



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**
Membre de
HONORIS UNITED UNIVERSITIES

Nom : LAARIBI LAHLOU

Prénom : YASSINE

Niveau : 4ème Année

Filière : Ingénierie Informatique Et Réseaux

Classe : G1/G12

**Site : Emsi Roudani / MAARIF - BIR
ANZARANE CASABLANCA**

Matricule : 244119

Compte Rendu D'Activité Pratique : *Mise en œuvre de l'injection des dépendances*

1- Introduction :

Dans un projet il y a les exigences fonctionnelles et les exigences techniques. Dans 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 ouverte 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).

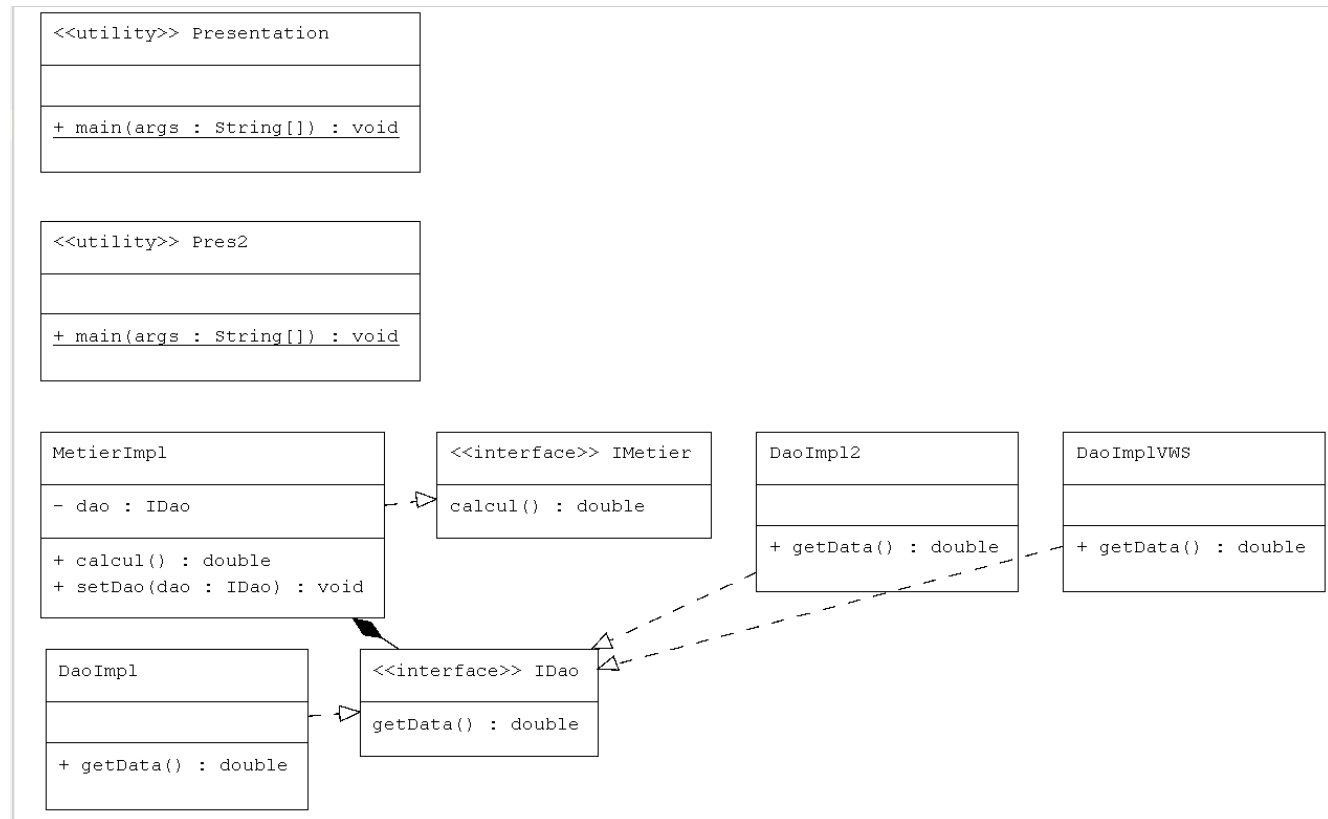
2- Enoncé :

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 :

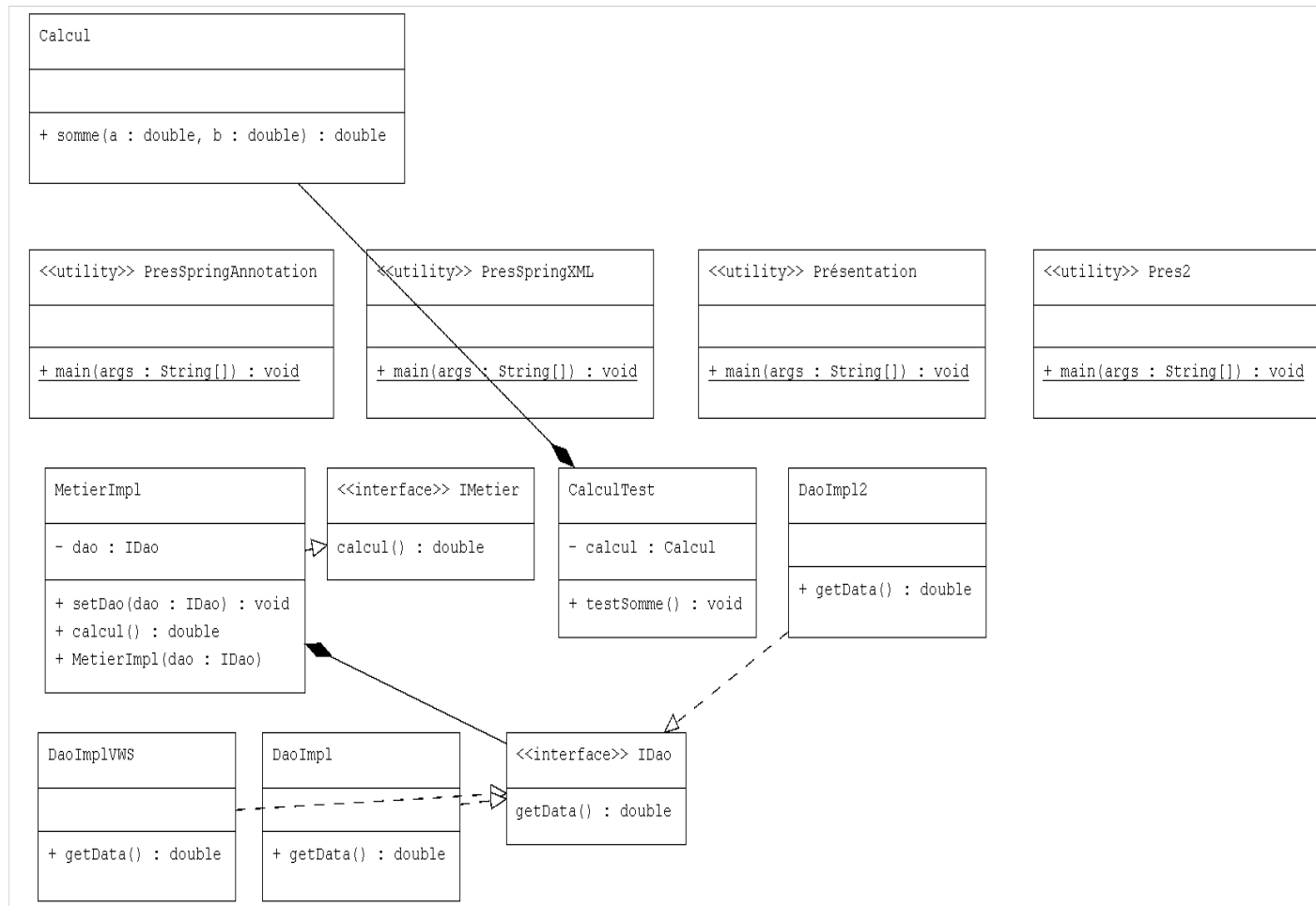
- 1) Créer l'interface IDAO
- 2) Créer une implémentation de cette interface IDAO
- 3) Créer l'interface IMétier
- 4) Créer une implémentation de l'interface IMétier
- 5) Créer la couche Présentation en faisant l'injection des dépendances
 - a) Par implémentation statique
 - b) Par implémentation dynamique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

3- Conception et Architecture :

Part 1 :



Part 2 :



