

COMPTE RENDU JEE

ACTIVITÉ PRATIQUE

INVERSION DE CONTRÔLE ET INJECTION DES DÉPENDANCES

ENCADRÉ PAR : MR MOHAMED YOUSFI

PRÉSENTÉ PAR : BENNABBOU WIAM



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

PLAN:

- Introduction
- Enoncé
- Conception
- Code Source
- Exécution
- Conclusion

INTRODUCTION:

- Dans un projet , nous trouvons des **exigences fonctionnelles** et des **exigences techniques**. Parmi les exigences techniques qu'il faut prendre en considération on trouve la **maintenance**.
- Pour créer une application facile à maintenir, elle doit être **fermée à la modification** et **ouverte à l'extension**.
- Pour créer un projet qui respecte ces critères, il faut se baser sur le principe **d'inversion de contrôle** qui permet aux développeurs de se focaliser sur le **code métier** alors que le Framework **Spring** par exemple si on parle de **l'architecture JEE**, qui va s'occuper du **code technique** (sécurité, la gestion des transactions, l'accès aux données...). Ce principe est rendu possible grâce à **la programmation orientée aspect (AOP)**.
- Si on veut créer une application fermée à la modification et ouverte à l'extension, il faut utiliser le **couplage faible** au lieu d'utiliser le couplage fort.

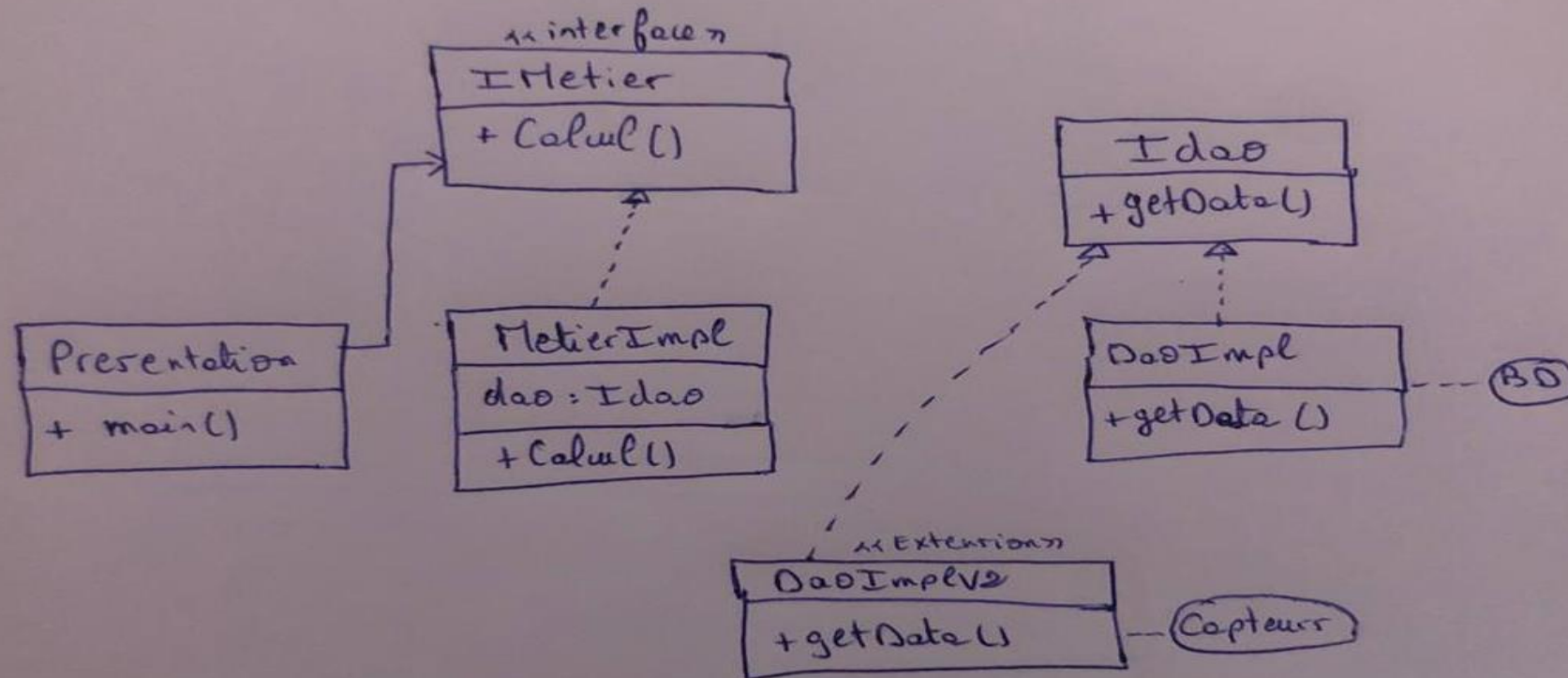


ÉNONCÉ:

