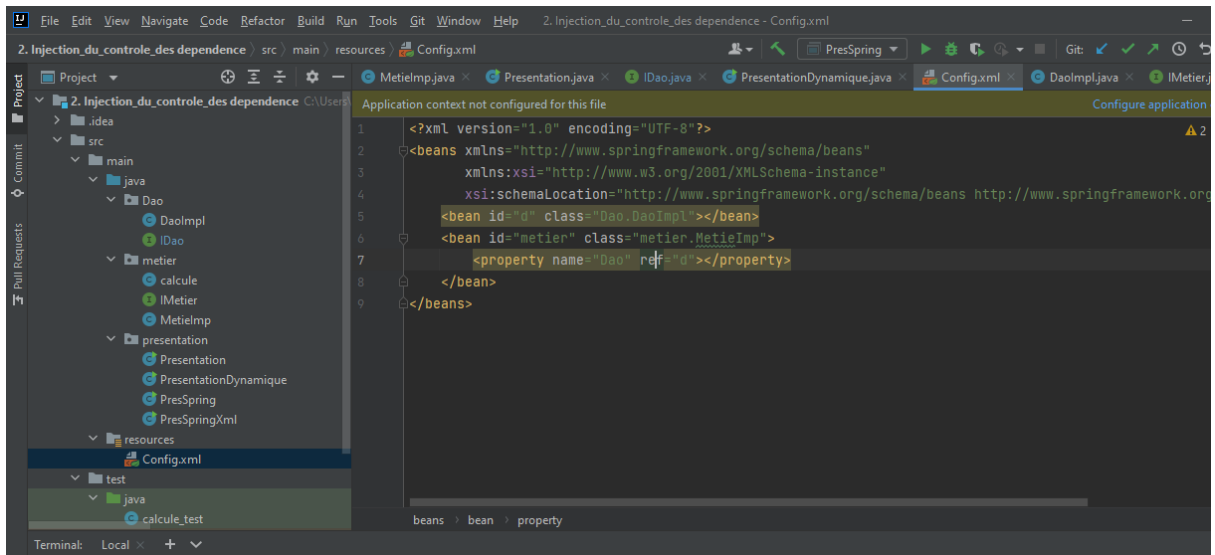
PS C:\Users\hajar\OneDrive\Bureau\cours\S4\JEE\J2ee Skills\2. Injection\_du\_controle\_des\_dependence> git add .

PS C:\Users\hajar\OneDrive\Bureau\cours\S4\JEE\J2ee Skills\2. Injection\_du\_controle\_des\_dependence> cd ..

PS C:\Users\hajar\OneDrive\Bureau\cours\S4\JEE\J2ee Skills> git add .

## c. En utilisant le Framework Spring

- Version XML



## - Version annotations

