

Compte Rendu

A. Introduction

Dans un projet il y a les exigences fonctionnelles et les exigences techniques (il faut prendre en considération la maintenance pour créer une application facile à maintenir il doit être fermé à la modification et ouvert à l'extension et pour créer ce projet qui respecte tous ces critères il faut utiliser

le principe de l'inversion de contrôle qui permet au développeur de se concentrer au code métier alors que le Framework (spring) qui va s'occuper du code technique (la sécurité, l'accès aux données, les transactions).

B. Enoncé

Les étapes sont comme suit :

- 1) Créer l'interface IDAO
- 2) Créer une implémentation de cette interface IDAO
- 3) Créer l'interface IMétier
- 4) Créer une implémentation de l'interface IMétier
- 5) Créer la couche Présentation en faisant l'injection des dépendances

- ❖ Par implémentation statique
- ❖ Par implémentation dynamique
- ❖ En utilisant le Framework Spring :

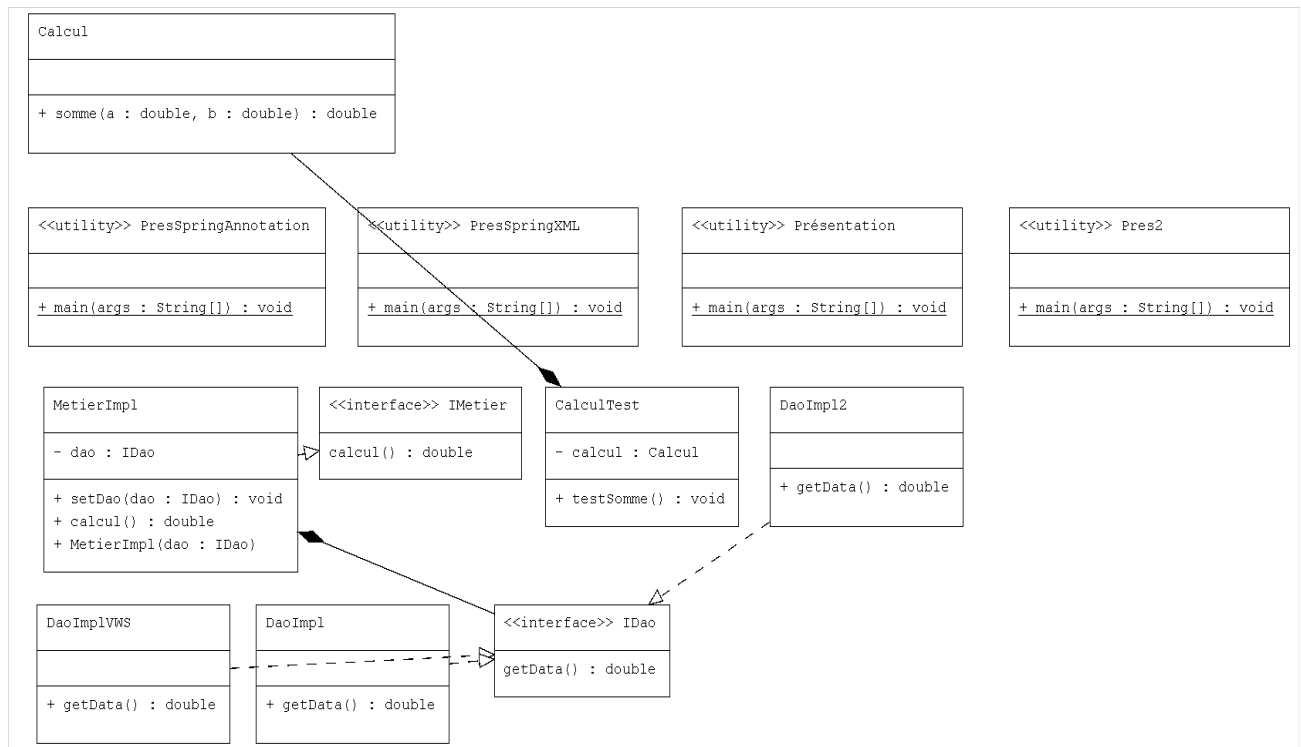


Avec la version XML

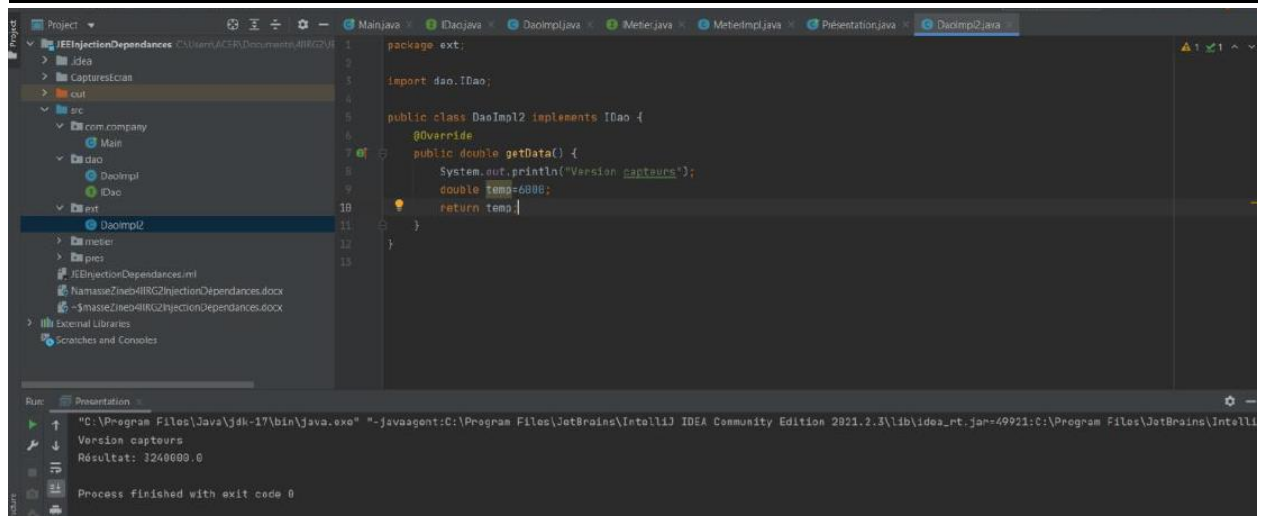
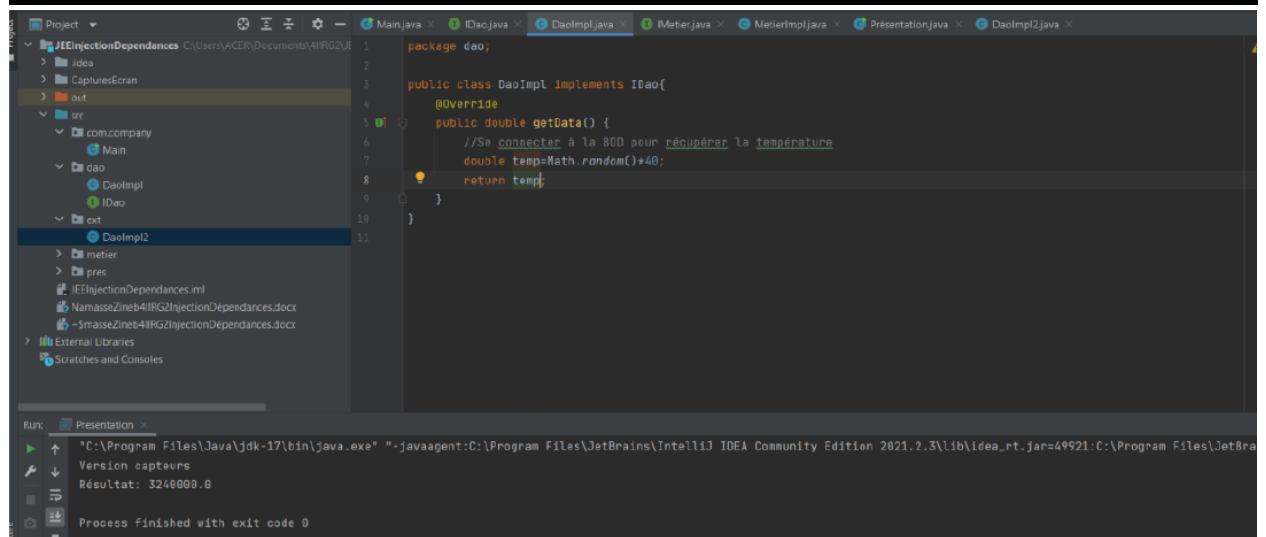
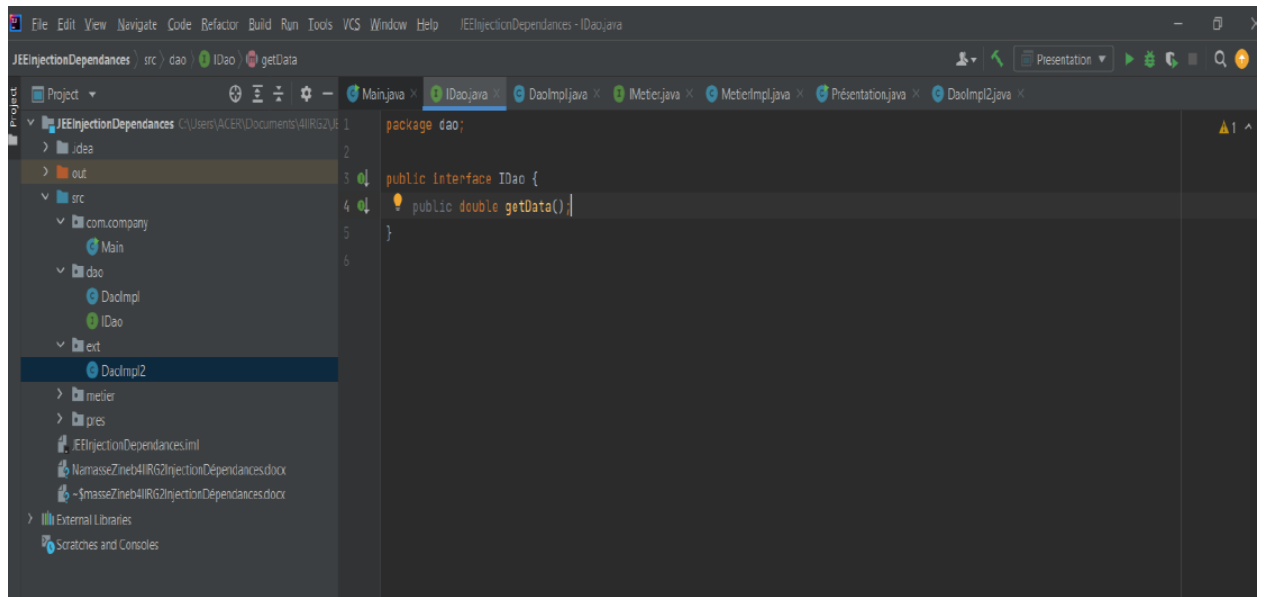