ACTIVITÉ PRATIQUE : INVERSION DE CONTRÔLE ET INJECTION DES DÉPENDANCES

1. Créer l'interface `Idao`
2. Créer une implémentation de `Idao`
3. Créer l'interface `Imetier`
4. Créer une implémentation de `Imetier`
5. Créer la couche présentation en faisant l'injection des dépendances:
 - Par instantiation statique
 - Par instantiation dynamique
 - En utilisant Spring Framework : Version XML et Annotations

CONCEPTION:

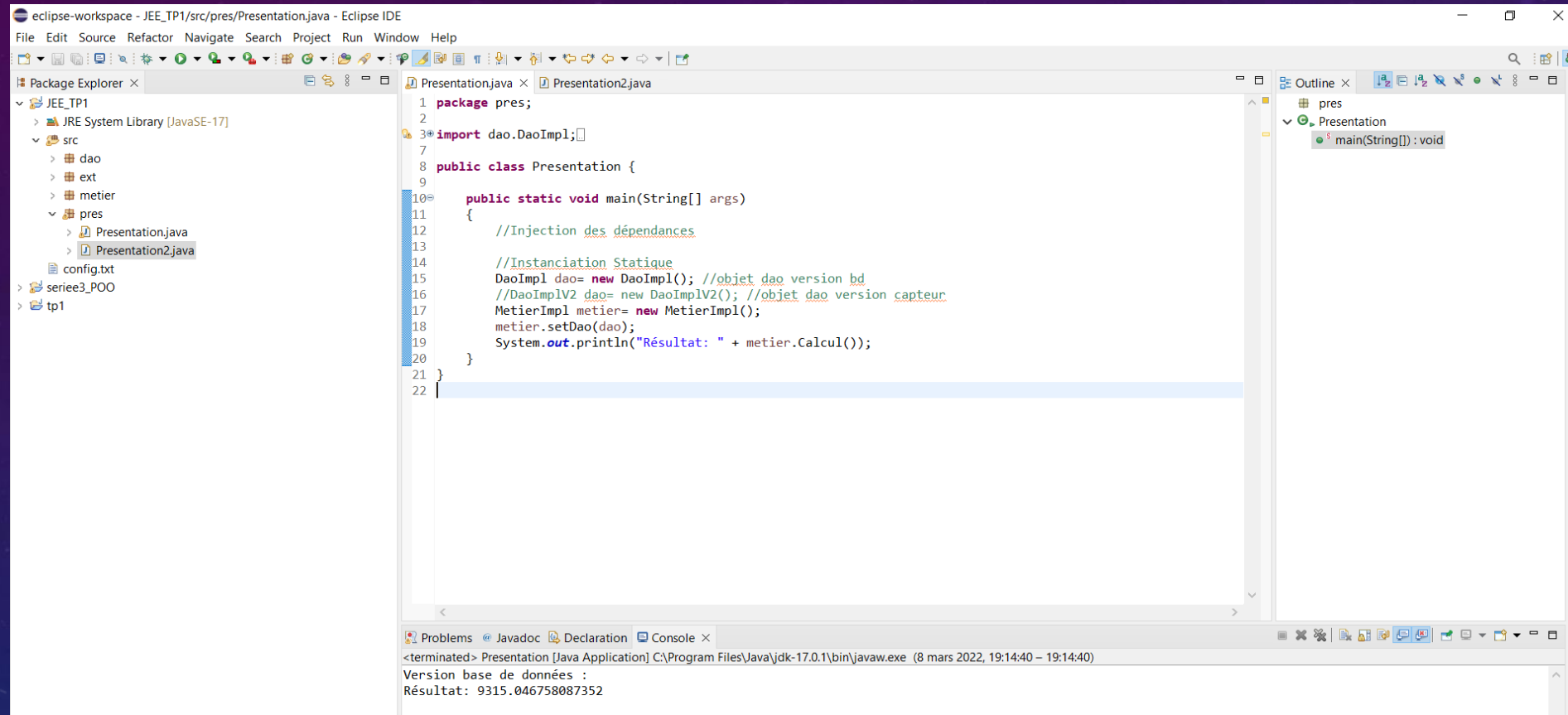


CODE SOURCE:

- Lien GitHub => Instanciation Statique et Dynamique
[bennabbouwiam/JEEProject \(github.com\)](https://github.com/bennabbouwiam/JEEProject)
- Lien GitHub => Version XML et Annotations
[bennabbouwiam/JEEProjectVersionSping \(github.com\)](https://github.com/bennabbouwiam/JEEProjectVersionSping)

EXÉCUTION :

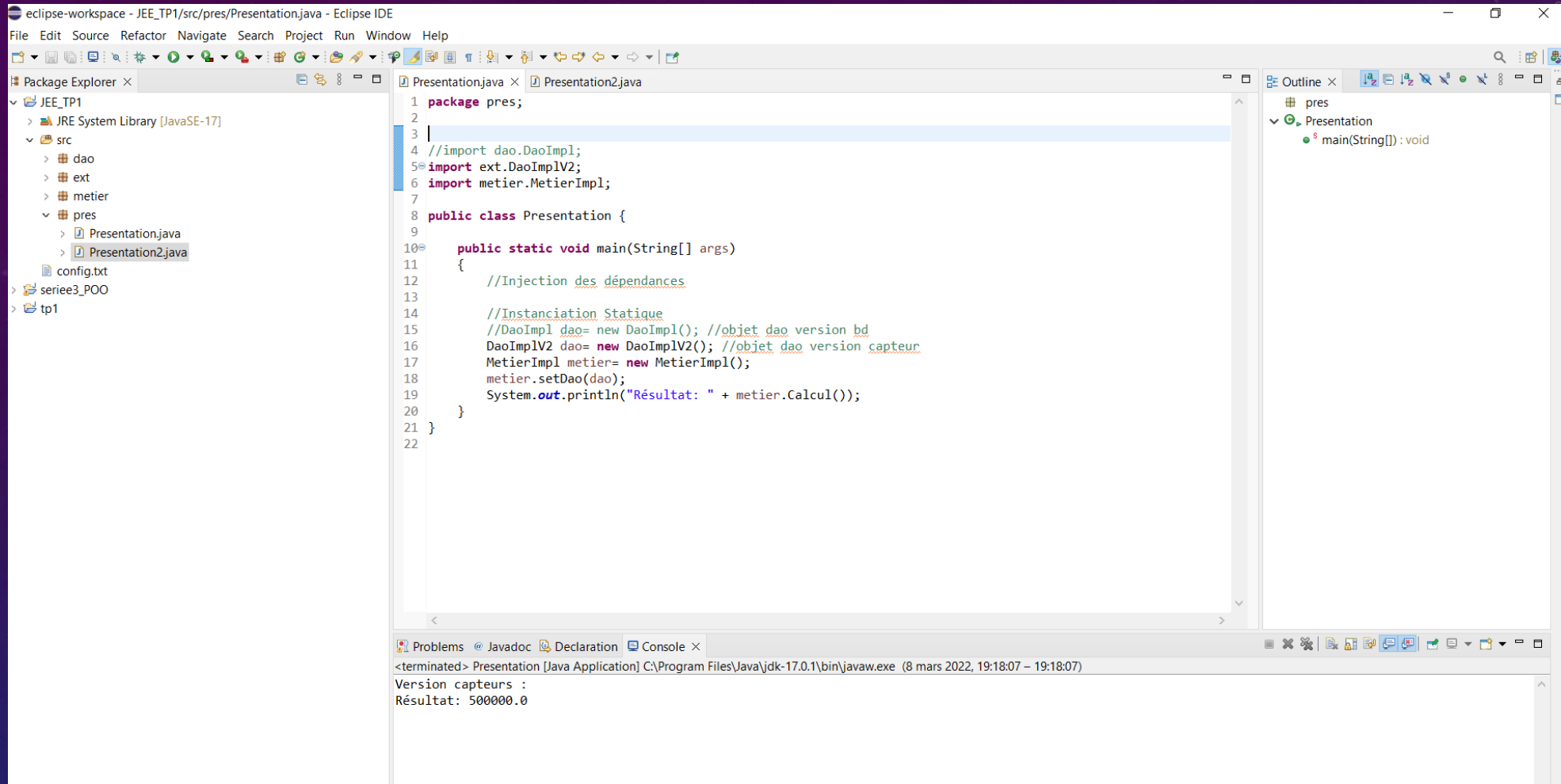
- **Instanciation Statique:**



The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure: JEE_TP1, JRE System Library [JavaSE-17], src, dao, ext, metier, pres, Presentation.java, Presentation2.java, config.txt, seriee3_POO, and tp1. The main editor window shows the code for Presentation.java, which includes a package declaration, an import statement, and a public class with a static main method. The main method performs static instantiation of DaoImpl, DaoImplV2, and MetierImpl, sets the dao on the metier, and prints the result. The Outline view on the right shows the class structure: pres, Presentation, and main(String[]): void. The Console view at the bottom shows the output of the application, indicating that it terminated successfully and displaying the result: 9315.046758087352.

