

## Activite 1 JEE

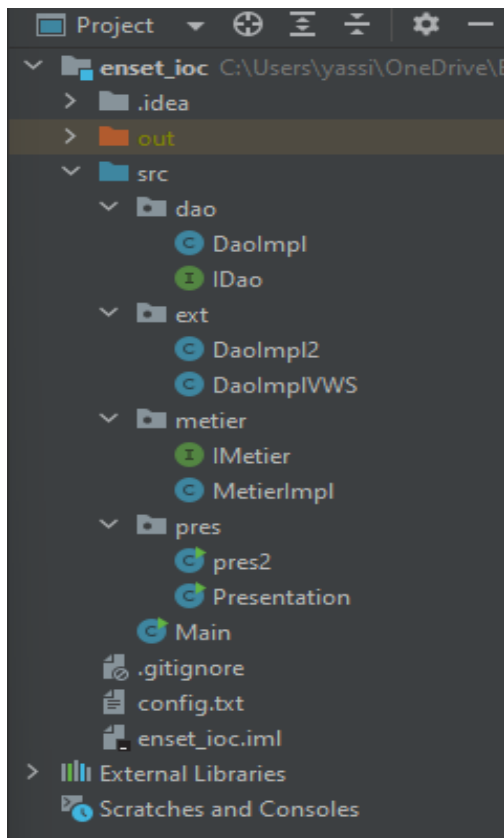
Hicham Yassine 4IIR G4

### Partie 1 :

L'application Enset-ioc qui contient 4 package (dao,ext,metier,pres)

Chaque package contient une ou plusieurs class

Pour package dao et métier on a deux interfaces



Exmpl d une interface

```
IDao.java × IMetier.java × DaoImpl.java ×
package metier;

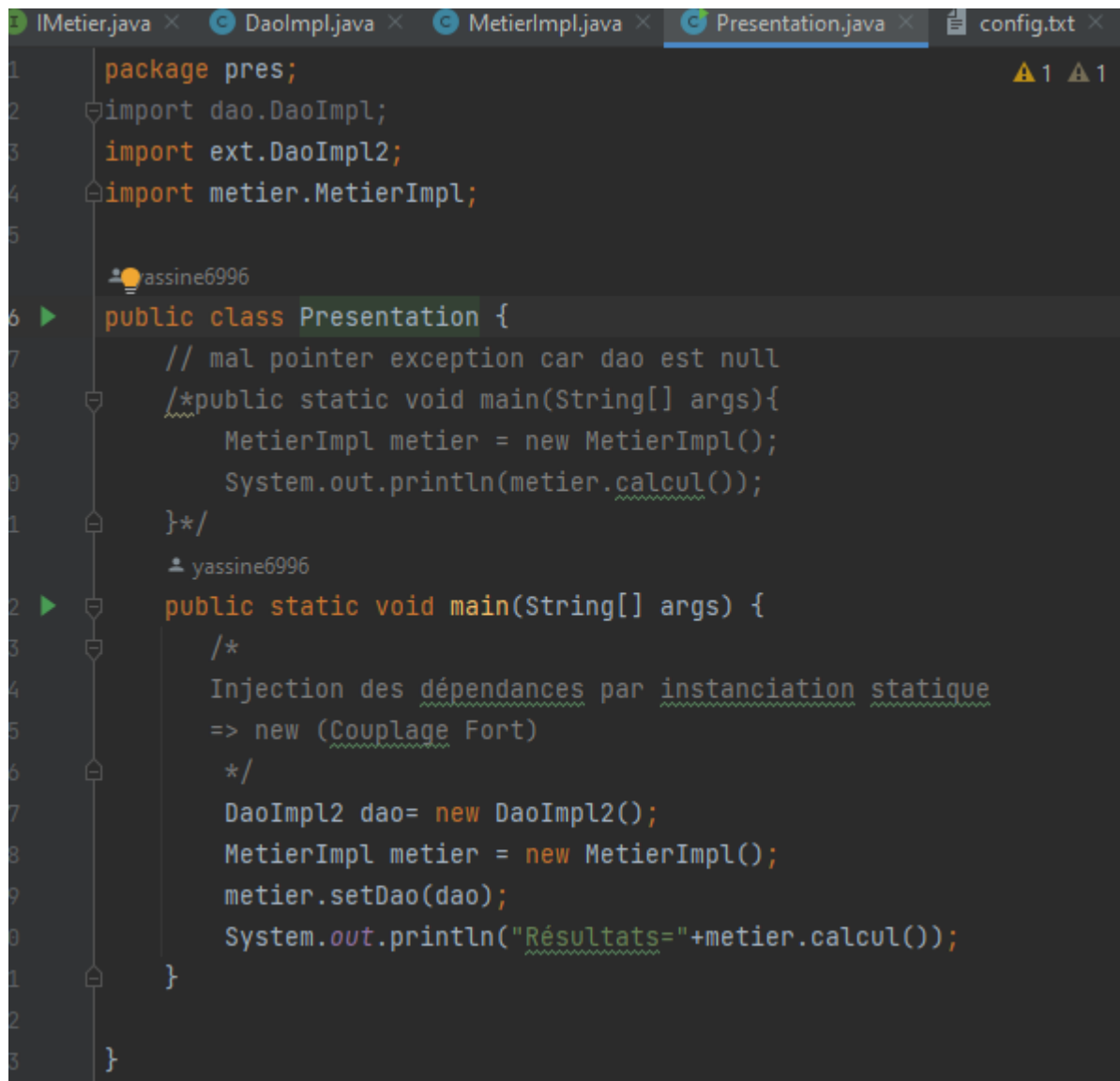
4 usages 1 implementation yassine6996
public interface IMetier {
    2 usages 1 implementation yassine6996
    double calcul();
}
```