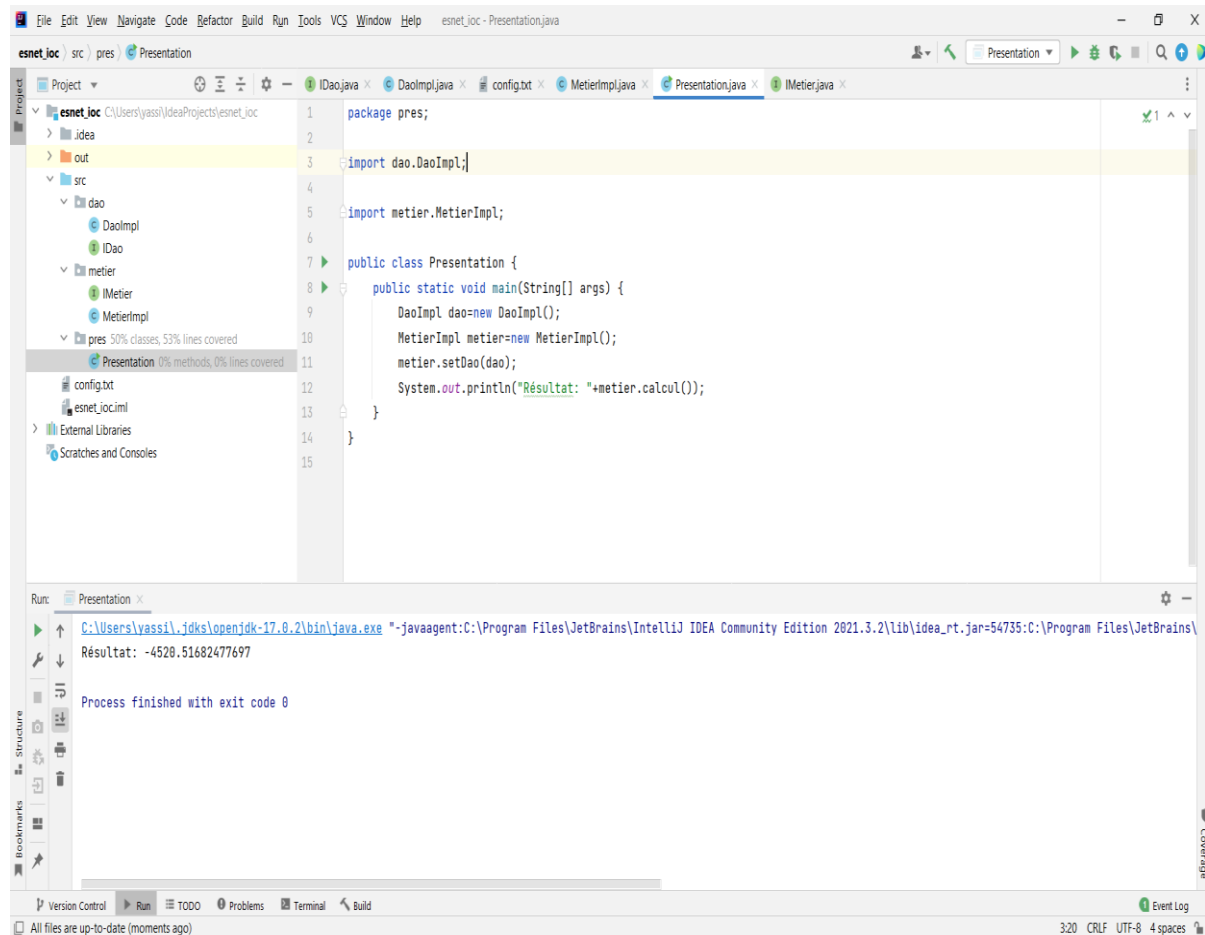
4- Code source :

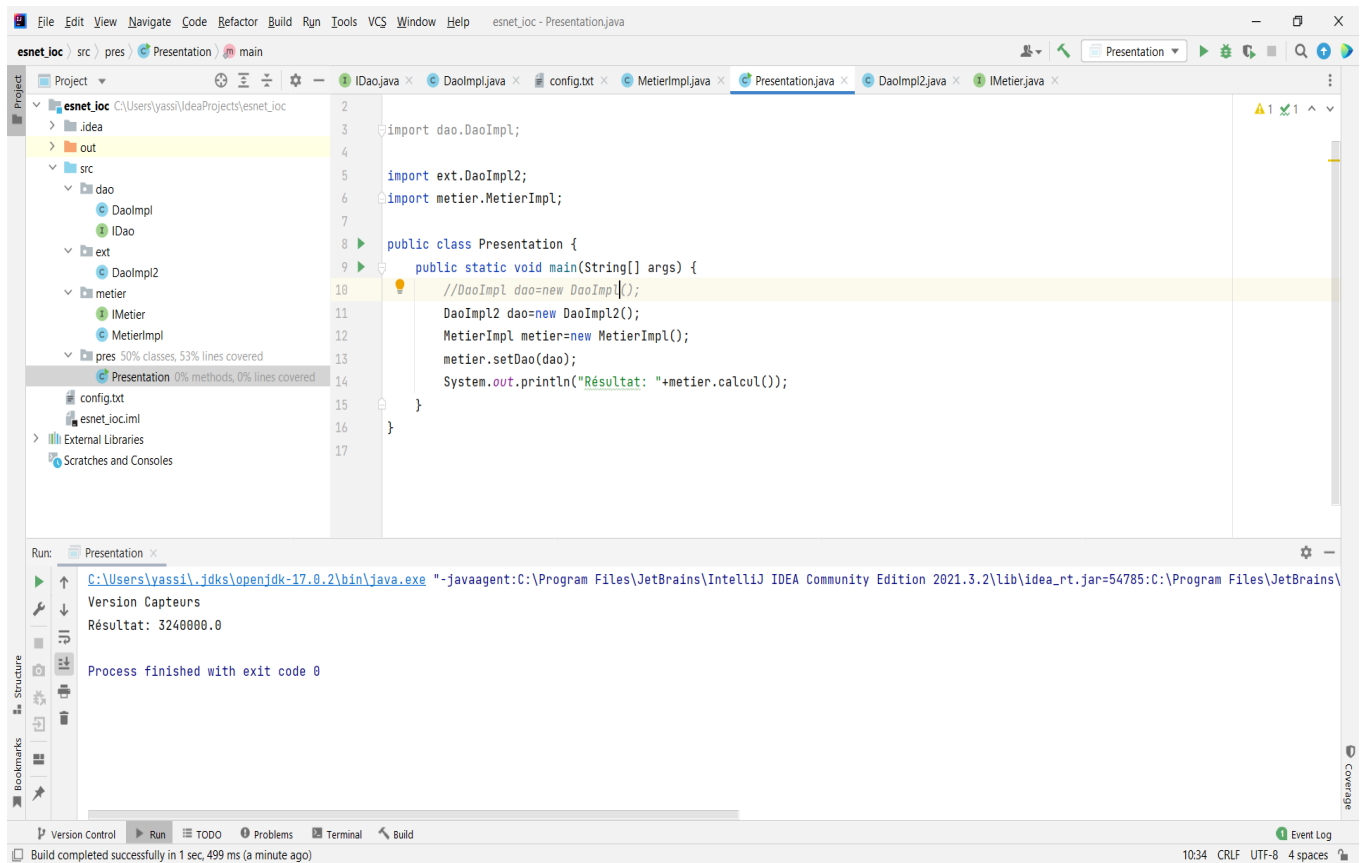
Part 1 :

