

Compte Rendu de L'activité Pratique de l'Injection des dépendances

I) Introduction

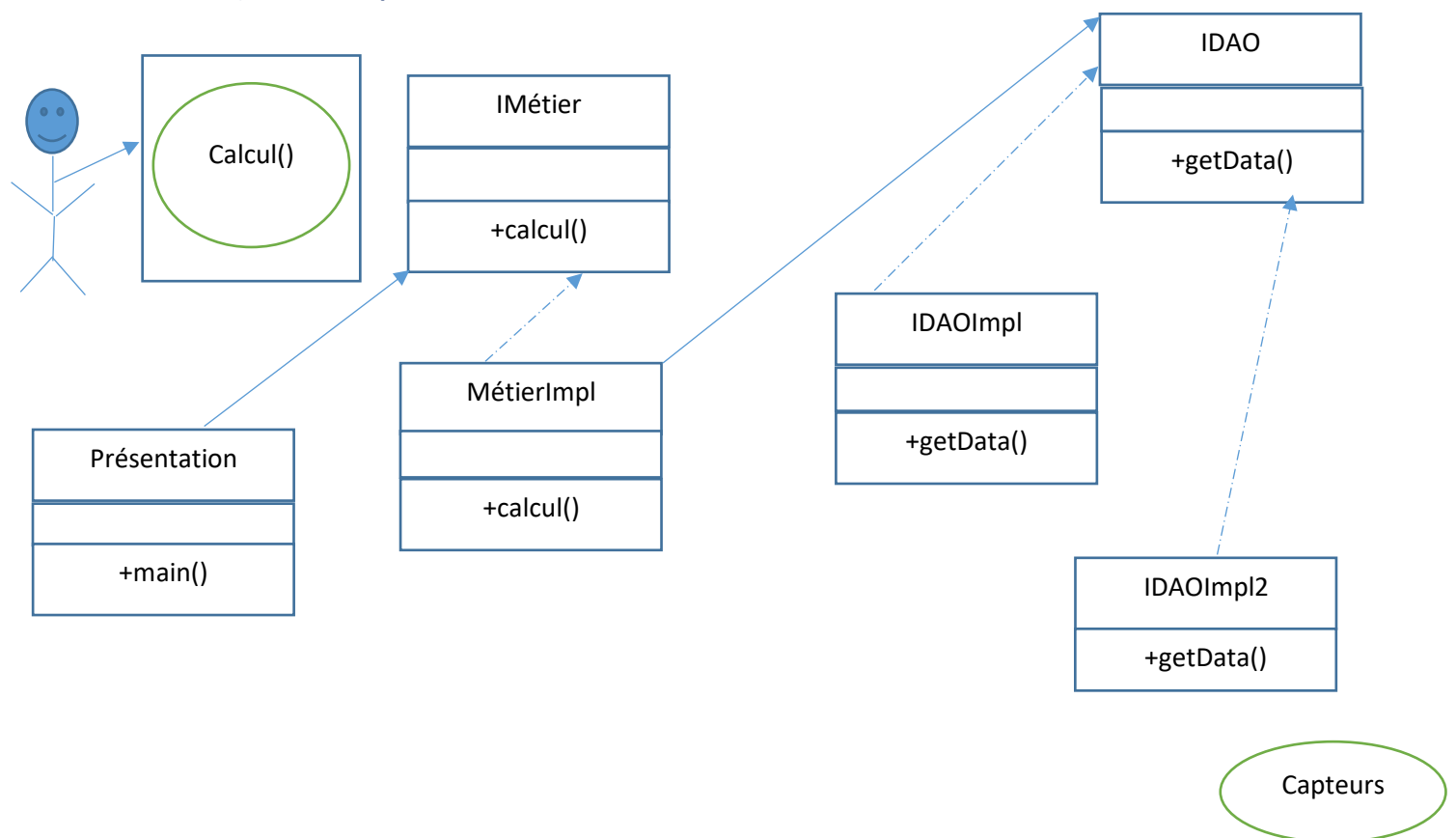
Comment créer une application fermée à la modification et ouverte à l'extension grâce à l'injection des dépendances ?