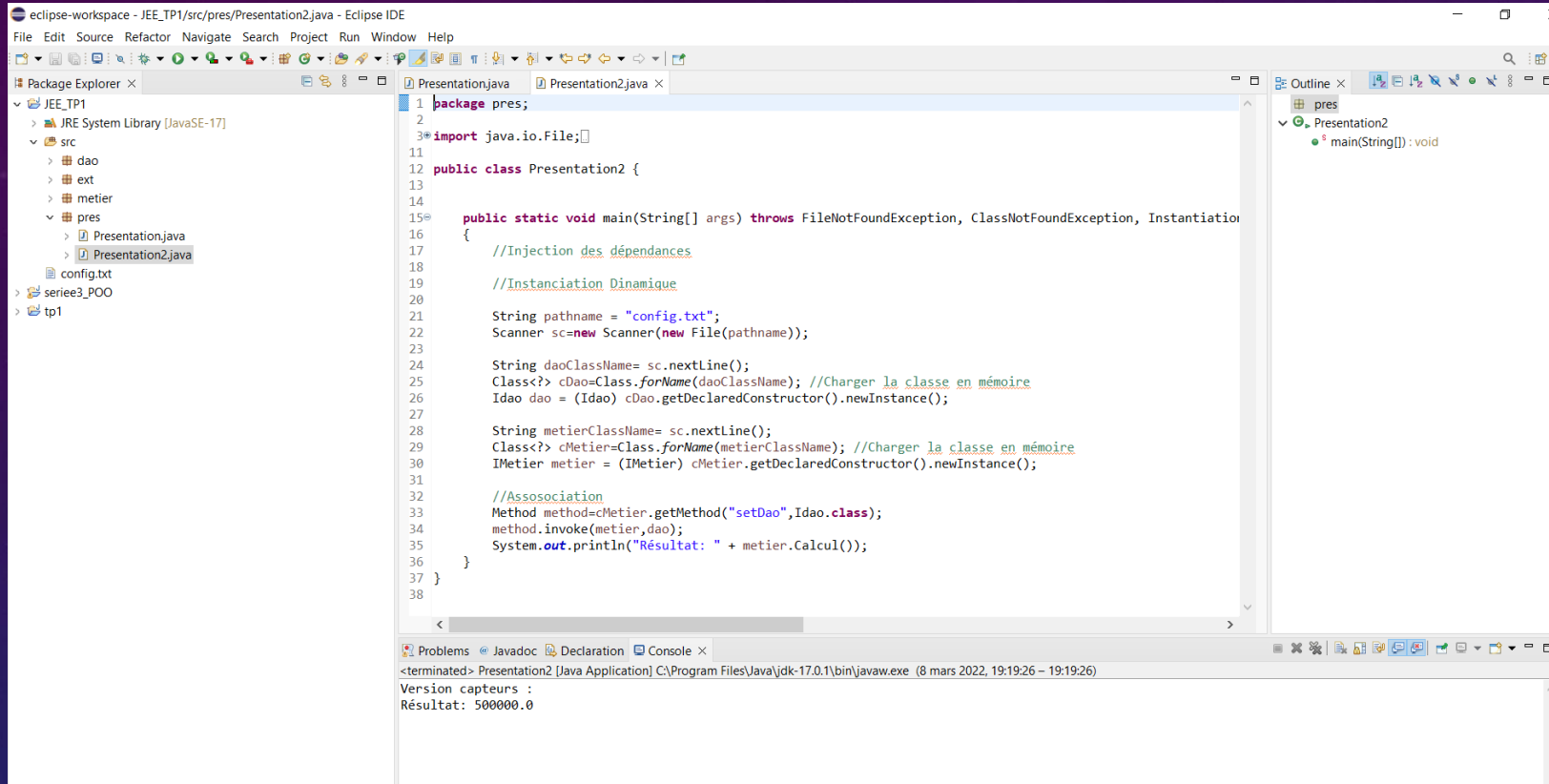
```
1 package pres;
2
3 import dao.DaoImpl;
4
5
6
7
8 public class Presentation {
9
10     public static void main(String[] args)
11     {
12         //Injection des dépendances
13
14         //Instanciation Statique
15         DaoImpl dao= new DaoImpl(); //objet dao version bd
16         DaoImplV2 dao= new DaoImplV2(); //objet dao version capteur
17         MetierImpl metier= new MetierImpl();
18         metier.setDao(dao);
19         System.out.println("Résultat: " + metier.Calcul());
20     }
21 }
22
```