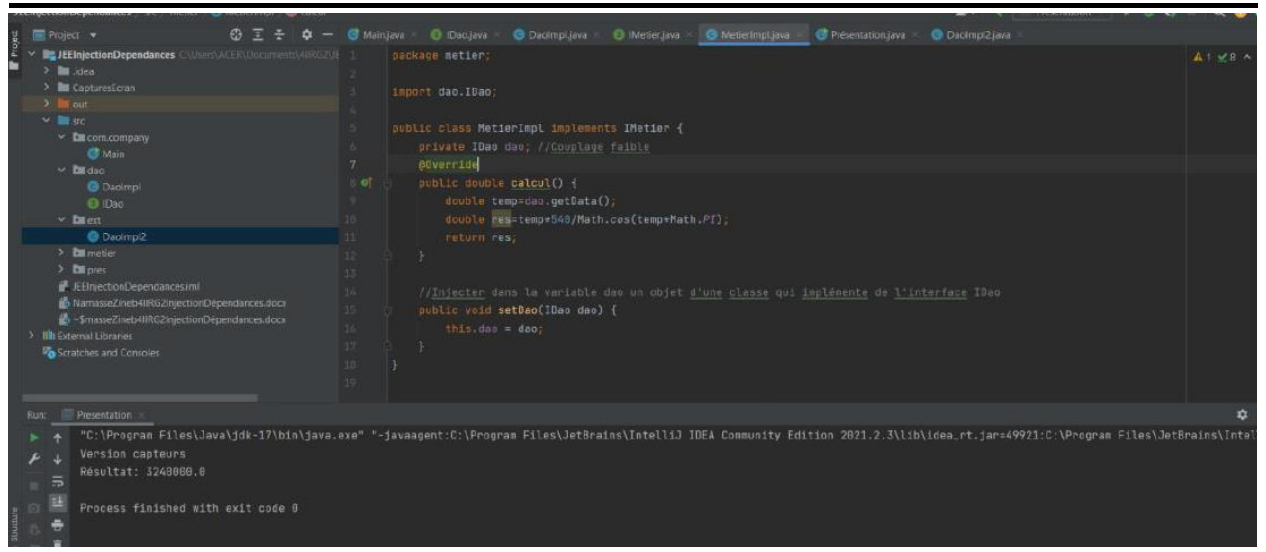
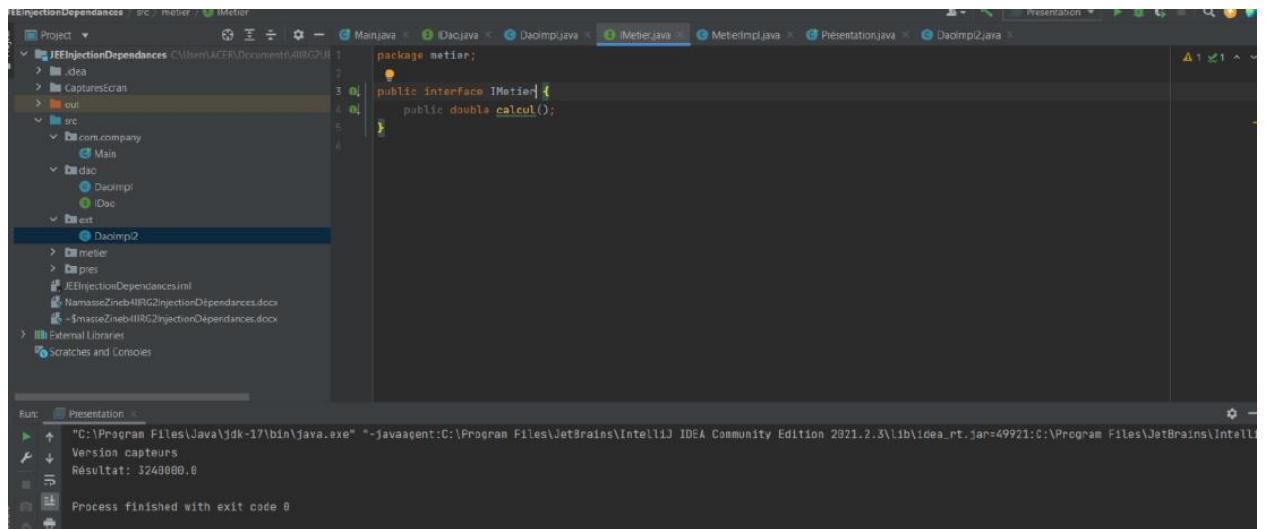
Avec la version Annotations

