



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**

Membre de 
HONORIS UNITED UNIVERSITIES

COMPTE RENDU :

TPI :

Réalisé par : - Mohamed Chakouri

Année universitaire : 2022/2023

OBJECTIF :

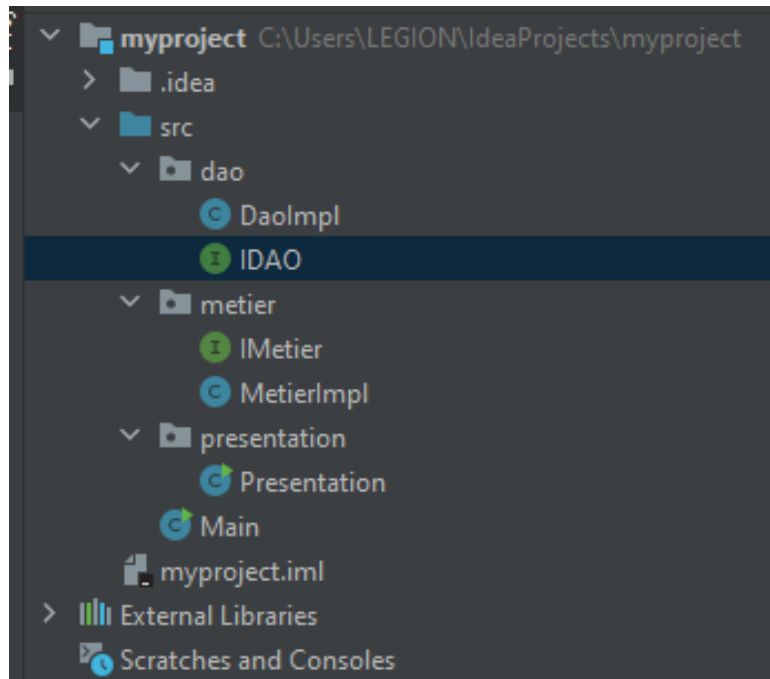
L'objectif est de savoir réaliser une application fermée à la modification et ouverte à l'extension en utilisant le couplage faible qui implique une dépendance indirecte entre les composants de l'application. Cela signifie que les composants peuvent interagir les uns avec les autres sans être directement liés.

MÉTHODES :

- La première méthode est : Instanciation Statique
- La deuxième méthode est : L'utilisation des annotations

Instanciación Estática

I- Composants du projet :



2- Interface:

La creation d'une interface **IDAO** dans le package **dao** :

```
1 package dao;
2
3 4 usages 1 implementation
public interface IDAO {
4 1 usage 1 implementation
    double getData();
5
6 }
```

La creation d'une interface **IMetier** dans le package **metier** :

```
1 package metier;  
2  
3 2 usages 1 implementation  
4 public interface IMetier {  
5     1 usage 1 implementation  
6     double calcul();  
7 }  
8
```

3-Class :

La classe **DaoImpl** qui implemente l'interface **IDAO** :

```
package dao;  
  
3 3 usages  
public class DaoImpl implements IDAO{  
    1 usage  
    @Override  
    public double getData() {  
        System.out.println("From SQL DB");  
        return(7);  
    }  
}
```

La classe **MetierImpl** qui implemente l'interface **IMetier** :

```
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*2021;
    }
    1 usage
    public void setDao(IDAO dao){
        this.dao=dao;
    }
}
```

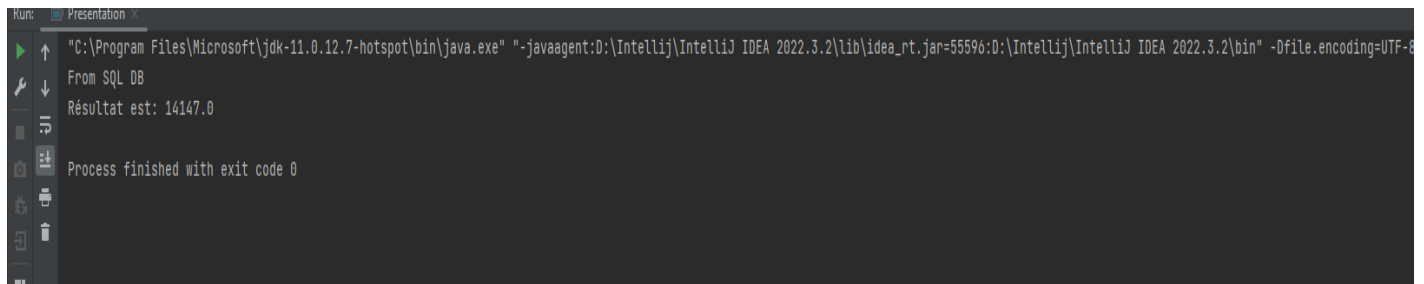
La classe *Presentation* dans le package *presentation* :

```
package presentation;
import dao.DaoImpl;
import metier.IMetier;
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 res = metier.calcul();
        System.out.println("Résultat est: "+ res);
    }
}
```