<terminated> Presentation [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (8 mars 2022, 19:14:40)
Version base de données :
Résultat: 9315.046758087352



EXÉCUTION :

- **Instanciación Dinámica:**

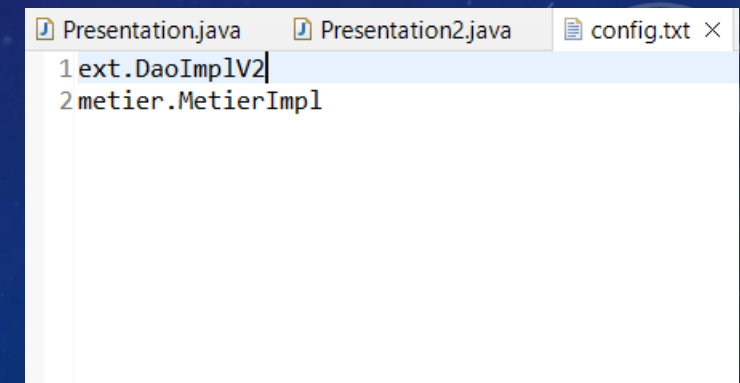


```
1 package pres;
2
3 import java.io.File;
4
5 public class Presentation2 {
6
7     public static void main(String[] args) throws FileNotFoundException, ClassNotFoundException, InstantiationException, IllegalAccessException {
8         //Injection des dépendances
9
10        //Instanciación Dinámica
11
12        String pathname = "config.txt";
13        Scanner sc = new Scanner(new File(pathname));
14
15        String daoClassName = sc.nextLine();
16        Class<?> cDao = Class.forName(daoClassName); //Charger la classe en mémoire
17        IDao dao = (IDao) cDao.getDeclaredConstructor().newInstance();
18
19        String metierClassName = sc.nextLine();
20        Class<?> cMetier = Class.forName(metierClassName); //Charger la classe en mémoire
21        IMetier metier = (IMetier) cMetier.getDeclaredConstructor().newInstance();
22
23        //Assosociation
24        Method method = cMetier.getMethod("setDao", IDao.class);
25        method.invoke(metier, dao);
26        System.out.println("Résultat: " + metier.Calcul());
27    }
28 }
```