II) Enoncé

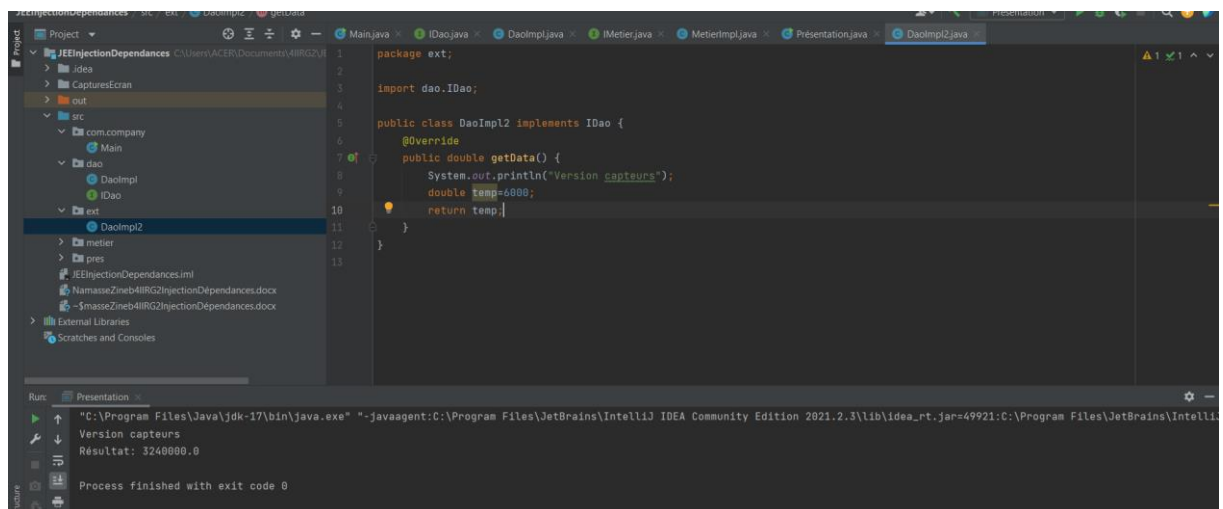
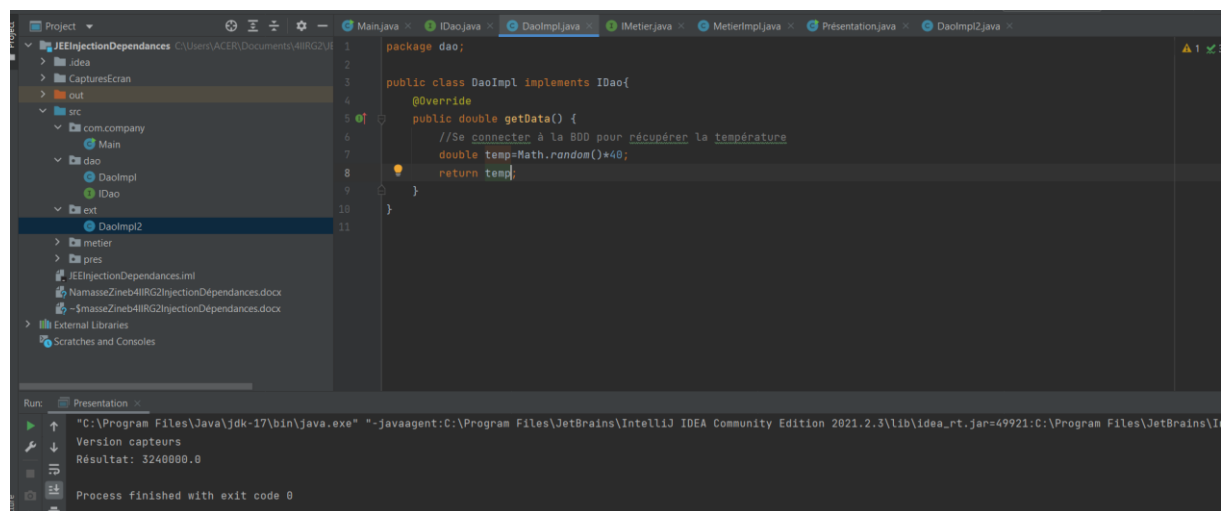
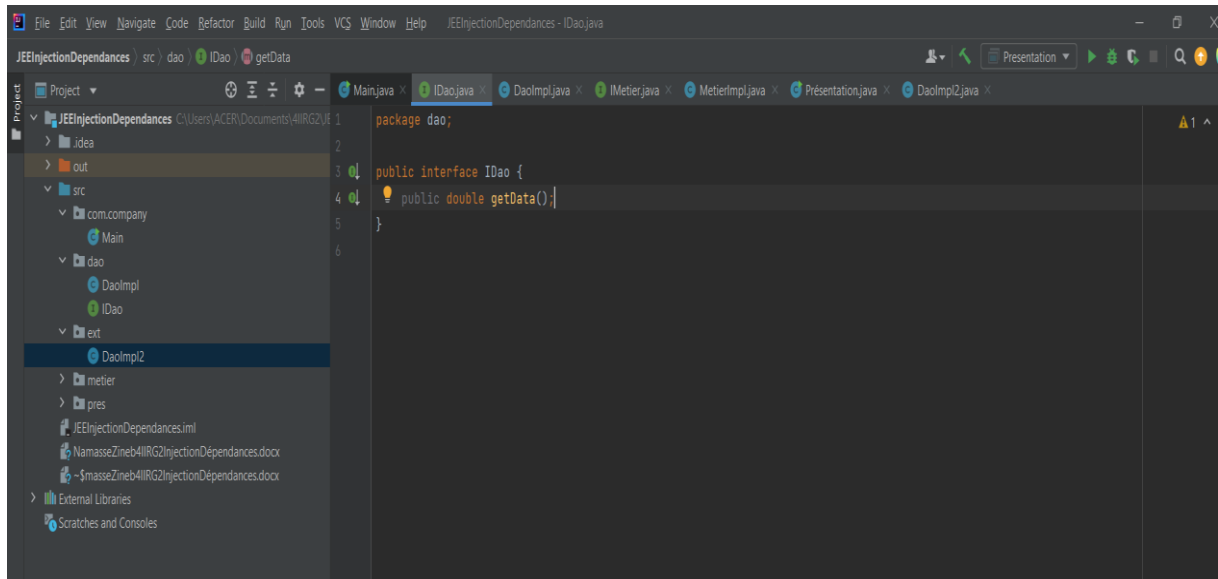
L'objectif de cette activité pratique est de concevoir une application fermée à la modification et ouverte à l'extension grâce à l'injection des dépendances. Les étapes sont comme suit :