C. Conception et Architecture

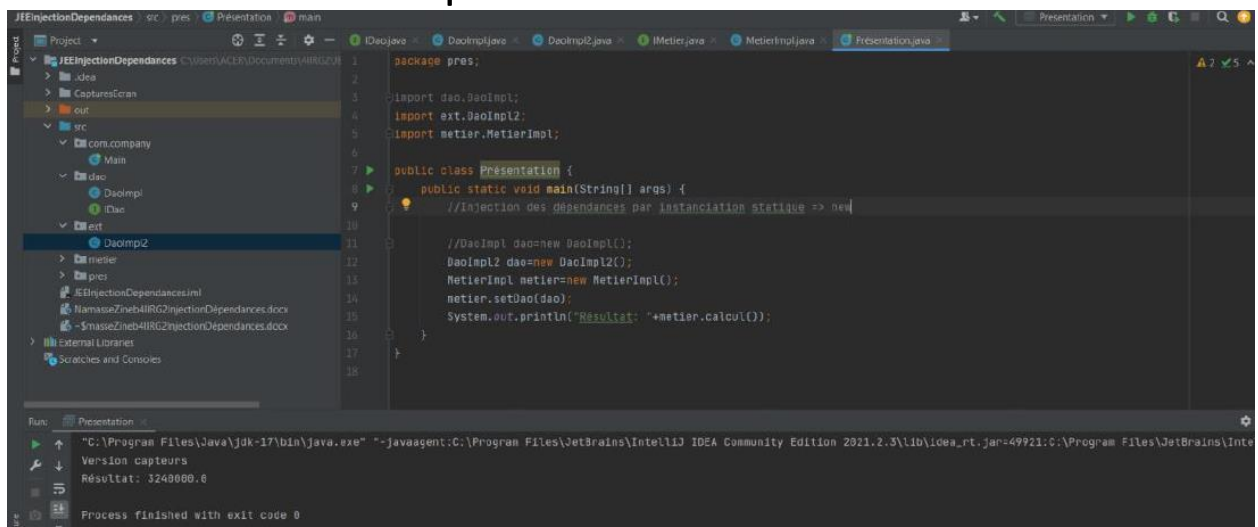


D. Captures d'écran :