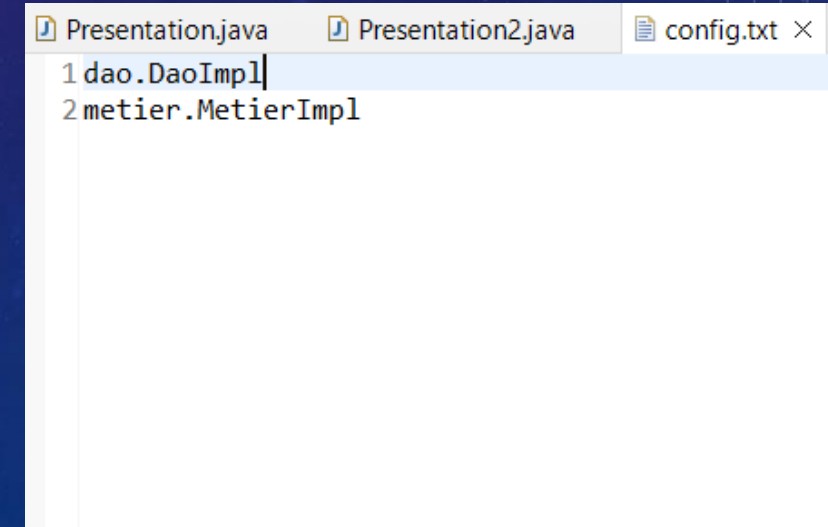
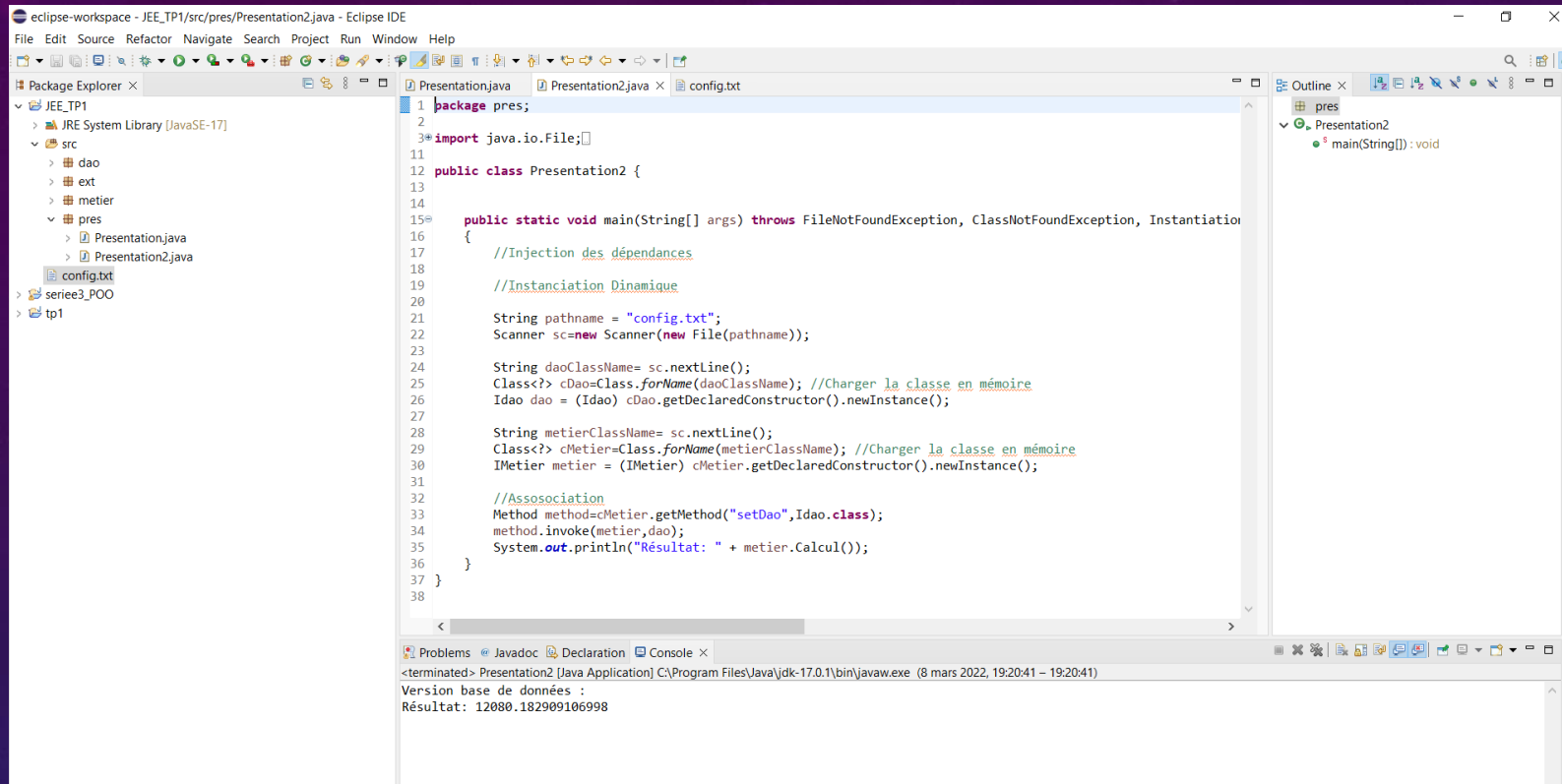
Problems | Javadoc | Declaration | Console

<terminated> Presentation2 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (8 mars 2022, 19:19:26 – 19:19:26)