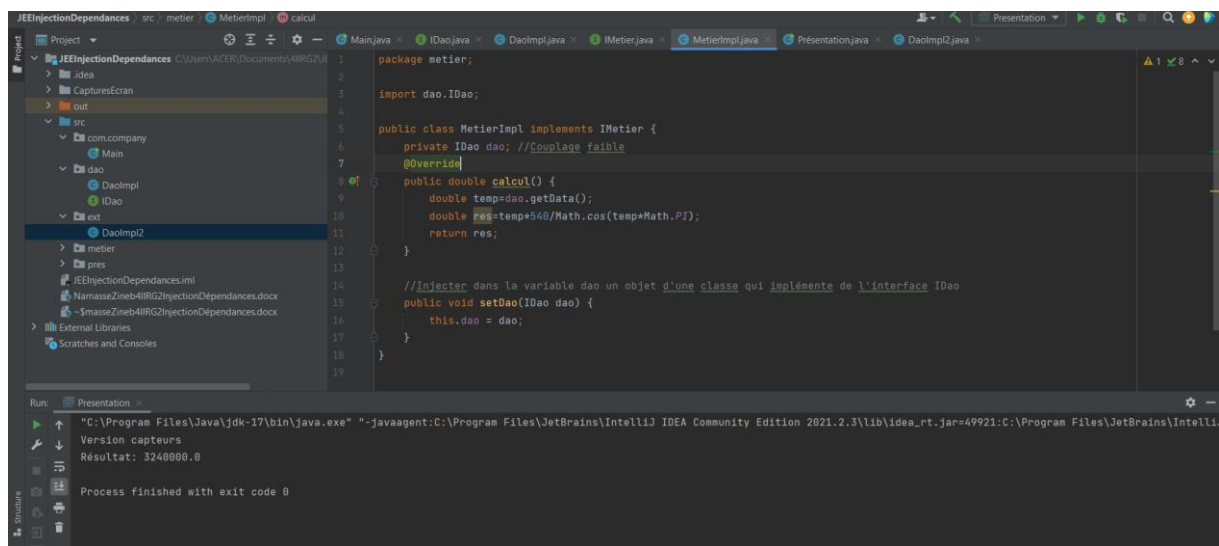
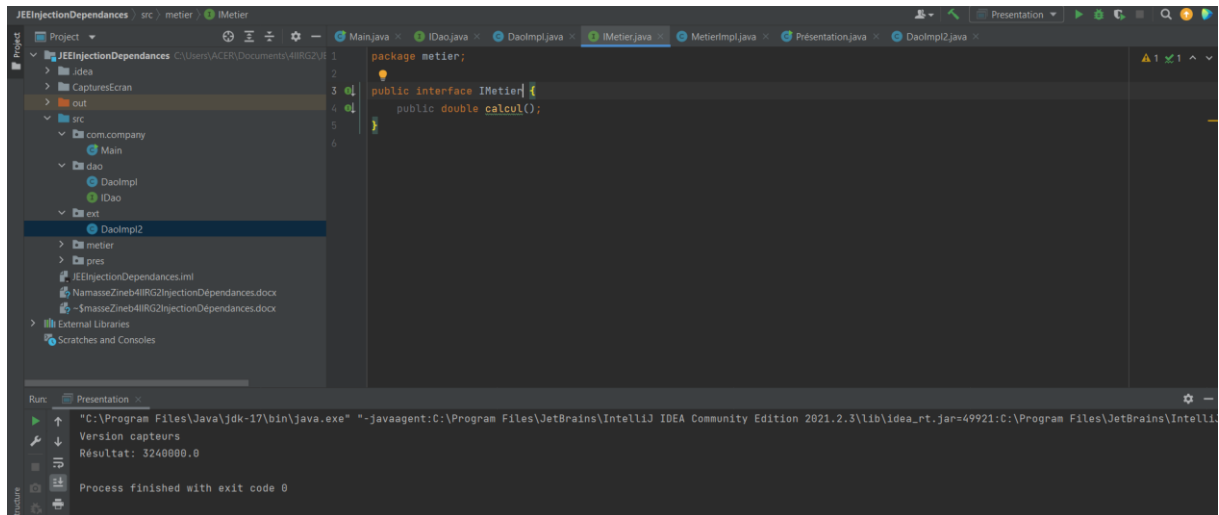
- 1) Créer l'interface IDAO
- 2) Créer une implémentation de cette interface IDAO
- 3) Créer l'interface IMétier
- 4) Créer une implémentation de l'interface IMétier
- 5) Créer la couche Présentation en faisant l'injection des dépendances
 - a) Par implémentation statique
 - b) Par implémentation dynamique
 - c) En utilisant le framework Spring :
 - ⇒ Avec la version XML
 - ⇒ Avec la version Annotations