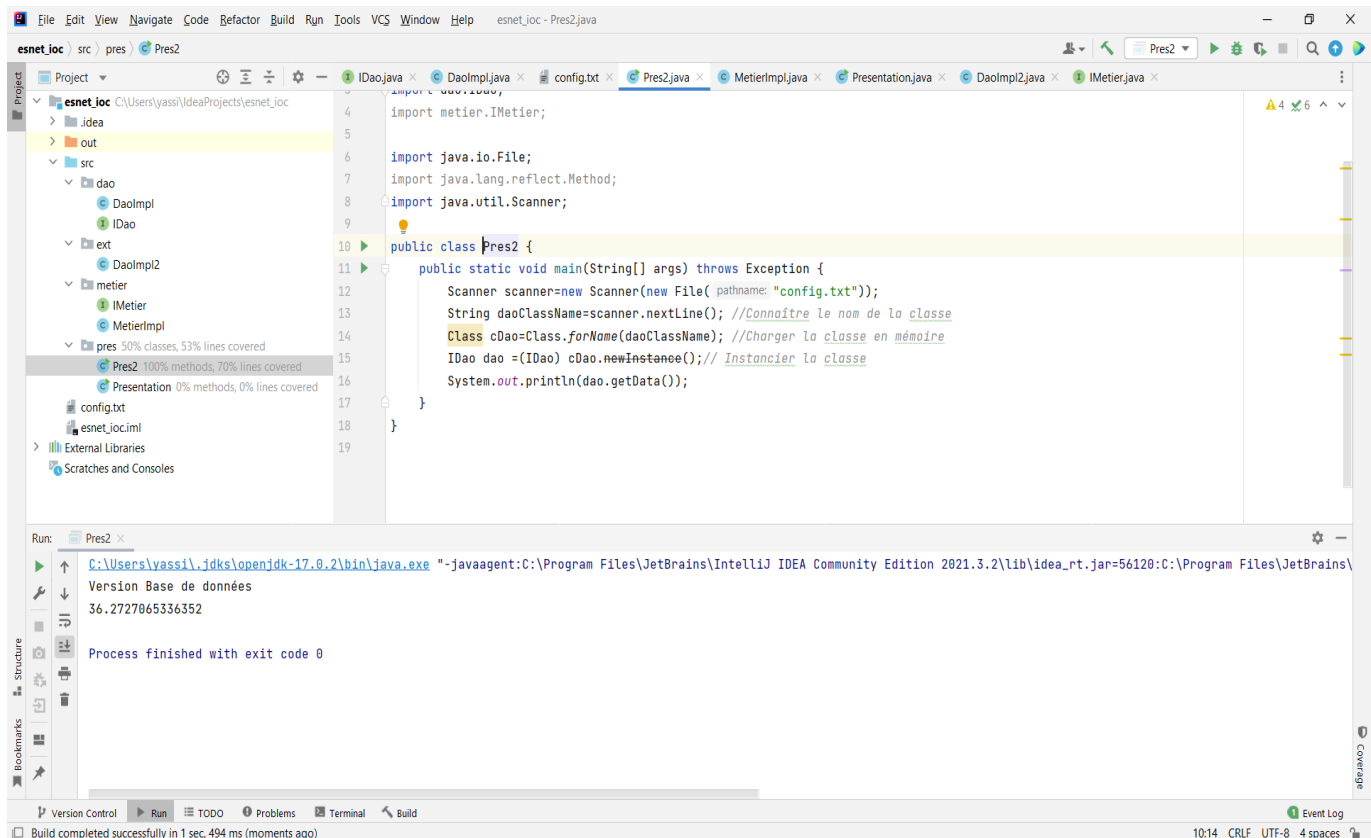
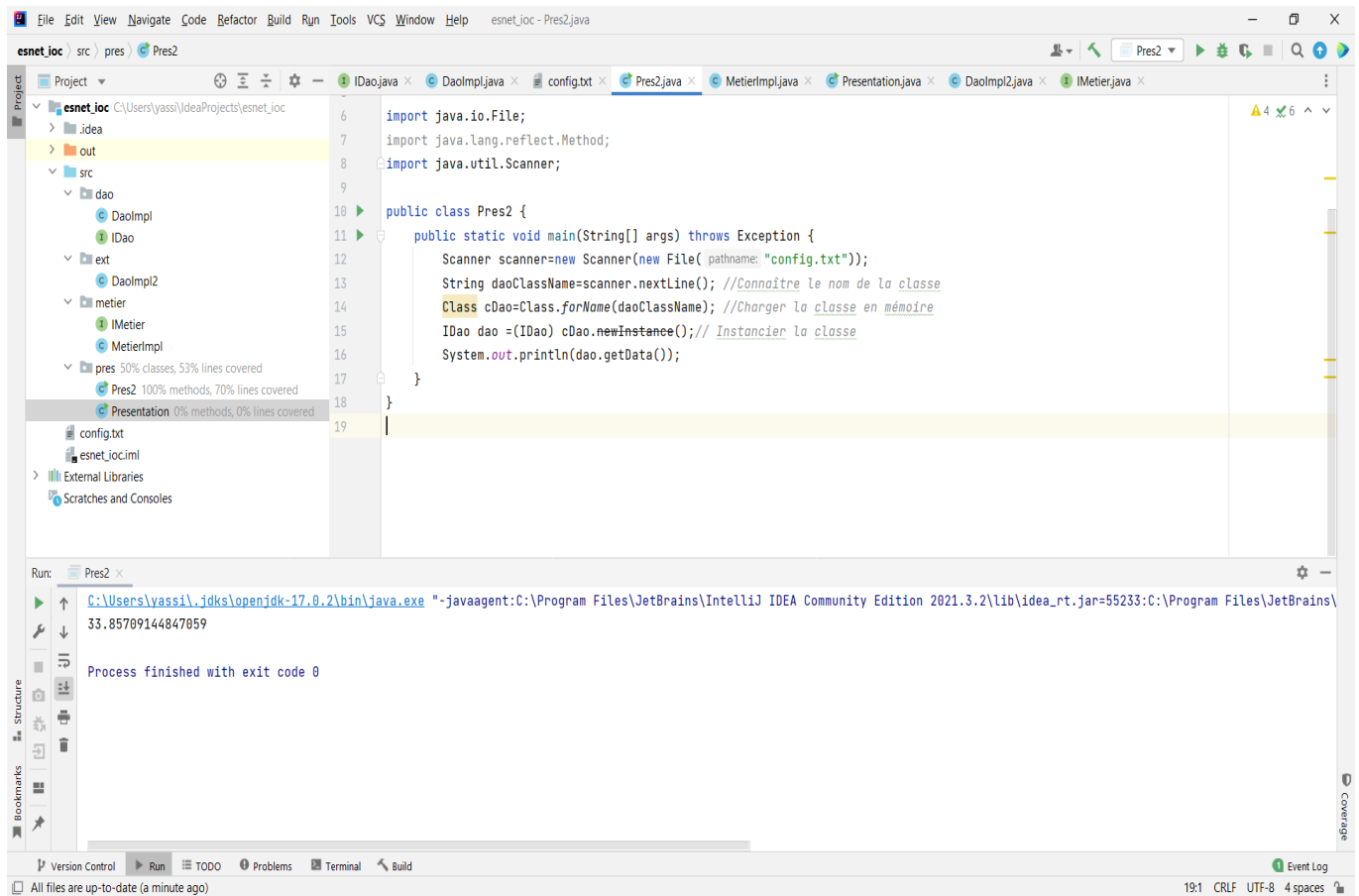
Part 2 :

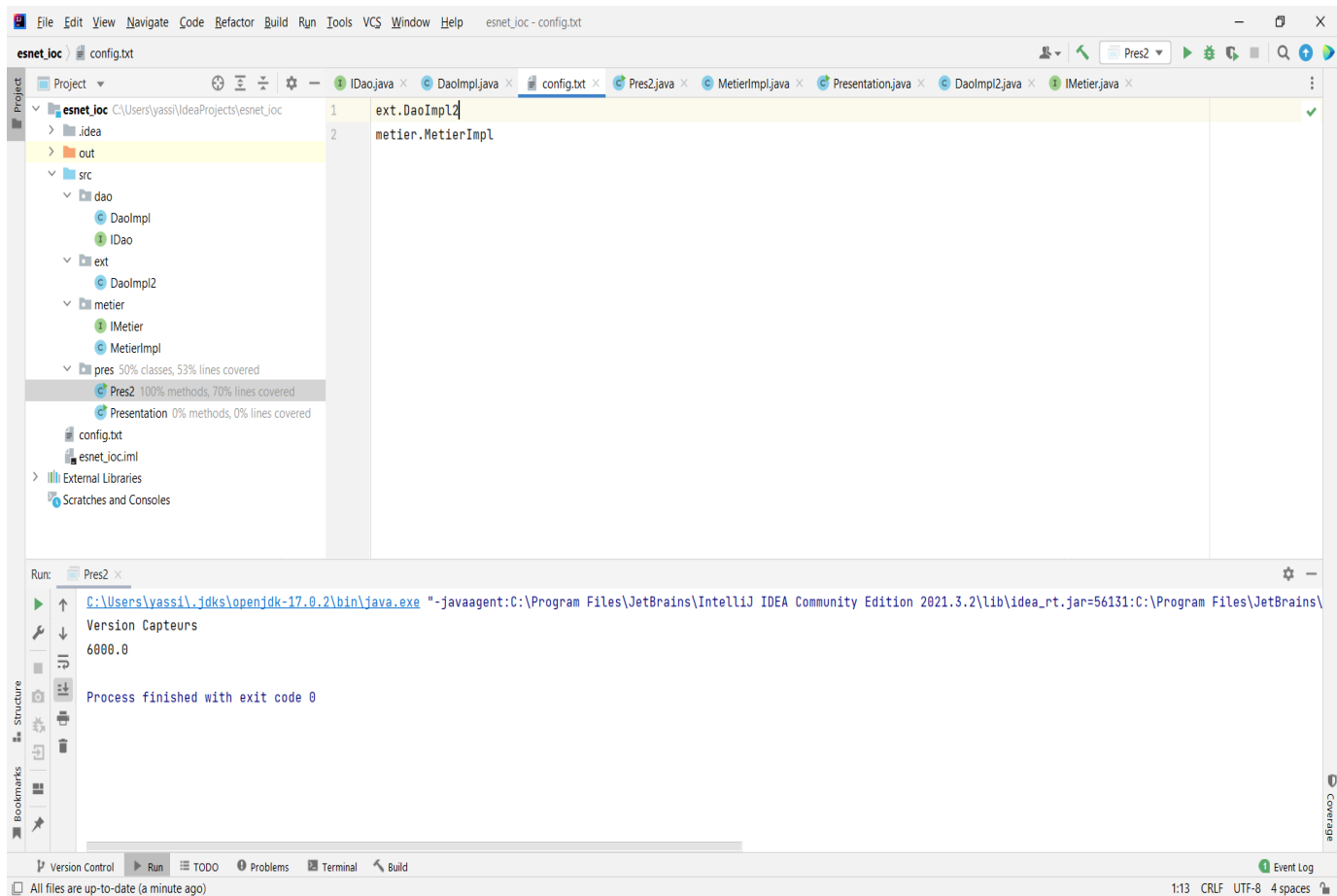
5- Captures écrans :