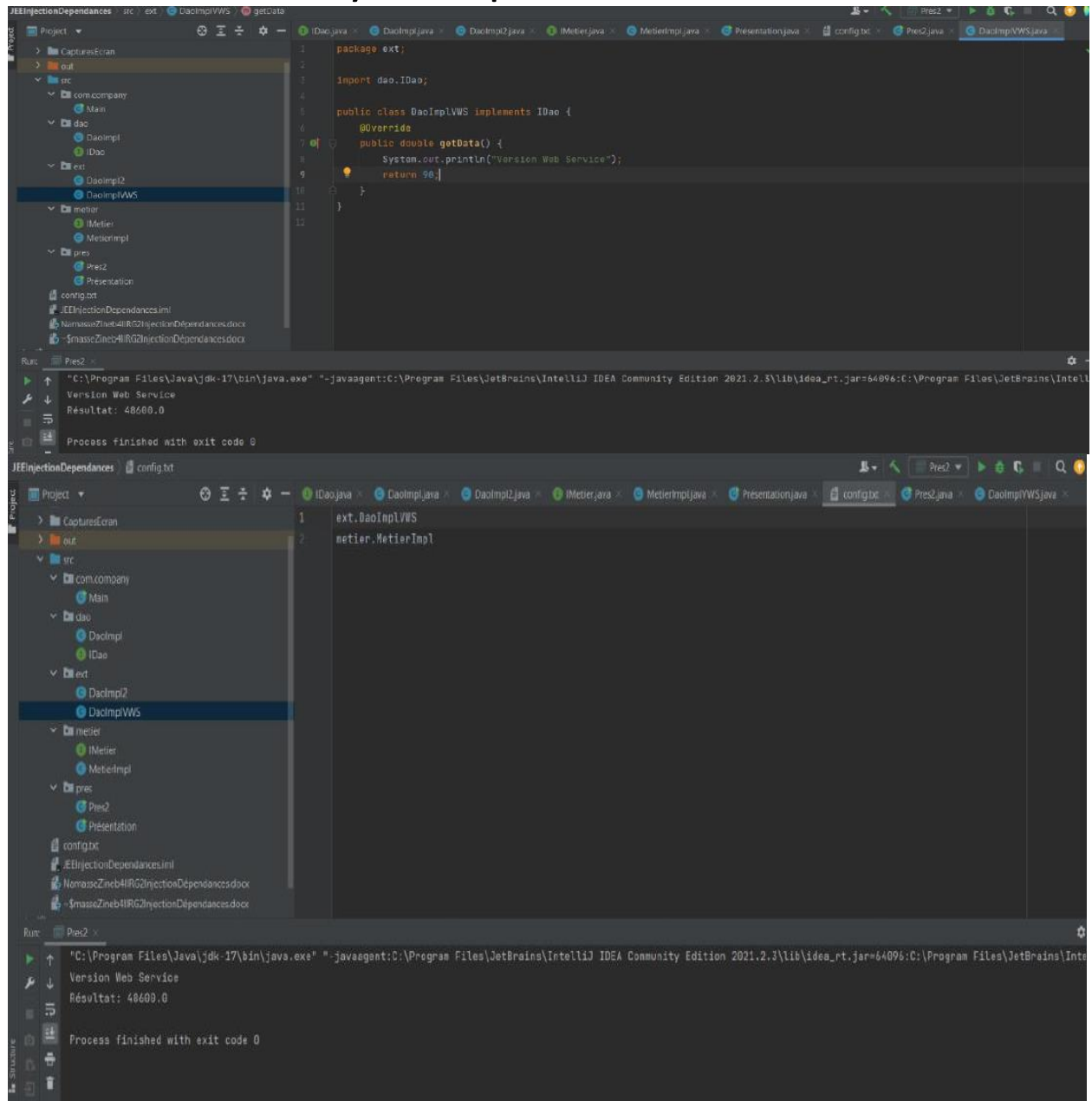


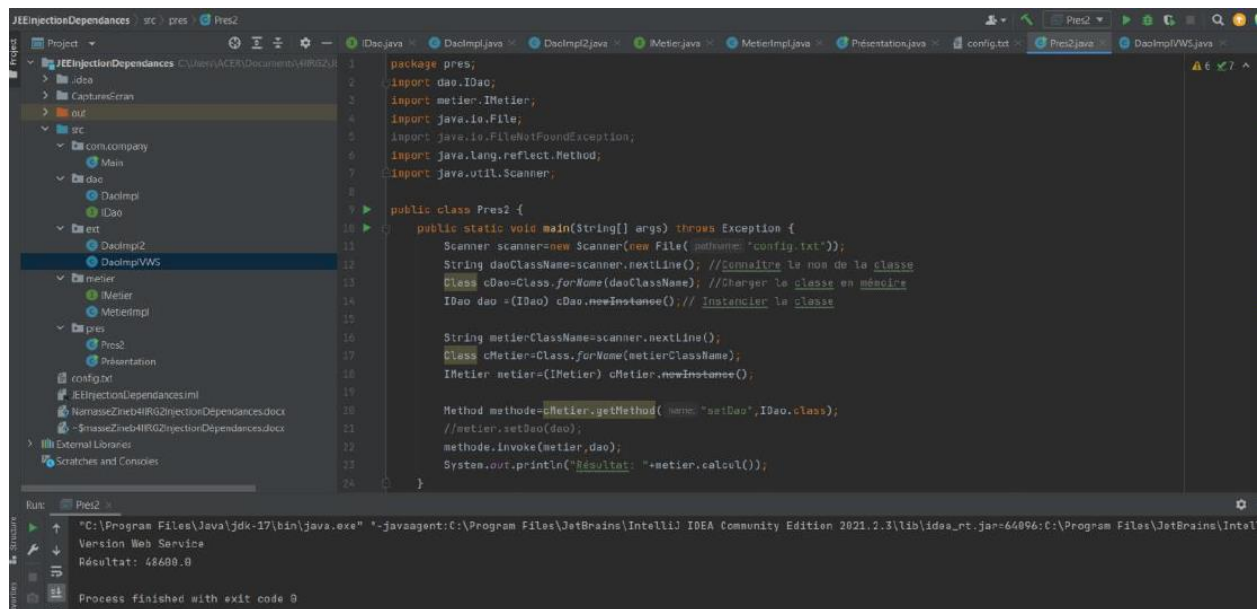


L'instanciation statique :