Version capteurs :
Résultat: 500000.0

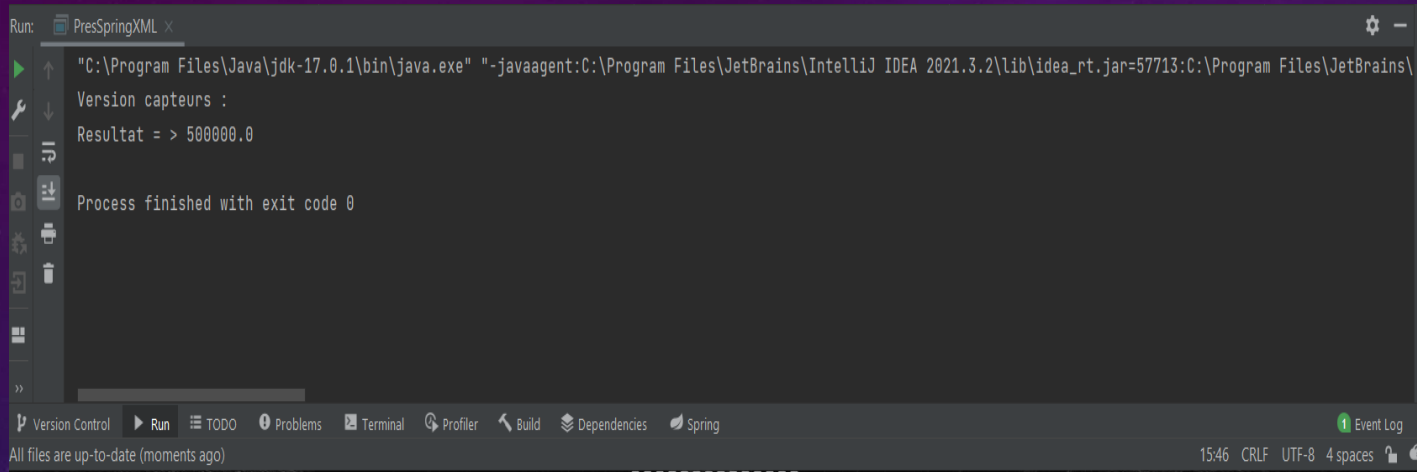


```
1 ext.DaoImplV2
2 metier.MetierImpl
```

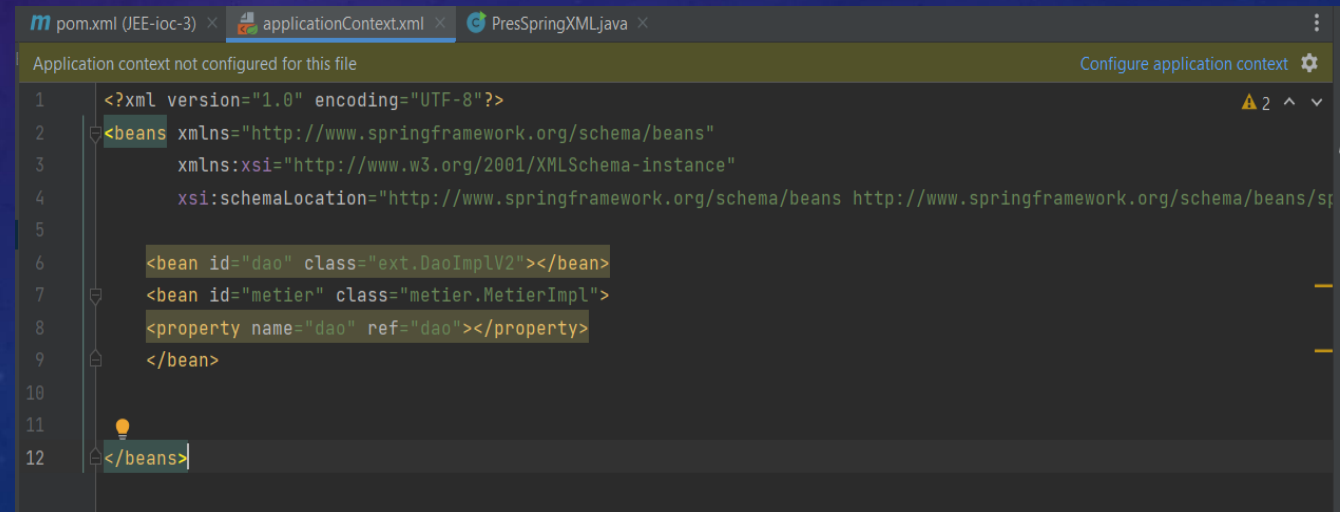


EXÉCUTION :

- **Instanciation version XML :**



```
Run: PresSpringXML x
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\lib\idea_rt.jar=57713:C:\Program Files\JetBrains\
Version capteurs :
Resultat = > 500000.0
Process finished with exit code 0
```



```
m pom.xml (JEE-ioc-3) x applicationContext.xml x PresSpringXML.java x
Application context not configured for this file Configure application context
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/sp
5
6   <bean id="dao" class="ext.DaoImplV2"></bean>
7   <bean id="metier" class="metier.MetierImpl">
8     <property name="dao" ref="dao"></property>
9   </bean>
10
11
12 </beans>
```

Run: PresSpringXML x

```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\lib\idea_rt.jar=57726:C:\Program Files\JetBrains\
Version base de donnees :
Resultat = > -17589.132309710814

Process finished with exit code 0
```

Version Control Run TODO Problems Terminal Profiler Build Dependencies Spring Event Log