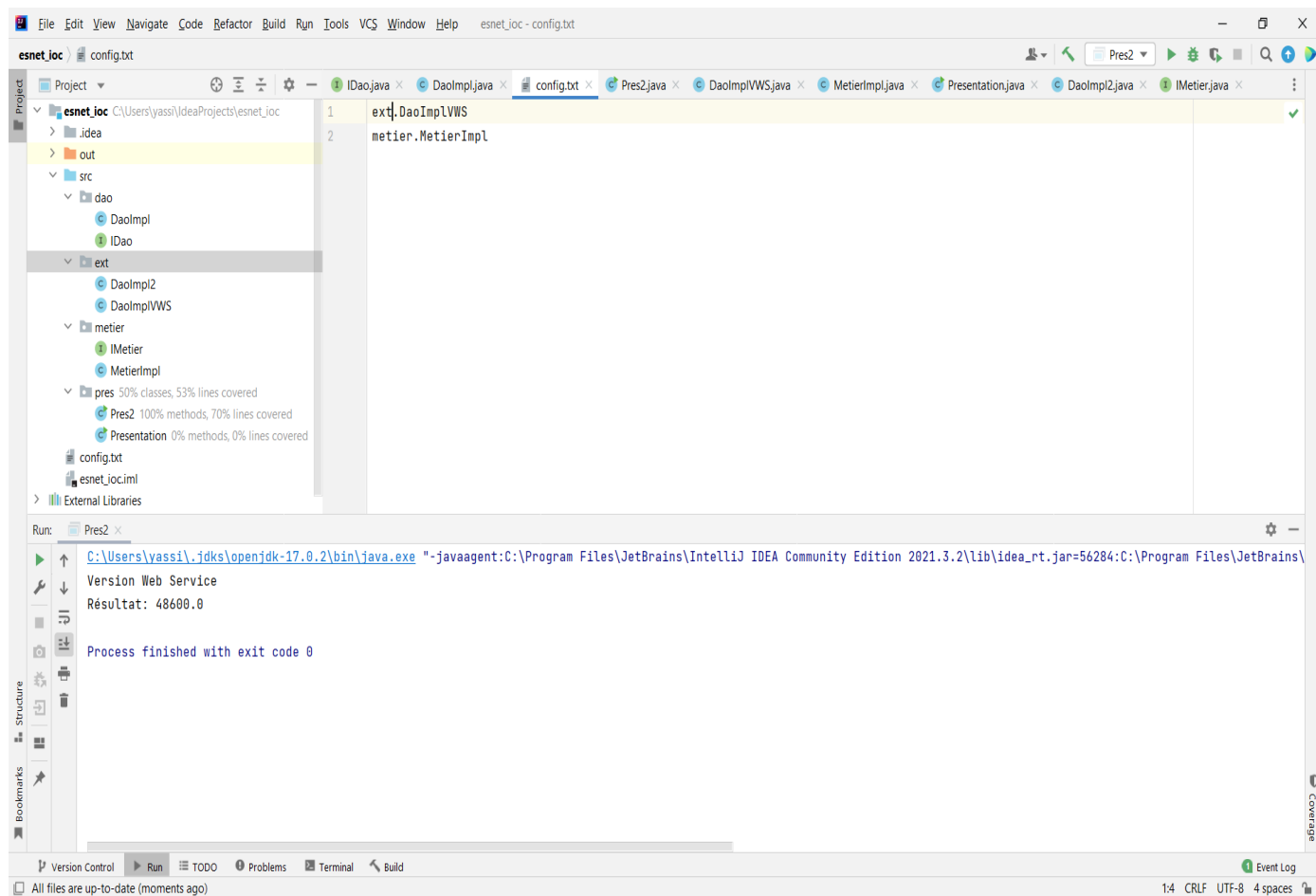
Part 1 :