III) Conception et Architecture



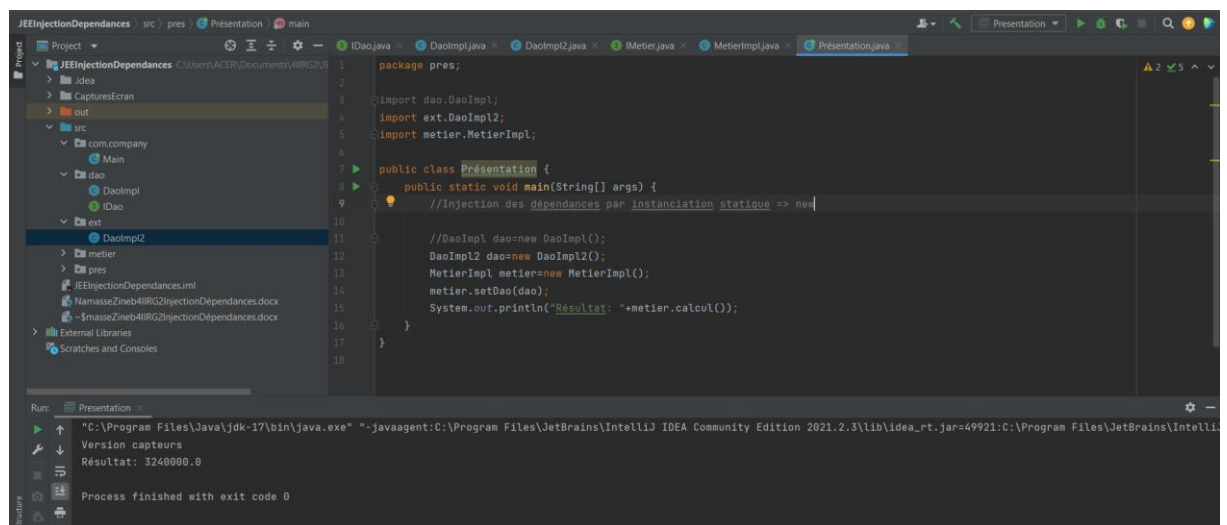
IV) Captures d'écran :



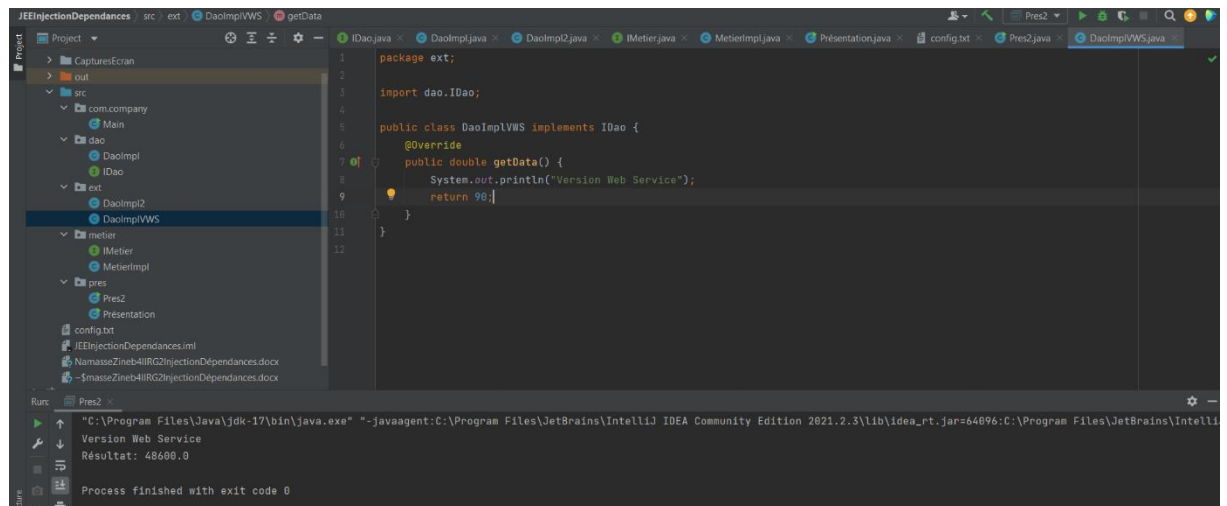


Injection des dépendances avec :

- 1) L'instanciation statique :



2) L'instanciation dynamique :