L'implémentation du code calcul

```
package metier;
import dao.IDao;

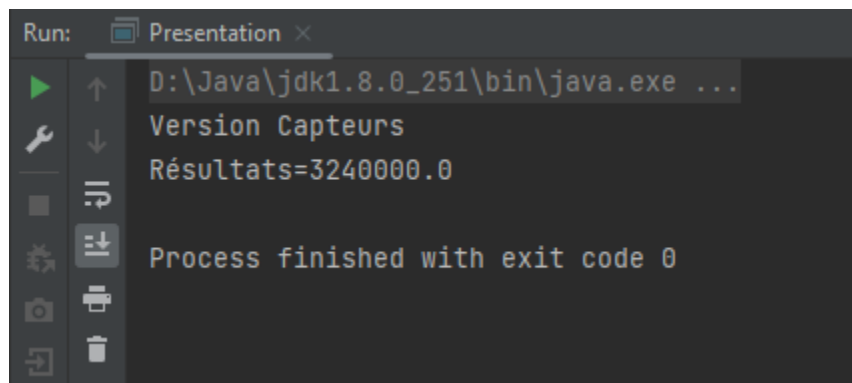
4 usages yassine6996
public class MetierImpl implements IMetier {
    //Couplage faible si on utilise new donc couplage fort
    2 usages
    private IDao dao;
    2 usages yassine6996
    public double calcul() {
        double tmp=dao.getData();
        double res=tmp*540/Math.cos(tmp*Math.PI);
        return res;
    }
    /*
    Injecter dans la variable dao un objet d'une classe qui
    implémente l'interface IDao
    */
    1 usage yassine6996
    public void setDao(IDao dao) {
        this.dao = dao;
    }
}
```

Le code Main de la présentation



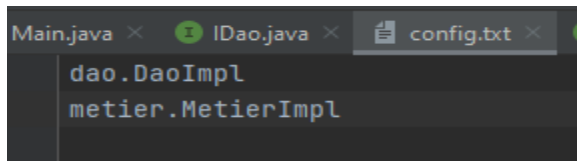
```
1 package pres;
2 import dao.DaoImpl;
3 import ext.DaoImpl2;
4 import metier.MetierImpl;
5
6 public class Presentation {
7     // mal pointer exception car dao est null
8     /*public static void main(String[] args){
9         MetierImpl metier = new MetierImpl();
10        System.out.println(metier.calcul());
11    }*/
12
13    public static void main(String[] args) {
14        /*
15        Injection des dépendances par instanciation statique
16        => new (Couplage Fort)
17        */
18        DaoImpl2 dao= new DaoImpl2();
19        MetierImpl metier = new MetierImpl();
20        metier.setDao(dao);
21        System.out.println("Résultats="+metier.calcul());
22    }
23 }
```