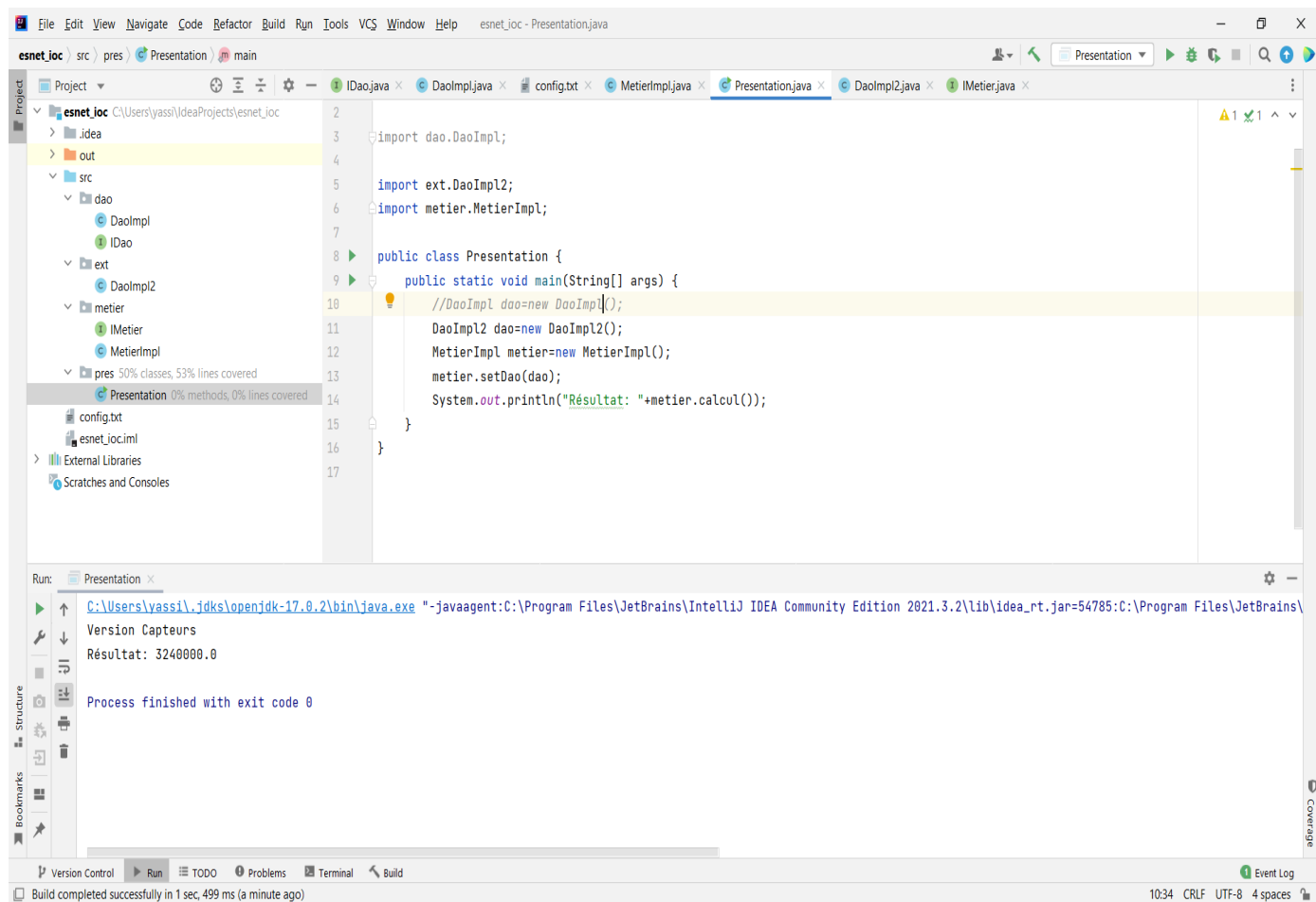




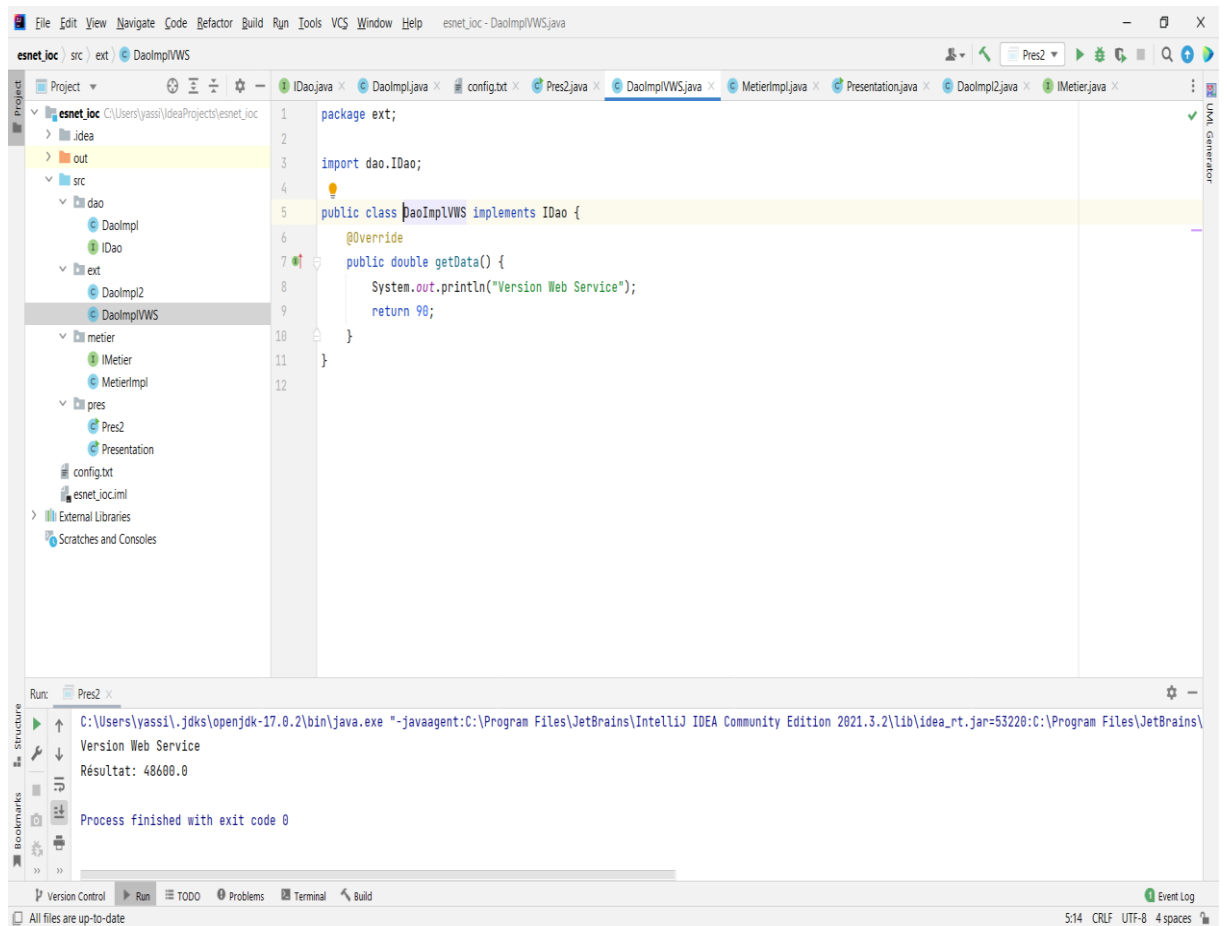


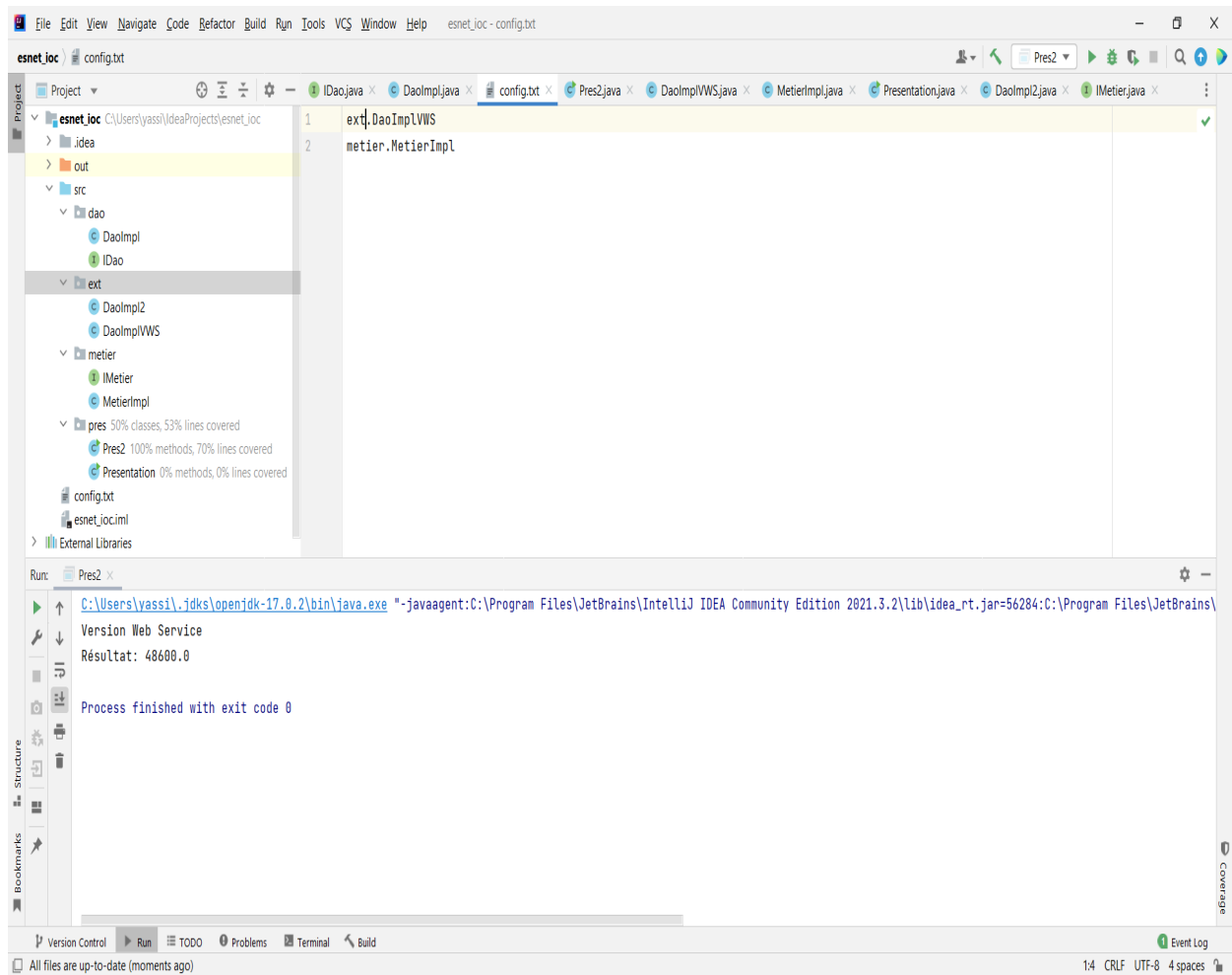
Injection des dépendances avec :

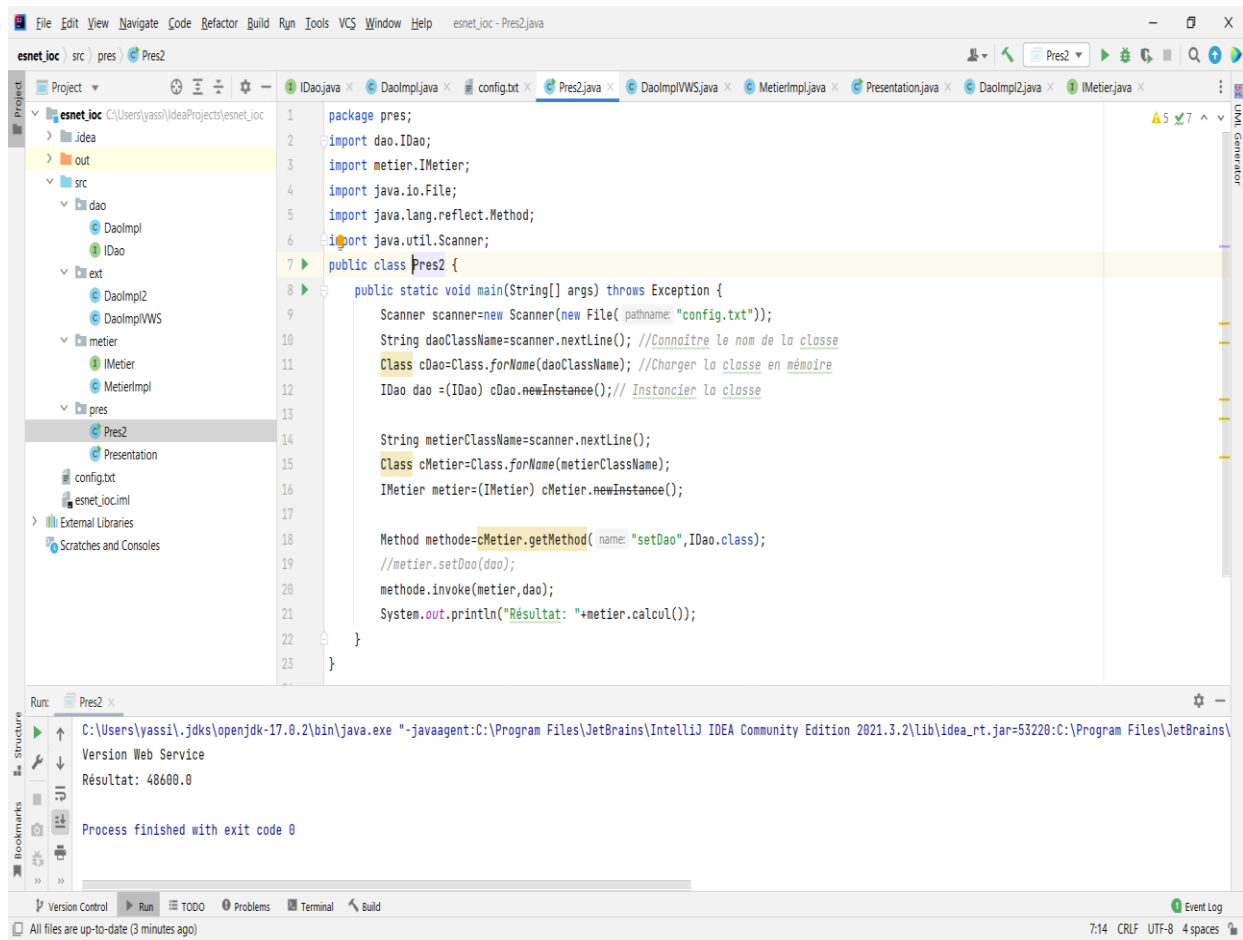
1) L'instanciation statique :