Build completed successfully in 991 ms (moments ago) 15:46 CRLF UTF-8 4 spaces

m pom.xml (JEE-ioc-3) x applicationContext.xml x PresSpringXML.java x

Application context not configured for this file Configure application context

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/s
5
6   <bean id="dao" class="dao.DaoImpl"></bean>
7   <bean id="metier" class="metier.MetierImpl">
8     <property name="dao" ref="dao"></property>
9   </bean>
10
11
12 </beans>
```


EXÉCUTION :

- **Instanciación version Annotations :**

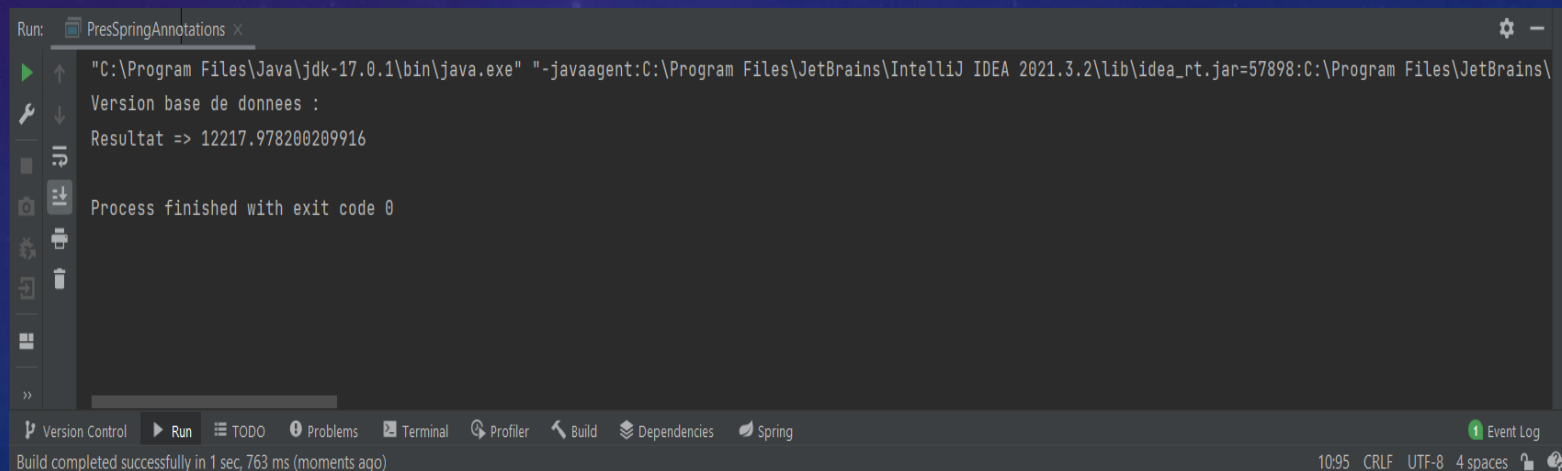


The screenshot shows the Run console in IntelliJ IDEA. The title bar indicates the run configuration is 'PresSpringAnnotations'. The console output shows the execution of a Java command using the JDK 17.0.1 and the IntelliJ RT JAR. The output includes 'Version capteurs :', 'Resultat => 500000.0', and 'Process finished with exit code 0'. The status bar at the bottom indicates the build completed successfully in 1 second and 988 milliseconds.

```
Run: PresSpringAnnotations x
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\lib\idea_rt.jar=57861:C:\Program Files\JetBrains\
Version capteurs :
Resultat => 500000.0

Process finished with exit code 0
```

Version Control Run TODO Problems Terminal Profiler Build Dependencies Spring Event Log
Build completed successfully in 1 sec, 988 ms (a minute ago) 10:95 CRLF UTF-8 4 spaces



This screenshot shows the Run console for the same 'PresSpringAnnotations' configuration. The output is similar to the first screenshot, but the result is 'Resultat => 12217.978200209916'. The status bar at the bottom shows the build completed successfully in 1 second and 763 milliseconds.

```
Run: PresSpringAnnotations x
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\lib\idea_rt.jar=57898:C:\Program Files\JetBrains\
Version base de donnees :
Resultat => 12217.978200209916

Process finished with exit code 0
```

Version Control Run TODO Problems Terminal Profiler Build Dependencies Spring Event Log
Build completed successfully in 1 sec, 763 ms (moments ago) 10:95 CRLF UTF-8 4 spaces

CONCLUSION:

- L'application qui n'évolue pas meurt. Elle doit s'adapter aux changements, et elle doit être fermée à la modification et ouverte à l'extension. Ce principe est basé sur le couplage faible qui utilise les interfaces.
- N'oublions pas que l'injection des dépendances restera aussi un principe très important et qui est la base de tout programme moderne.