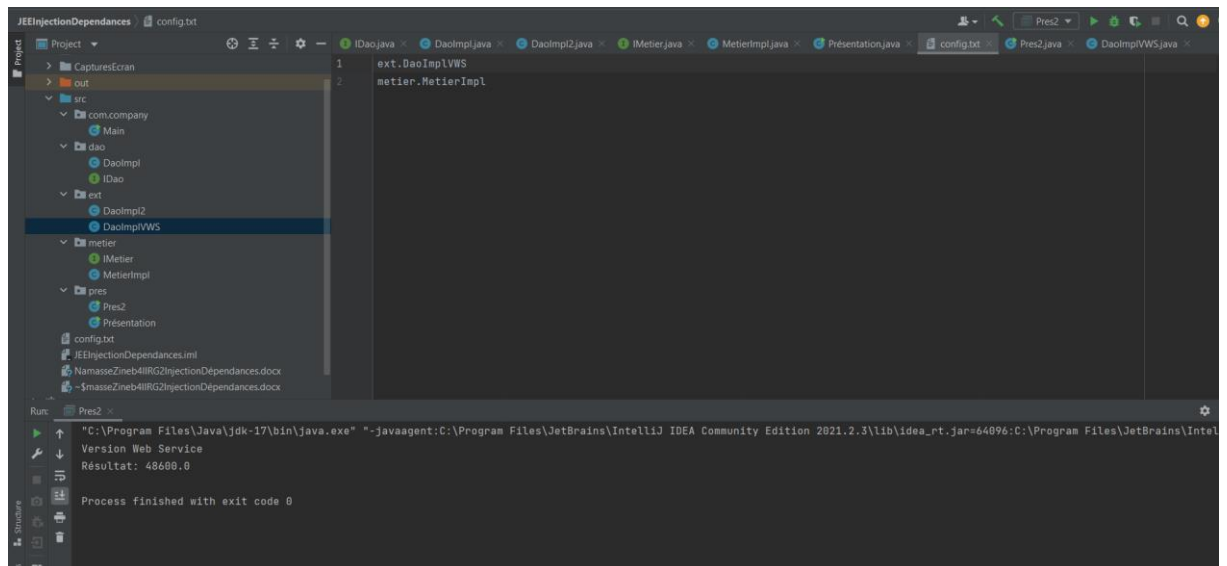
```
1 package ext;
2
3 import dao.IDao;
4
5 public class DaoImplVWS implements IDao {
6     @Override
7     public double getData() {
8         System.out.println("Version Web Service");
9         return 98;
10    }
11 }
12
```

Run: Pres2

"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\lib\idea_rt.jar=64096:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\bin" -jar C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\bin\idea_rt.jar 64096

Version Web Service
Résultat: 48600.0

Process finished with exit code 0



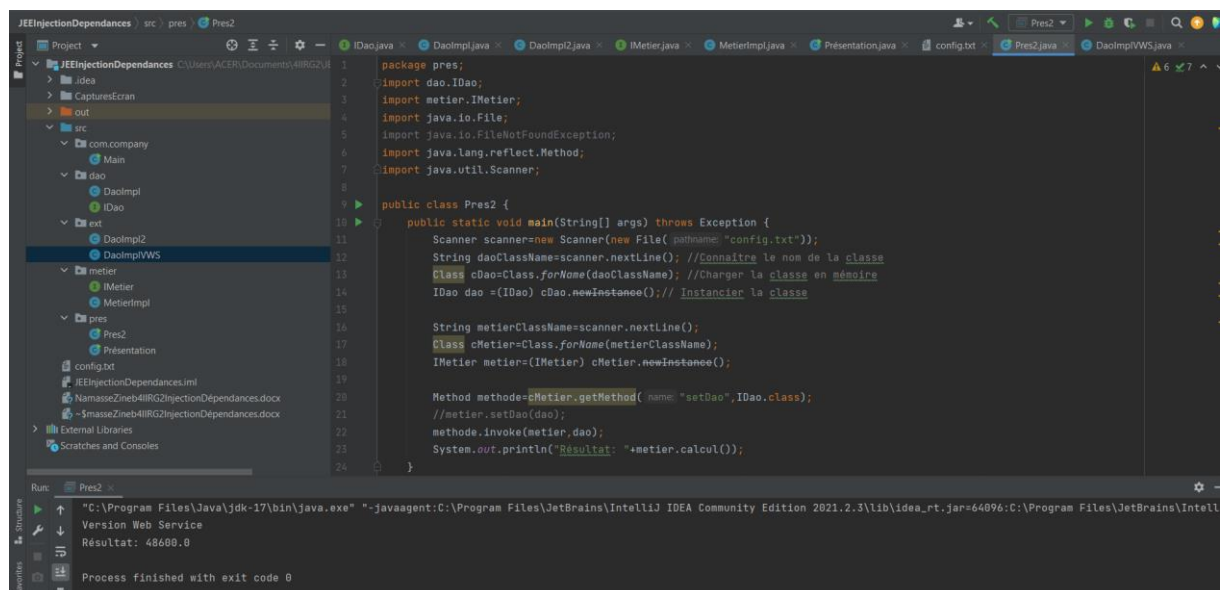
```
1 ext.DaoImplVWS
2 metier.MetierImpl
```