2) L'instanciation dynamique :



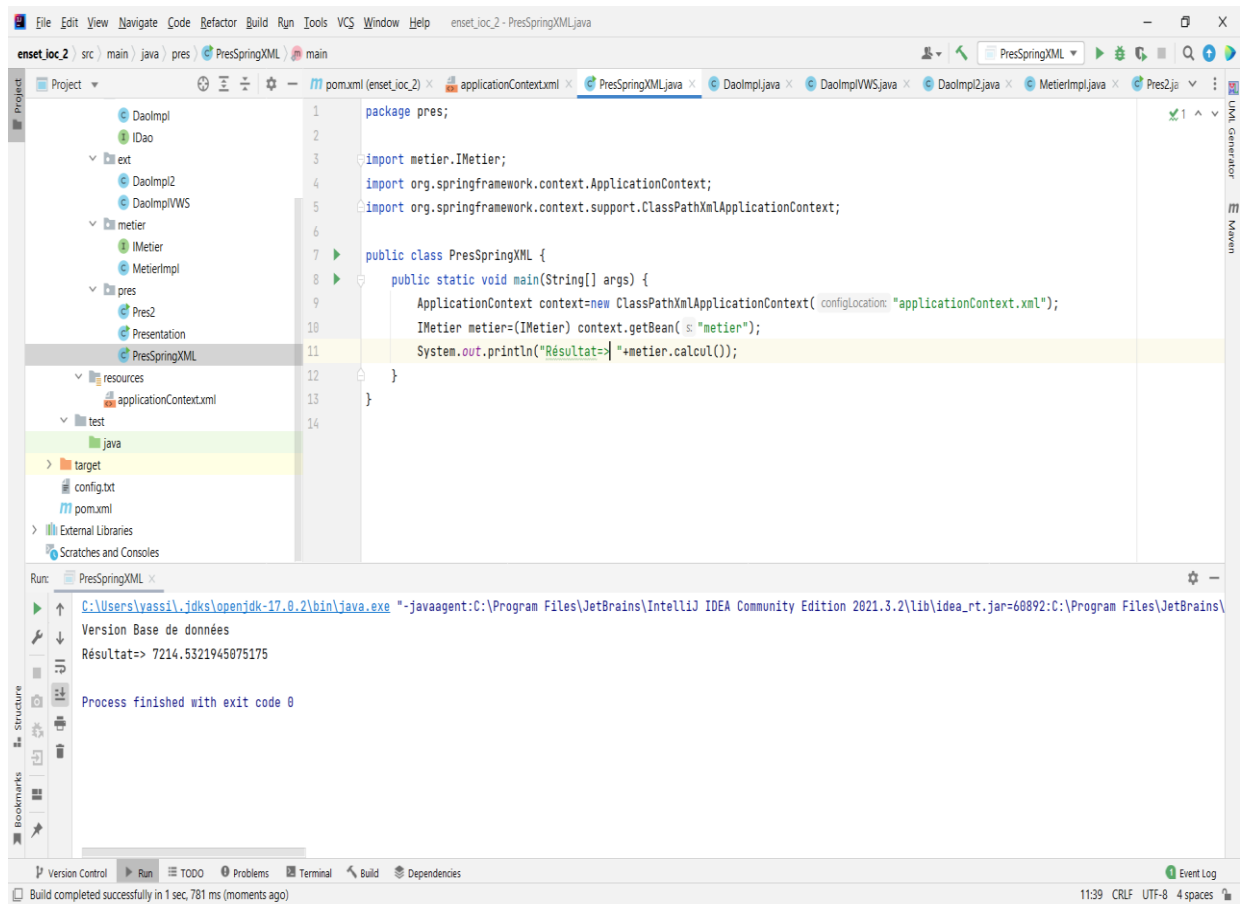


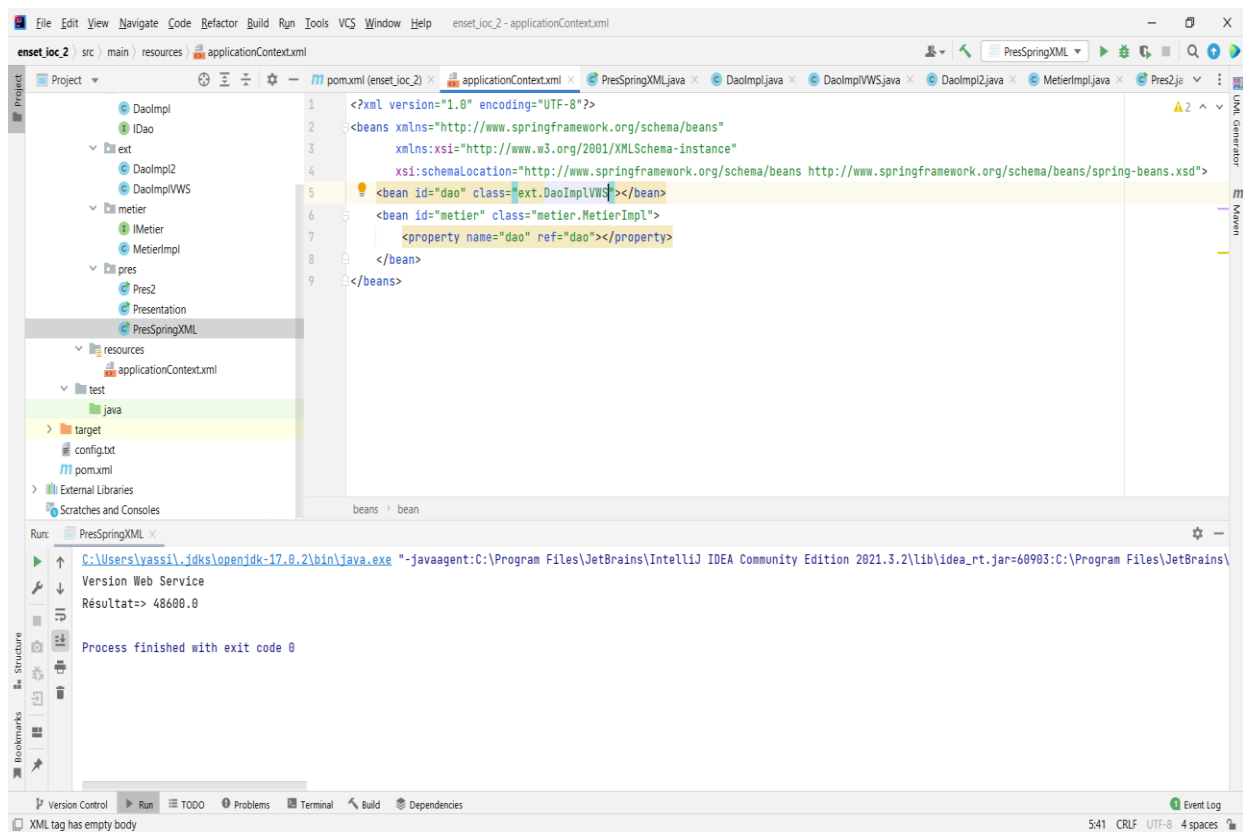
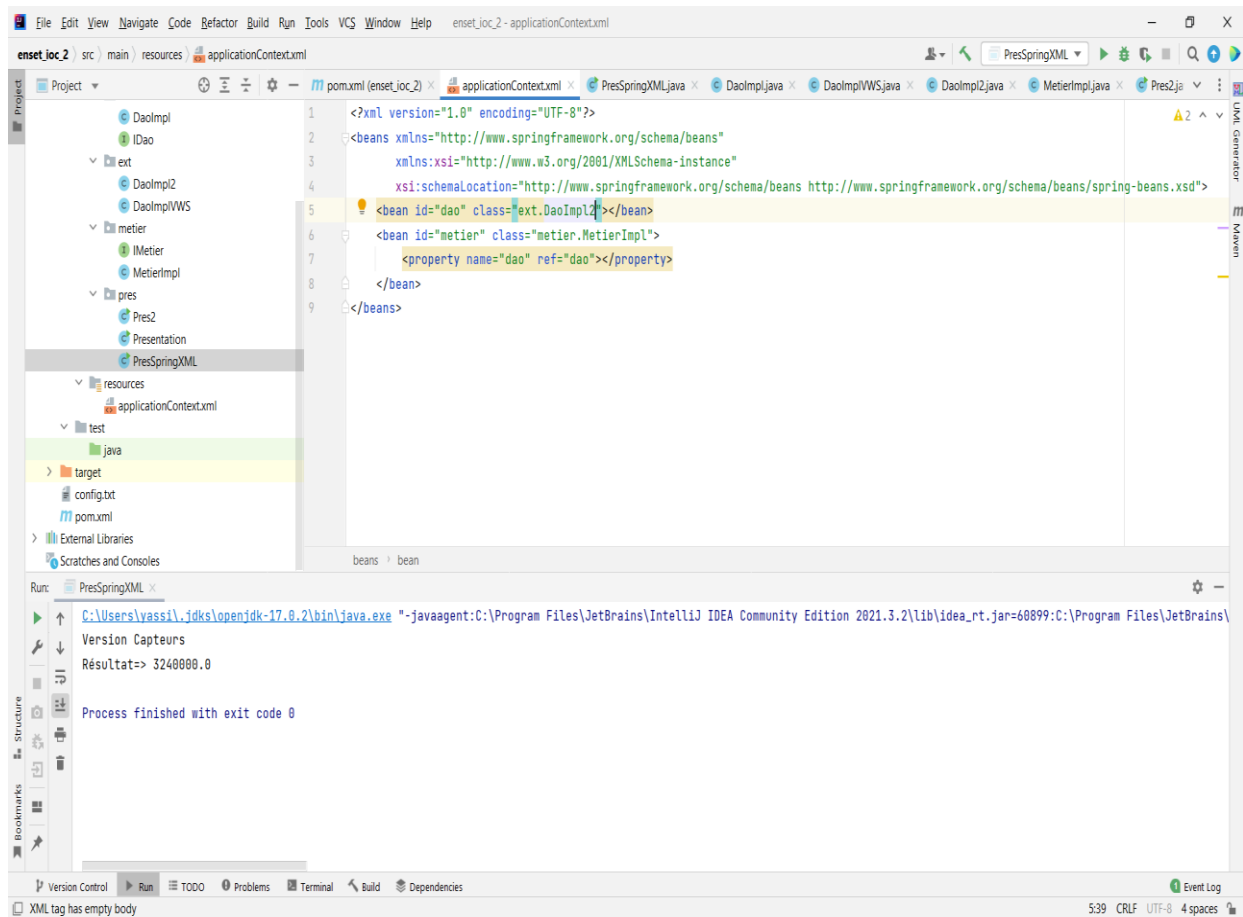