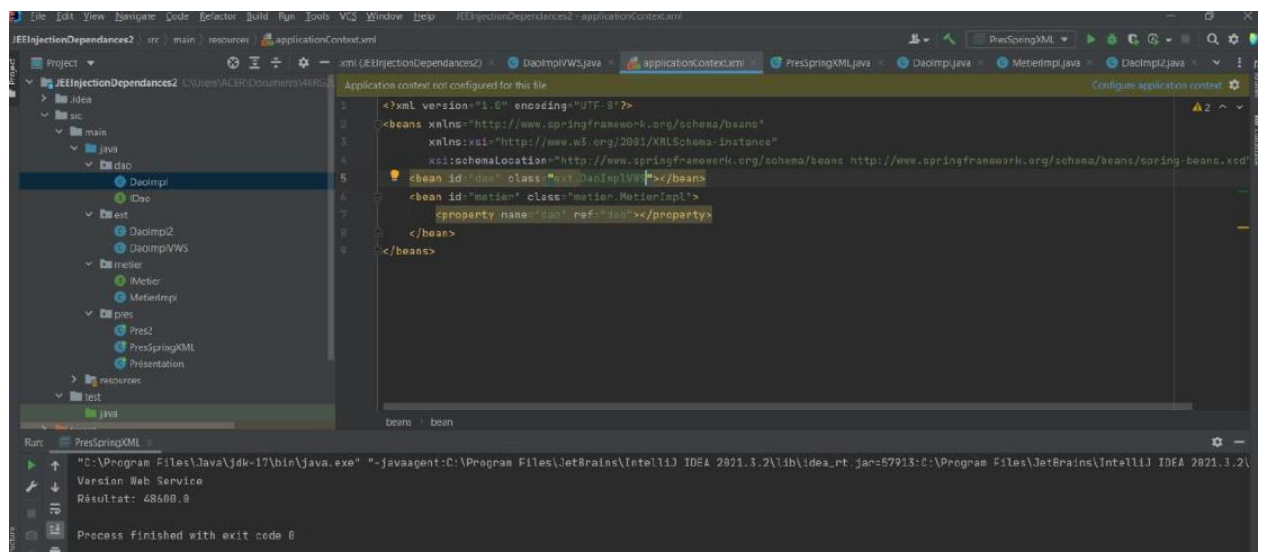
L'instanciation dynamique :