Execution du code presentation avec version capteurs



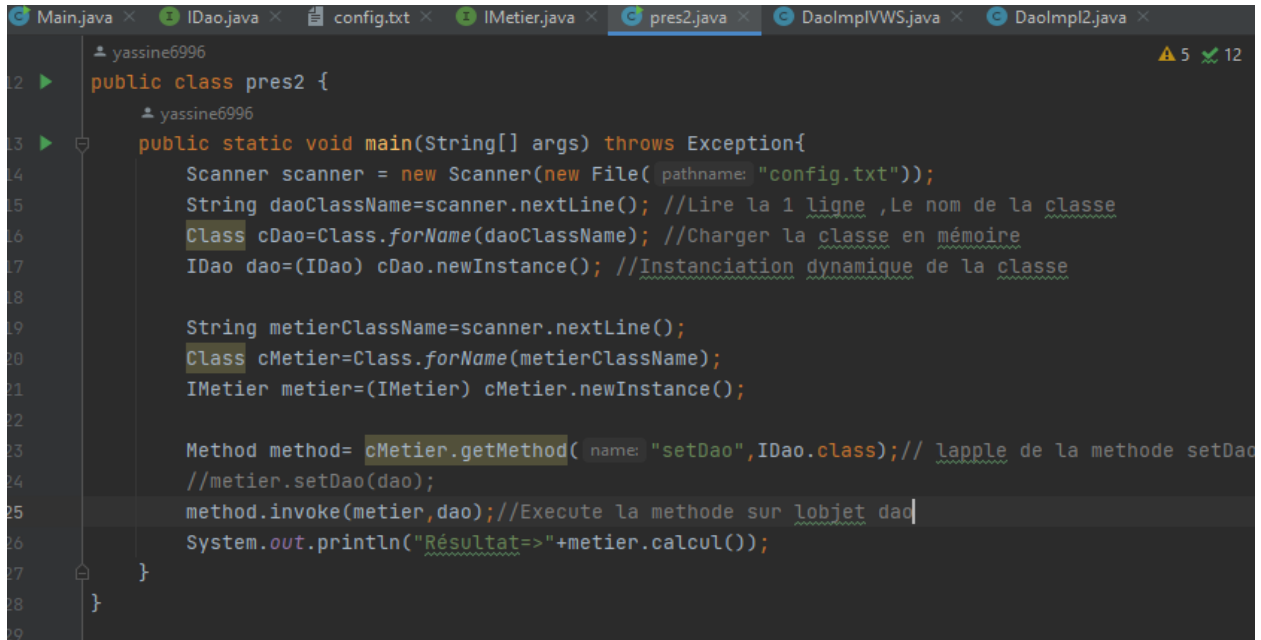
```
Run: Presentation
D:\Java\jdk1.8.0_251\bin\java.exe ...
Version Capteurs
Résultats=3240000.0
Process finished with exit code 0
```

Config file



```
dao.DaoImpl
metier.MetierImpl
```

## Presentation 2

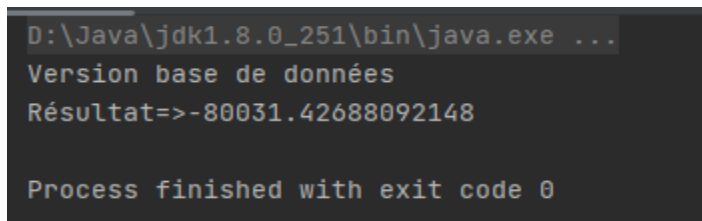


```
public class pres2 {
    public static void main(String[] args) throws Exception{
        Scanner scanner = new Scanner(new File("config.txt"));
        String daoClassName=scanner.nextLine(); //Lire la 1 ligne ,Le nom de la classe
        Class cDao=Class.forName(daoClassName); //Charger la classe en mémoire
        IDao dao=(IDao) cDao.newInstance(); //Instanciation dynamique de la classe

        String metierClassName=scanner.nextLine();
        Class cMetier=Class.forName(metierClassName);
        IMetier metier=(IMetier) cMetier.newInstance();

        Method method= cMetier.getMethod("setDao",IDao.class); // l'appelle de la methode setDao
        //metier.setDao(dao);
        method.invoke(metier,dao); //Execute la methode sur l'objet dao
        System.out.println("Résultat=>" + metier.calcul());
    }
}
```

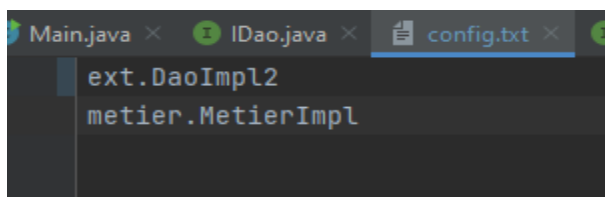
## Exécution



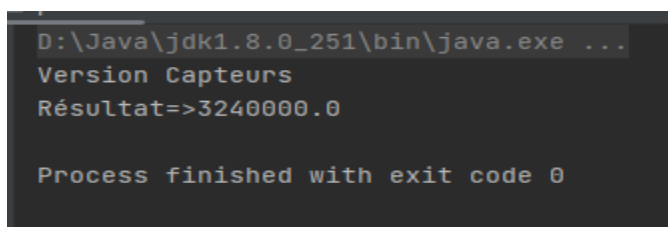
```
D:\Java\jdk1.8.0_251\bin\java.exe ...
Version base de données
Résultat=>-80031.42688092148

Process finished with exit code 0
```