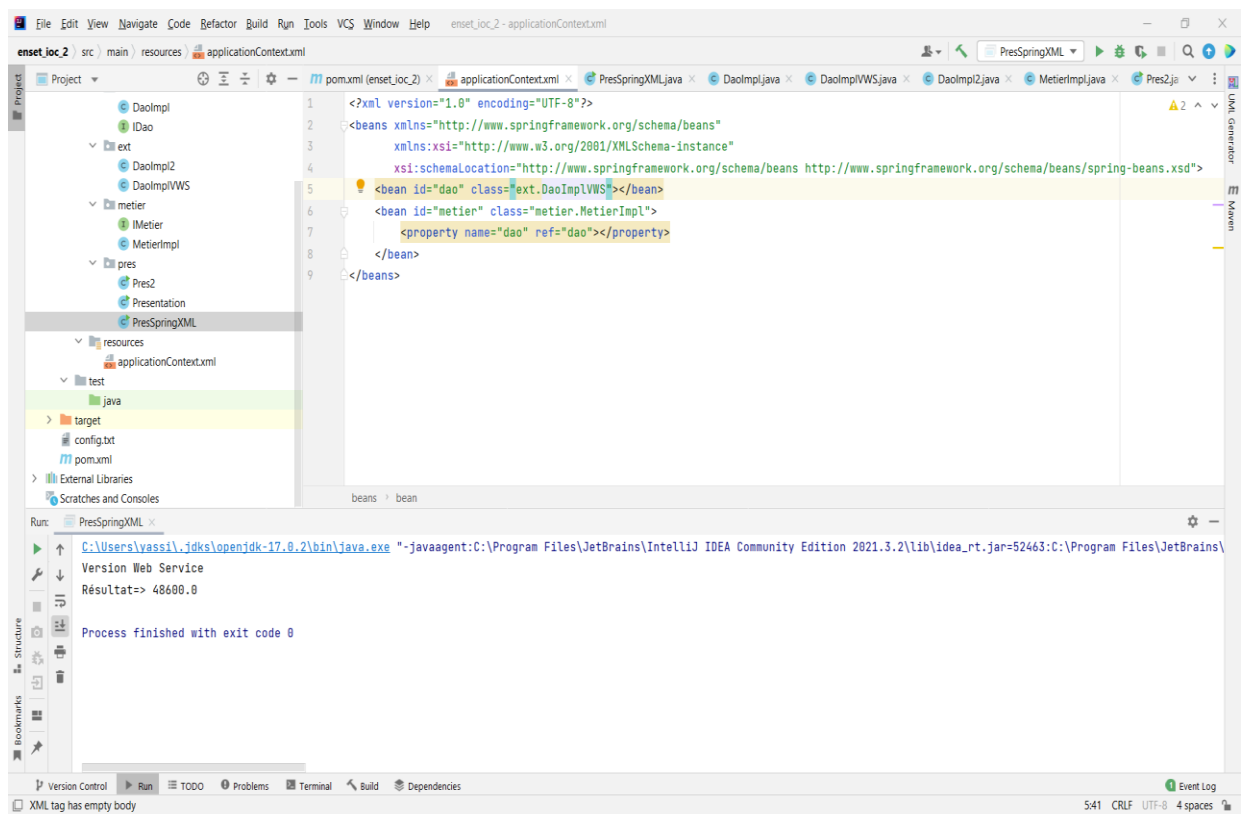
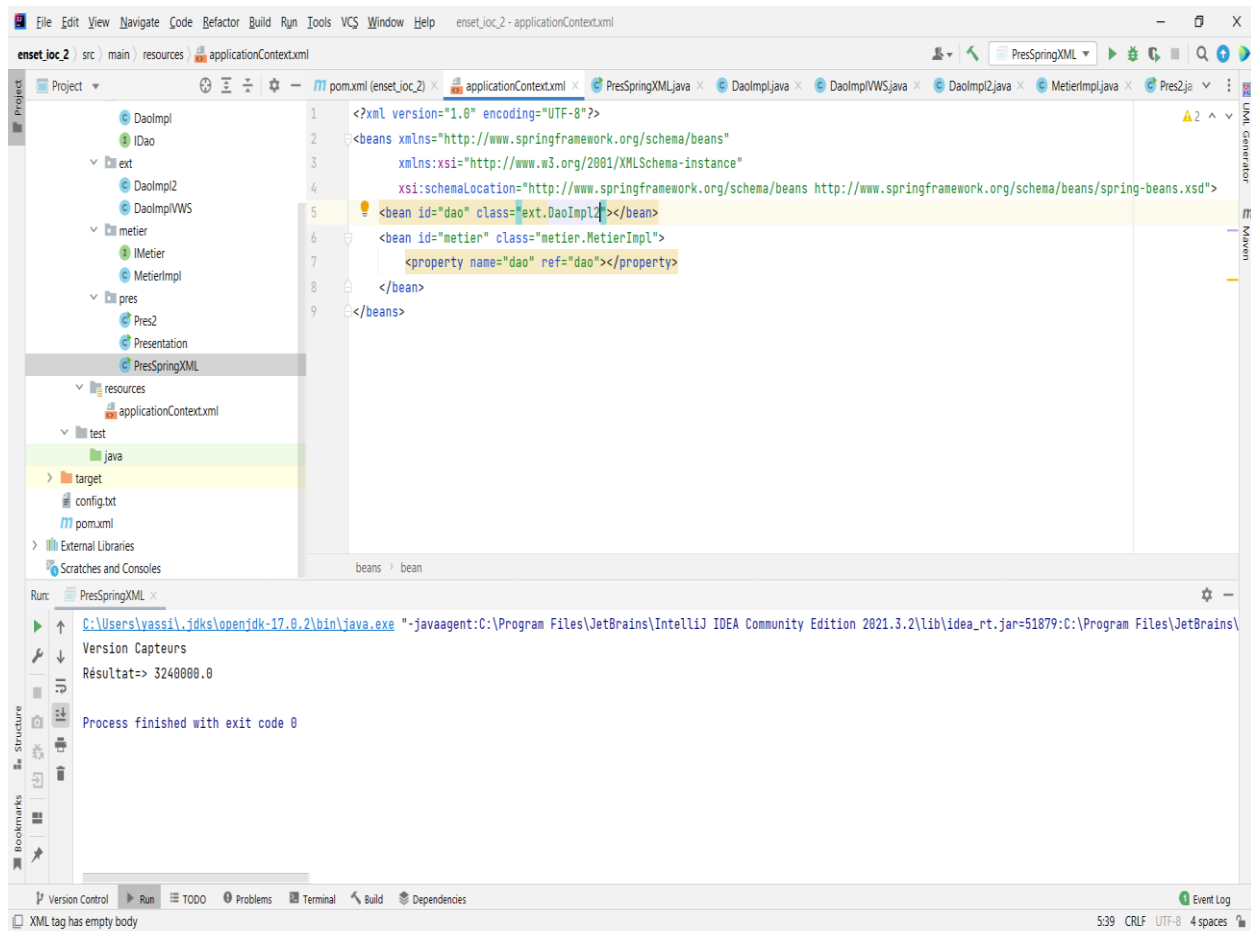
Part 2 :