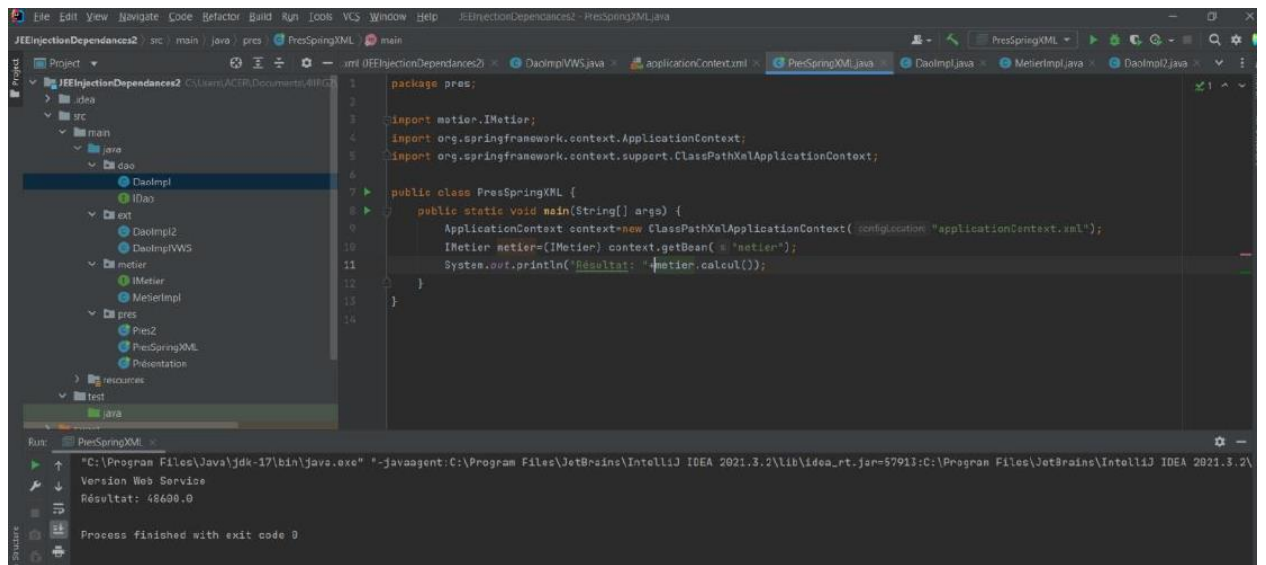




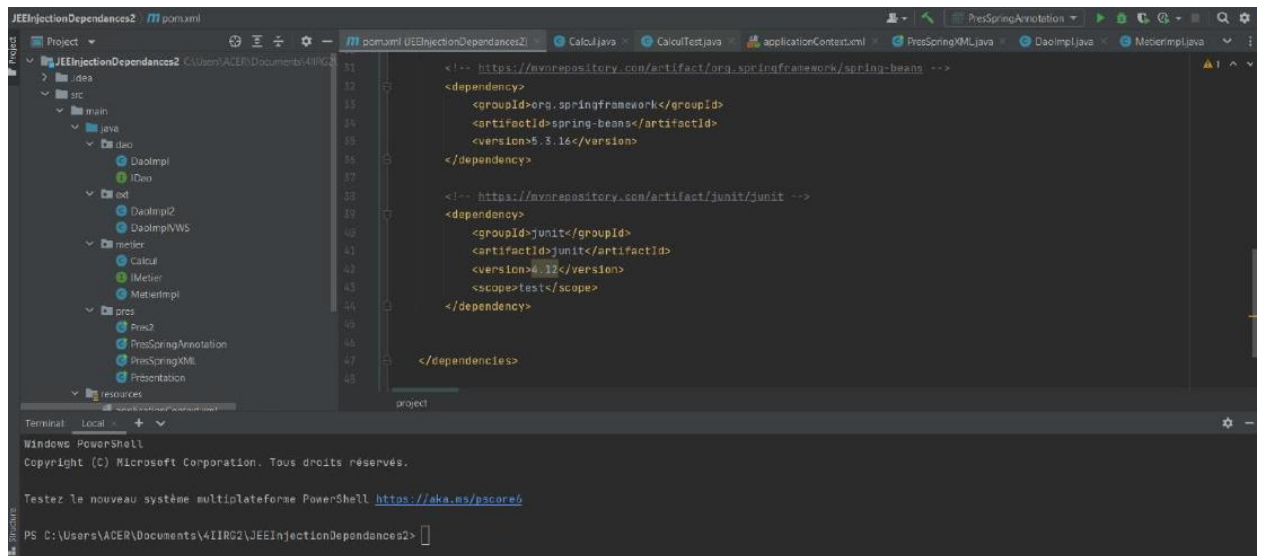
Spring :

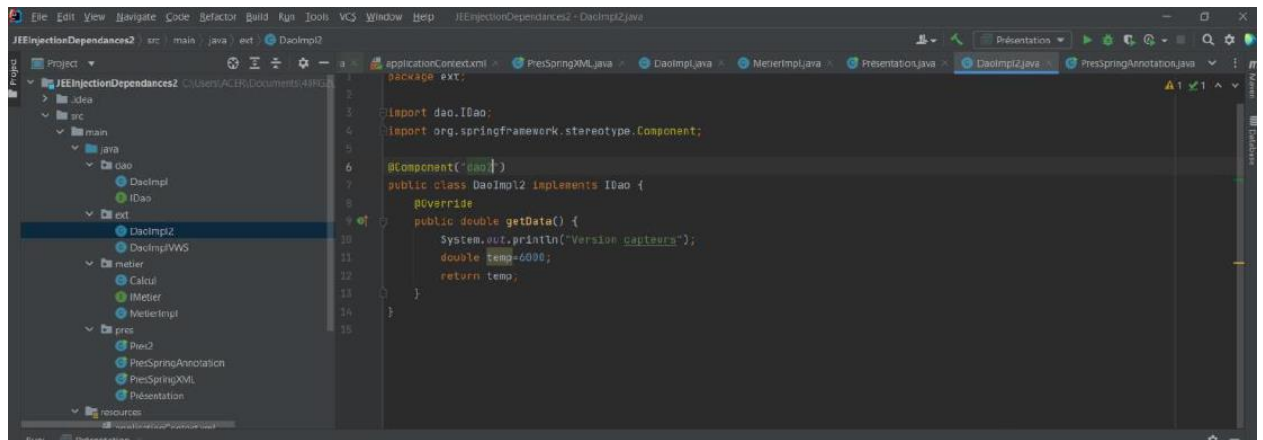
a) Version XML :