Si on change le config file



```
ext.DaoImpl2
metier.MetierImpl
```



```
D:\Java\jdk1.8.0_251\bin\java.exe ...
Version Capteurs
Résultat=>3240000.0

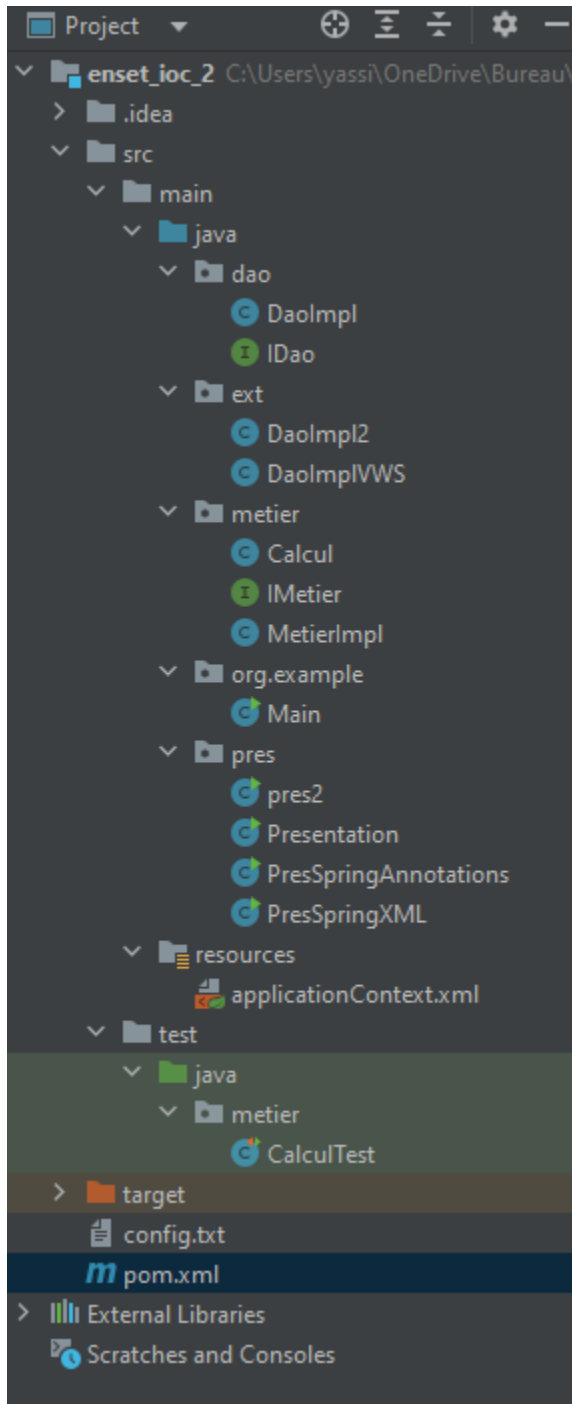
Process finished with exit code 0
```

Le resultat sera changer car on a exécuter le code sur DaoImpl2

## Partie 2 :

Cette app va être créer avec maven pour ajouter des dependences

1<sup>er</sup> on va réutiliser les packages du partie 1



L'ajout des dépendances dans le fichier pom.xml

```
<!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>5.3.20</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>5.3.20</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-beans -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-beans</artifactId>
  <version>5.3.20</version>
</dependency>

<!-- https://mvnrepository.com/artifact/junit/junit -->
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.12</version>
  <scope>test</scope>
</dependency>
```

La creation du appcontext

```
applicationContext.xml x PresSpringAnnotations.java x DaoImplVWS.java x Calcul.java x
Application context not configured for this file
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 h
5     <bean id="dao" class="ext.DaoImplVWS"></bean>
6     <bean id="metier" class="metier.MetierImpl">
7         <constructor-arg ref="dao"></constructor-arg>
8     </bean>
9 </beans>
```

Expl du code d annotation pour scanner

```
applicationContext.xml x PresSpringAnnotations.java x DaoImplVWS.java x Calcul.java x CalculTest.java x pom.xml (enst_ioc_2) x config.txt x Me
1 package pres;
2
3 import metier.IMetier;
4 import org.springframework.context.ApplicationContext;
5 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
6
7 public class PresSpringAnnotations {
8     public static void main(String[] args) {
9         ApplicationContext context=new AnnotationConfigApplicationContext( ...basePackages: "dao","metier","ext");//scanner les pac
10        IMetier metier=context.getBean(IMetier.class);
11        System.out.println(metier.calcul());
12    }
13 }
```

L'utilisation du appcontext.xml

```
1 package pres;
2
3 import metier.IMetier;
4 import org.springframework.context.ApplicationContext;
5 import org.springframework.context.support.ClassPathXmlApplicationContext;
6
7 public class PresSpringXML {
8     public static void main(String[] args) {
9         ApplicationContext context=new ClassPathXmlApplicationContext( configLocation: "applicationContext.xml");
10        IMetier metier=(IMetier) context.getBean( s: "metier");//cast sur linterface
11        System.out.println("Résultat=>" +metier.calcul());
12    }
13 }
```