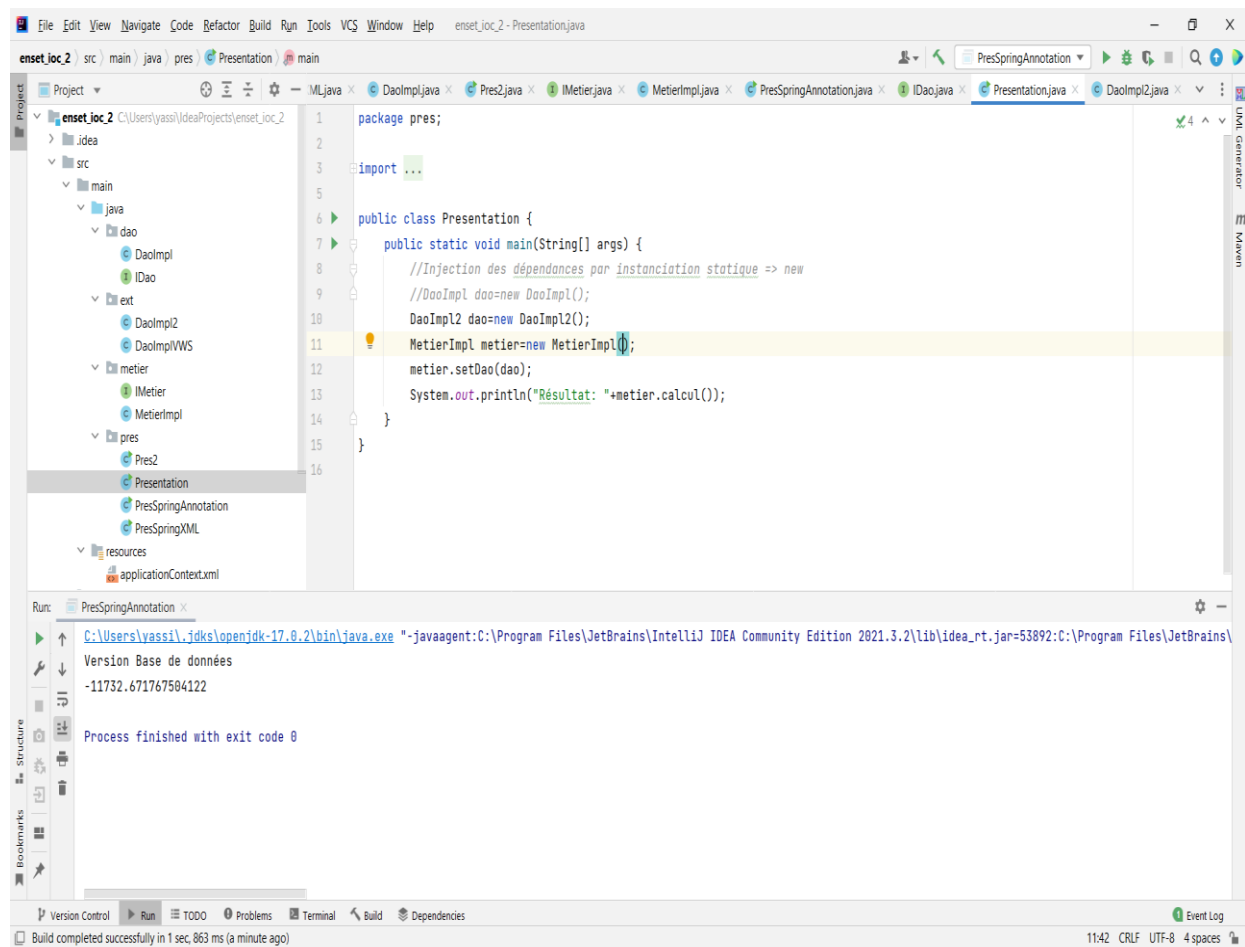
3) Spring :

a) Version XML :









b) Version Annotation :

Git Window Help JEE-master - pom.xml (JEEInjectionDependencies2)

Pres2

Daolmpl2.java x DaolmplVWS.java x MetierImpl.java x Pres2.java x pom.xml (JEEInjectionDependencies2) x PresSpringAnnotation.java x PresSpringXML

```
24
25
26 <dependency>
27   <groupId>org.springframework</groupId>
28   <artifactId>spring-context</artifactId>
29   <version>5.3.16</version>
30 </dependency>
31
32 <!-- https://mvnrepository.com/artifact/org.springframework/spring-beans -->
33 <dependency>
34   <groupId>org.springframework</groupId>
35   <artifactId>spring-beans</artifactId>
36   <version>5.3.16</version>
37 </dependency>
38
39 <!-- https://mvnrepository.com/artifact/junit/junit -->
40 <dependency>
41   <groupId>junit</groupId>
42   <artifactId>junit</artifactId>
43   <version>4.12</version>
44   <scope>test</scope>
45 </dependency>
46
47 </dependencies>
48
49 </project>
```

project

Dependencies

Event Log

49:11 LF UTF-8 4 spaces

Git Window Help JEE-master - Daolmpl2.java

Pres2

Daolmpl2.java x DaolmplVWS.java x MetierImpl.java x Pres2.java x pom.xml (JEEInjectionDependencies2) x PresSpringAnnotation.java x PresSpringXML

```
1 package ext;
2
3 import ...
4
5
6 @Component("dao2")
7 public class DaoImpl2 implements IDao {
8   @Override
9   public double getData() {
10     System.out.println("Version capteurs");
11     double temp=6000;
12     return temp;
13   }
14 }
15
```

Dependencies

Event Log

7:14 LF UTF-8 4 spaces

Git Window Help JEE-master - Présentation.java

Présentation

UML Generator Maven

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

Run: Présentation

C:\Users\yassi\jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\idea_rt.jar" C:\Users\yassi\workspace\JEE-master\src\main\java\pres\Presentation.java

Version capteurs

Résultat: 3240000.0

Process finished with exit code 0

Git Window Help JEE-master - MetierImpl.java

Présentation

UML Generator Maven

```
1 package metier;
2
3 import ...
4
5
6 @Component
7
8 public class MetierImpl implements IMetier {
9     //Couplage faible
10    private IDao dao;
11
12    public MetierImpl(IDao dao) { this.dao = dao; }
13
14
15    @Override
16    public double calcul() {
17        double temp=dao.getData();
18        double res=temp*540/Math.cos(temp*Math.PI);
19        return res;
20    }
21
22    //Injecter dans la variable dao un objet d'une classe qui implémente de l'interface IDao
23    public void setDao(IDao dao) { this.dao = dao; }
24
25 }
26
27
28
29
```

dependencies

Event Log

9:14 LF UTF-8 4 spaces

```
1 package pres;
2
3 import ...
4
5
6
7
8 public class PresSpringAnnotation {
9     public static void main(String[] args) {
10         ApplicationContext context=new AnnotationConfigApplicationContext( ...basePackages: "dao","metier","ext");
11         IMetier metier= context.getBean(IMetier.class);
12         System.out.println("Résultat: "+metier.calcul());
13     }
14 }
15
```

Run: Présention x

C:\Users\yassi\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\lib\...
Version capteurs
Résultat: 3240000.0
Process finished with exit code 0

6- 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 ».