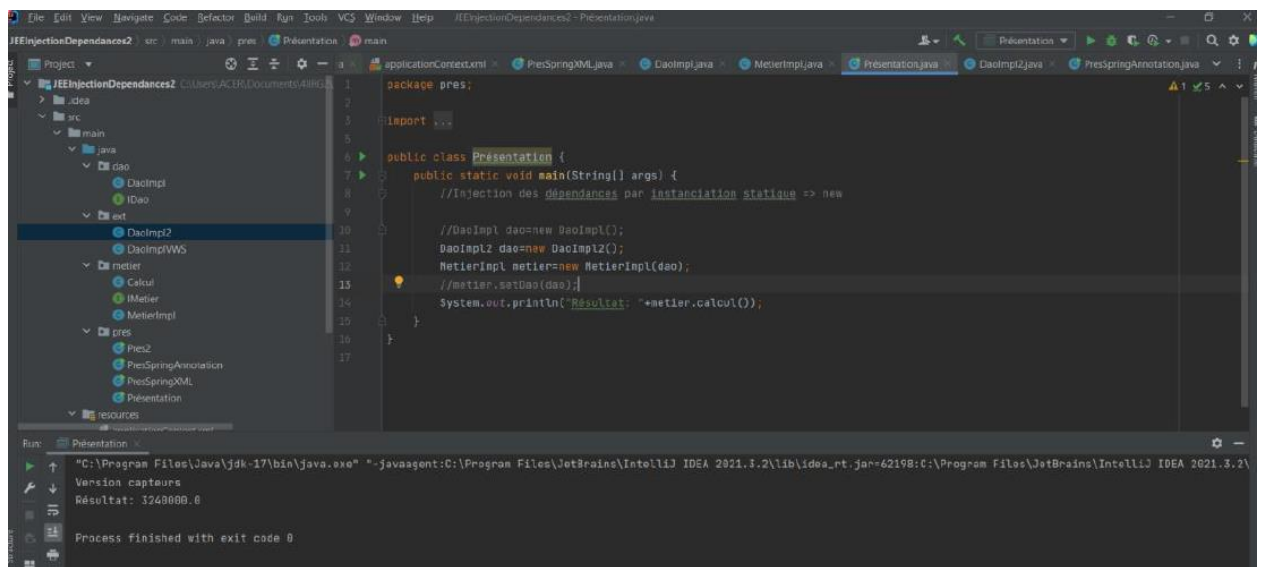


b) Version Annotation :





```
1 package ext;
2
3 import dao.IDao;
4 import org.springframework.stereotype.Component;
5
6 @Component("dao")
7 public class DaoImpl2 implements IDao {
8     @Override
9     public double getData() {
10         System.out.println("Version captureurs");
11         double temp=6096;
12         return temp;
13     }
14 }
15
```



```
1 package pres;
2
3 import java.util.*;
4
5 public class Presentation {
6     public static void main(String[] args) {
7         //Injection des dependances par instanciation statique => new
8
9         //DaoImpl dao=new DaoImpl();
10         DaoImpl2 dao=new DaoImpl2();
11         MetierImpl metier=new MetierImpl(dao);
12         //metier.setDao(dao);
13         System.out.println("Résultat: "+metier.calcul());
14     }
15 }
16
17
```

Run: Presentation

"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\lib\idea_rt.jar=62198:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\bin" -Didea.config.path=C:\Users\ACR6\Documents\IIRG\workspace\JEEInjectionDependances2 -Didea.system.path=C:\Users\ACR6\Documents\IIRG\workspace\JEEInjectionDependances2 -Didea.version=2021.3.2

Version captureurs

Résultat: 3248696.6

Process finished with exit code 0


```

1 package metier;
2
3
4 import dao.IDao;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.beans.factory.annotation.Qualifier;
7 import org.springframework.stereotype.Component;
8
9 @Component
10 public class MetierImpl implements IMetier {
11     //Complète la classe
12     private IDao dao;
13
14     public MetierImpl(IDao dao) {
15         this.dao = dao;
16     }
17
18     @Override
19     public double calcul() {
20         double temp=dao.getRate();
21         double res=temp*549/Math.cos(temp*Math.PI);
22         return res;
23     }
24
25     //injecter dans la variable dao un objet d'une classe qui implémente de l'interface IDao
26     public void setDao(IDao dao) { this.dao = dao; }
27 }

```

```

1 package pres;
2
3 import metier.IMetier;
4 import org.springframework.context.ApplicationContext;
5 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
6 import org.springframework.context.annotation.ContextAnnotationAutowireCandidateResolver;
7
8 public class PresSpringAnnotation {
9     public static void main(String[] args) {
10         ApplicationContext context=new AnnotationConfigApplicationContext( "dao","metier","ext");
11         IMetier metier= context.getBean(IMetier.class);
12         System.out.println("Résultat: "+metier.calcul());
13     }
14 }

```

Run: Présentation

Process finished with exit code 0

E. Conclusion

La réalisation de ce TP, m'a permis d'apprendre de nouveaux astuces et technologies, car c'est une première avec le logiciel IntelliJ JDEA ainsi que la technologie JEE, mais grâce aux vidéos du

professeur la tâche m'est devenue plus simple et ça m'a permis aussi de comprendre le principe de « l'ouverture à l'extension et la fermeture à la modification ».