4- Execution :

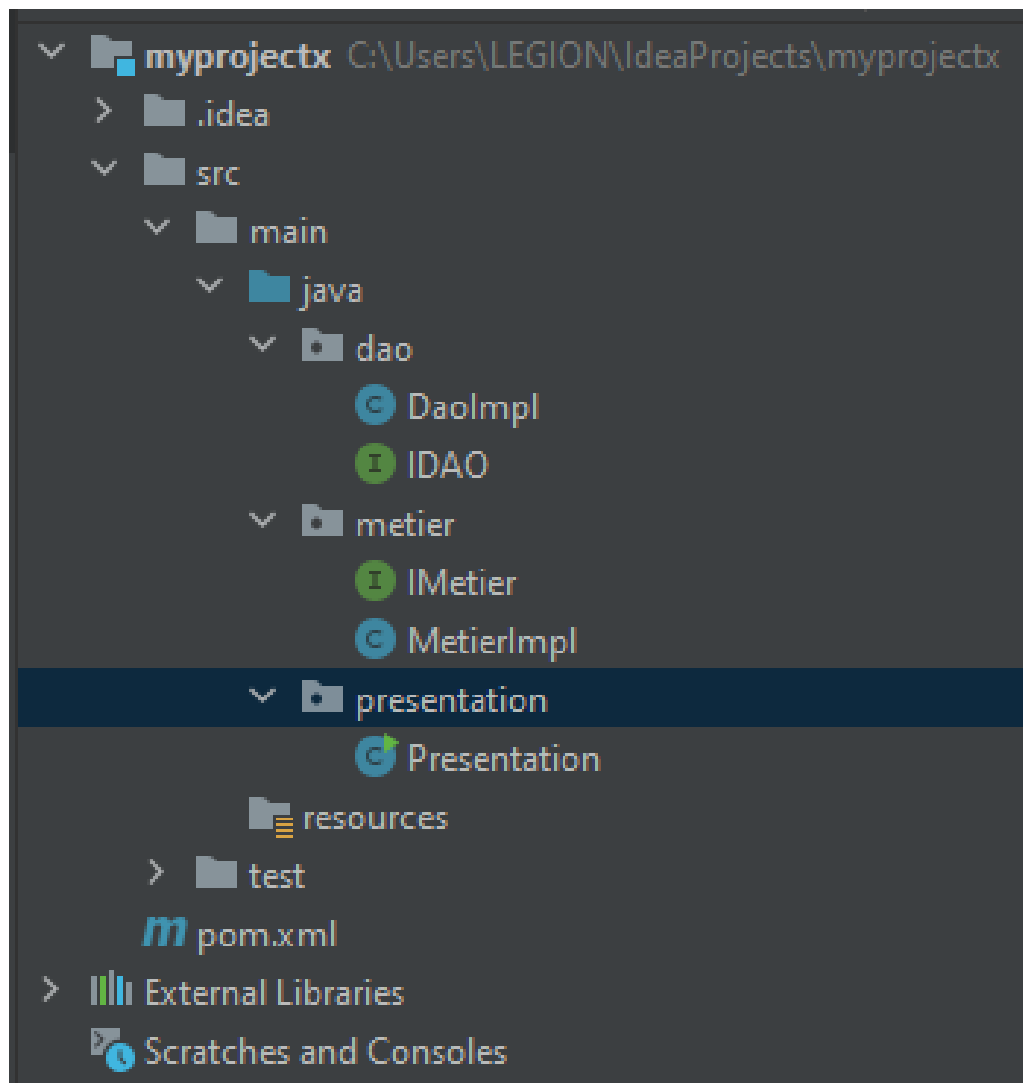


```
Kum: Presentation
" C:\Program Files\Microsoft\jdk-11.0.12.7-hotspot\bin\java.exe " -javaagent:D:\IntelliJ\IntelliJ IDEA 2022.3.2\lib\idea_rt.jar=55596:D:\IntelliJ\IntelliJ IDEA 2022.3.2\bin" -Dfile.encoding=UTF-8
From SQL DB
Résultat est: 14147.0
Process finished with exit code 0
```