Run: Pres2

"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\lib\idea_rt.jar=64096:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\bin" -jar C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\bin\idea_rt.jar 64096

Version Web Service
Résultat: 48600.0

Process finished with exit code 0



```
1 package pres;
2 import dao.IDao;
3 import metier.IMetier;
4 import java.io.File;
5 import java.io.FileNotFoundException;
6 import java.lang.reflect.Method;
7 import java.util.Scanner;
8
9 public class Pres2 {
10     public static void main(String[] args) throws Exception {
11         Scanner scanner = new Scanner(new File("config.txt"));
12         String daoClassName = scanner.nextLine(); //Connaitre le nom de la classe
13         Class cDao = Class.forName(daoClassName); //Changer la classe en mémoire
14         IDao dao = (IDao) cDao.newInstance(); // Instancier la classe
15
16         String metierClassName = scanner.nextLine();
17         Class cMetier = Class.forName(metierClassName);
18         IMetier metier = (IMetier) cMetier.newInstance();
19
20         Method methode = cMetier.getMethod("setDao", IDao.class);
21         methode.invoke(metier, dao);
22         methode = cMetier.getMethod("calcul");
23         System.out.println("Résultat: " + metier.calcul());
24     }
25 }
```

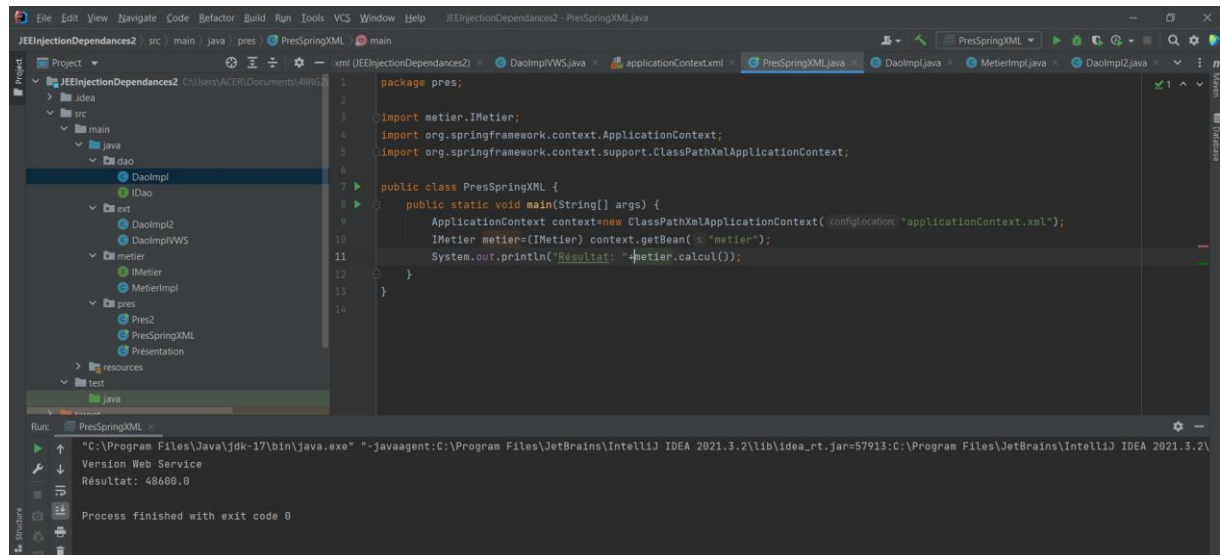
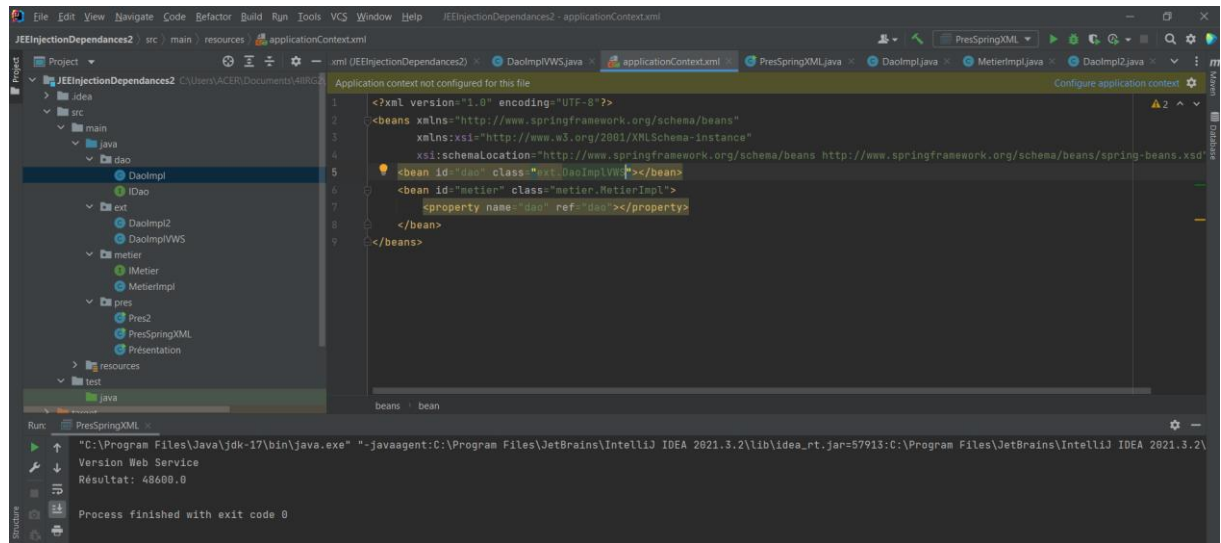
Run: Pres2