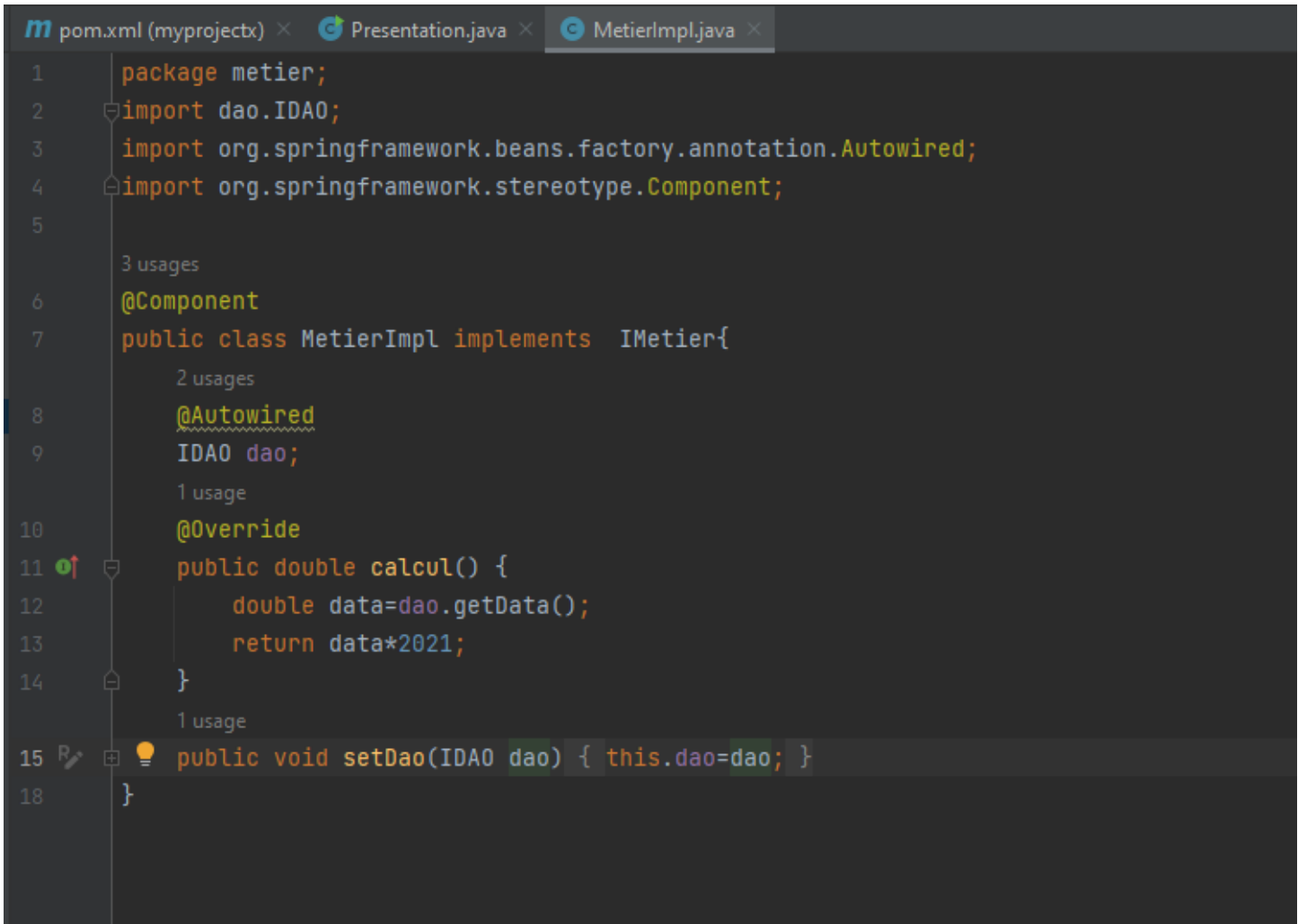



L'utilisation des annotations

I- Composants du projet :



2-Class :



```
1 package metier;
2 import dao.IDAO;
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.stereotype.Component;
5
6 @Component
7 public class MetierImpl implements IMetier{
8     @Autowired
9     IDAO dao;
10
11     @Override
12     public double calcul() {
13         double data=dao.getData();
14         return data*2021;
15     }
16
17     public void setDao(IDAO dao) { this.dao=dao; }
18 }
```

- l'annotation "@Component" est utilisée pour marquer une classe Java en tant que composant Spring. Cette annotation indique que la classe doit être gérée par le conteneur Spring et qu'elle peut être utilisée dans l'injection de dépendances.
- "@Autowired" : c'est une annotation qui permet d'activer l'injection de dépendance d'une manière automatique

La classe *Presentation* :



```
m pom.xml (myprojectx) x Presentation.java x MetierImpl.java x IDAO.java x
1 package presentation;
2 import metier.IMetier;
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
5
6 public class Presentation {
7     public static void main(String[] args){
8
9         ApplicationContext context= new AnnotationConfigApplicationContext( ...basePackages: "dao","metier");
10        IMetier metier= context.getBean(IMetier.class);
11        System.out.println("Résultat est: "+ metier.calcul() );
12    }
13 }
```

3- Execution :

```
Kun: Presentation
" C:\Program Files\Microsoft\jdk-11.0.12.7-hotspot\bin\java.exe " "-javaagent:D:\IntelliJ\IntelliJ IDEA 2022.3.2\lib\idea_rt.jar=55596:D:\IntelliJ\IntelliJ IDEA 2022.3.2\bin" -Dfile.encoding=UTF-8
From SQL DB
Résultat est: 14147.0
Process finished with exit code 0
```