"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\lib\idea_rt.jar=64096:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\bin" -jar C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\bin\idea_rt.jar 64096

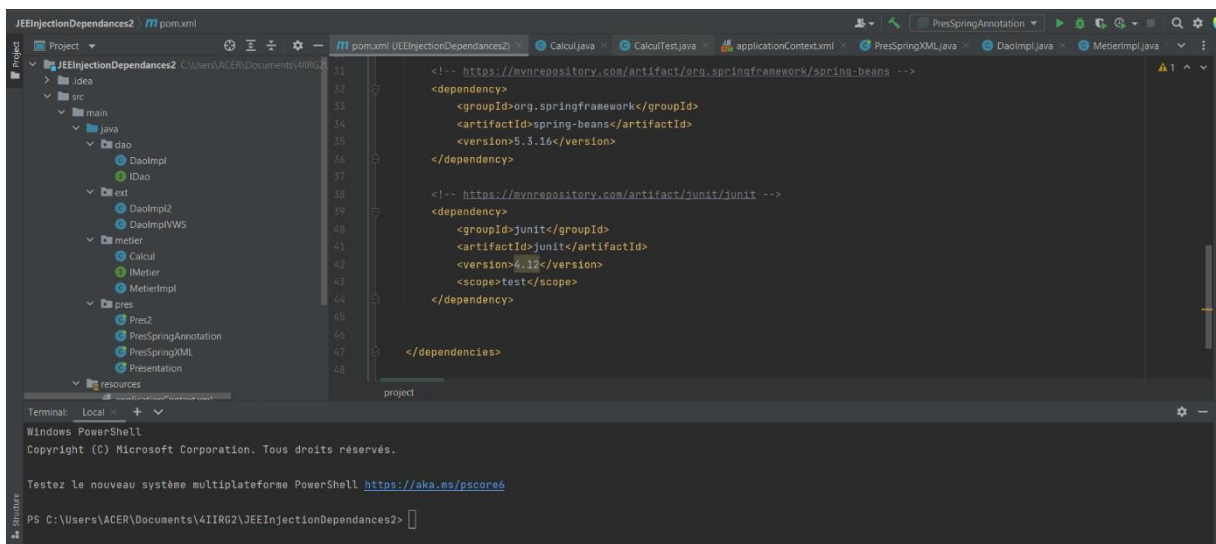
Version Web Service
Résultat: 48600.0

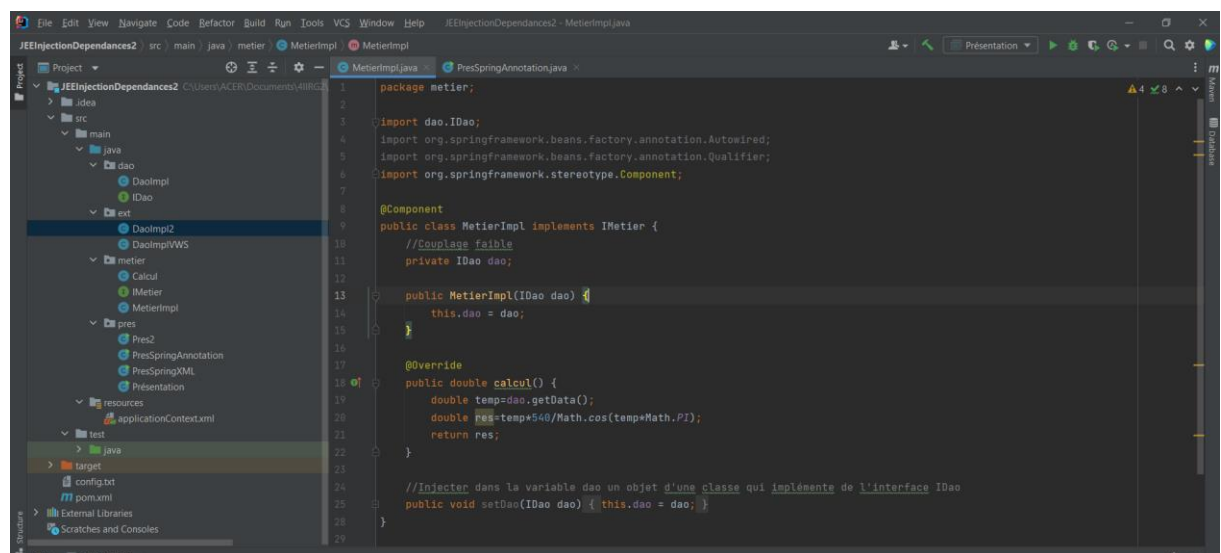
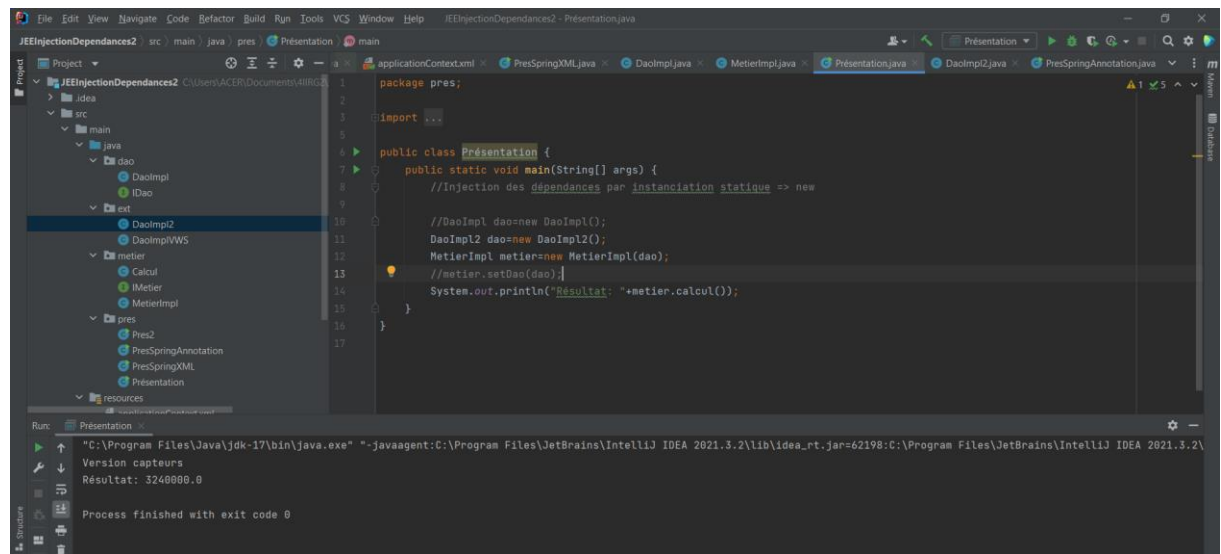
Process finished with exit code 0

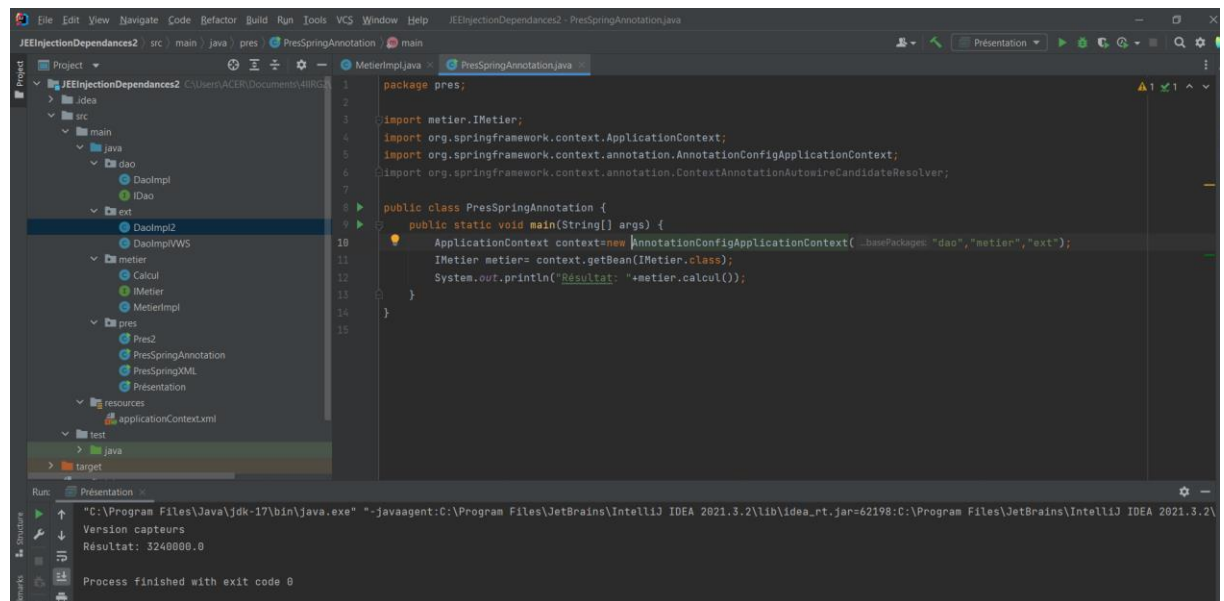
- 3) Spring :
a) Version XML :



- b) Version Annotation :







V) Conclusion :

En conclusion, on peut déduire que parmi les méthodes d'injection des dépendances, c'est la méthode Spring qui facilite le plus la vie du développeur.