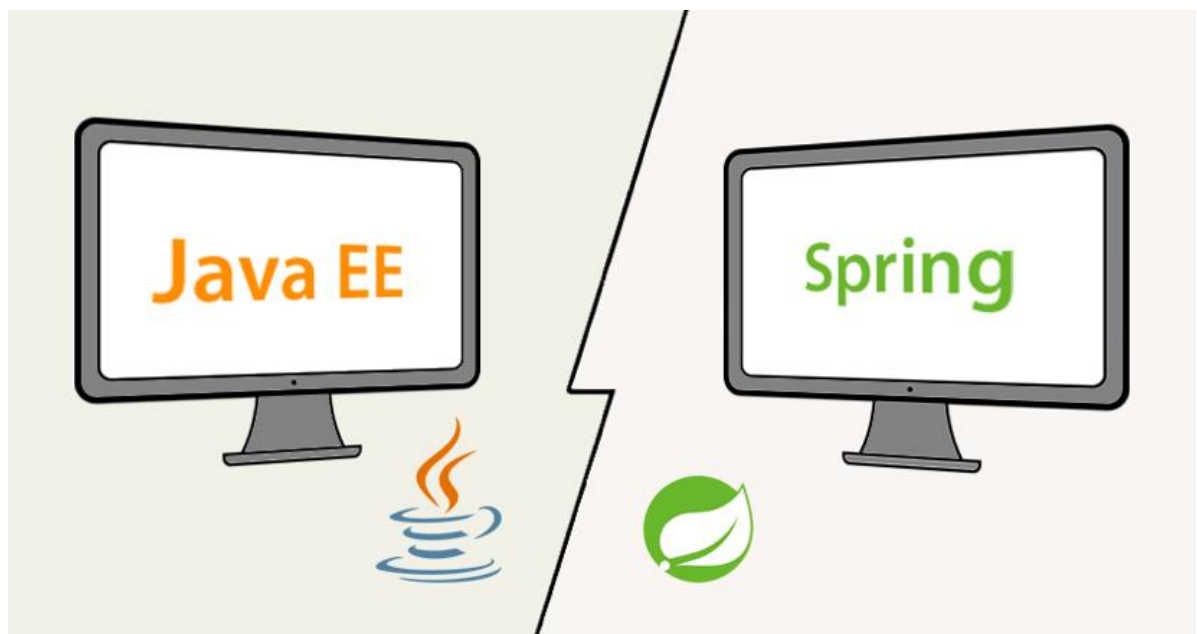


Compte rendu du TP1 : Inversion de contrôle et Injection des dépendances



Réalisé Par : AYOUIJIL Soukayna
Département : Mathématiques et Informatique
Filière : II-BDCC
Professeur : M. Mohamed YOUSSEFI

Année Universitaire : 2021/2022

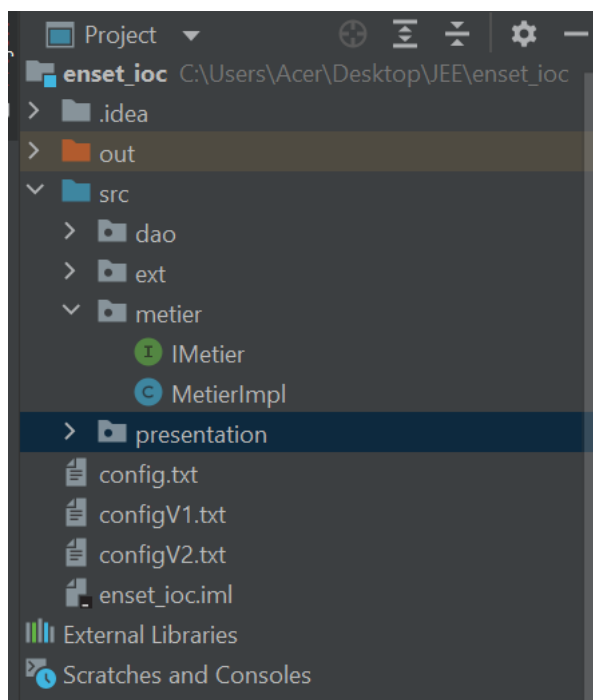
1. Créer l'interface IDao :

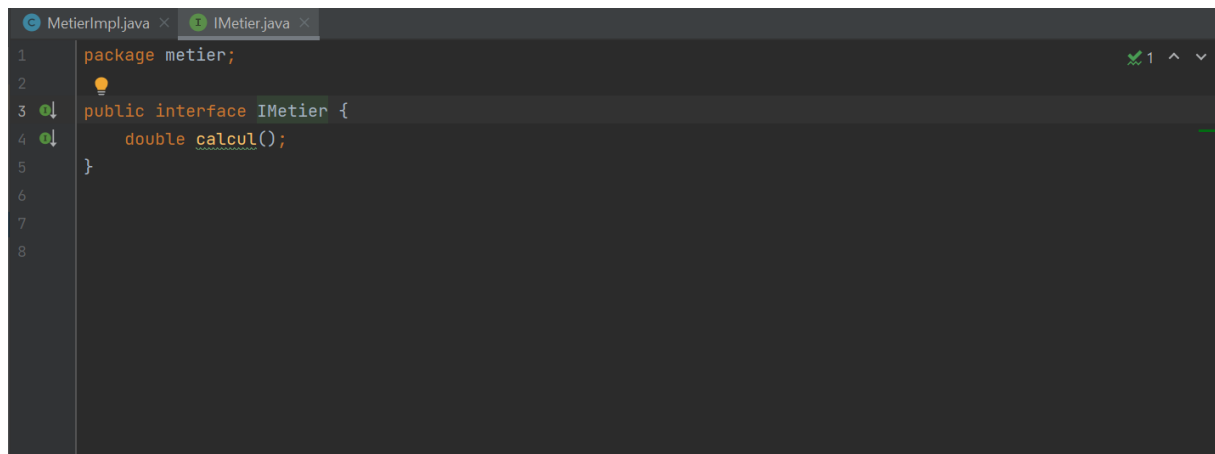
```
IDao.java
1 package dao;
2
3 public interface IDao {
4     double getData();
5 }
6
```

2. Créer une implémentation de cette interface

```
DaoImpl.java
1 package dao;
2
3 public class DaoImpl implements IDao {
4     @Override
5     public double getData() {
6         /*
7          * Se connecter à la BD pour récupérer la température
8          */
9         System.out.println("version base de données");
10        double temp = Math.random()*40;
11        return temp;
12    }
13 }
14
```

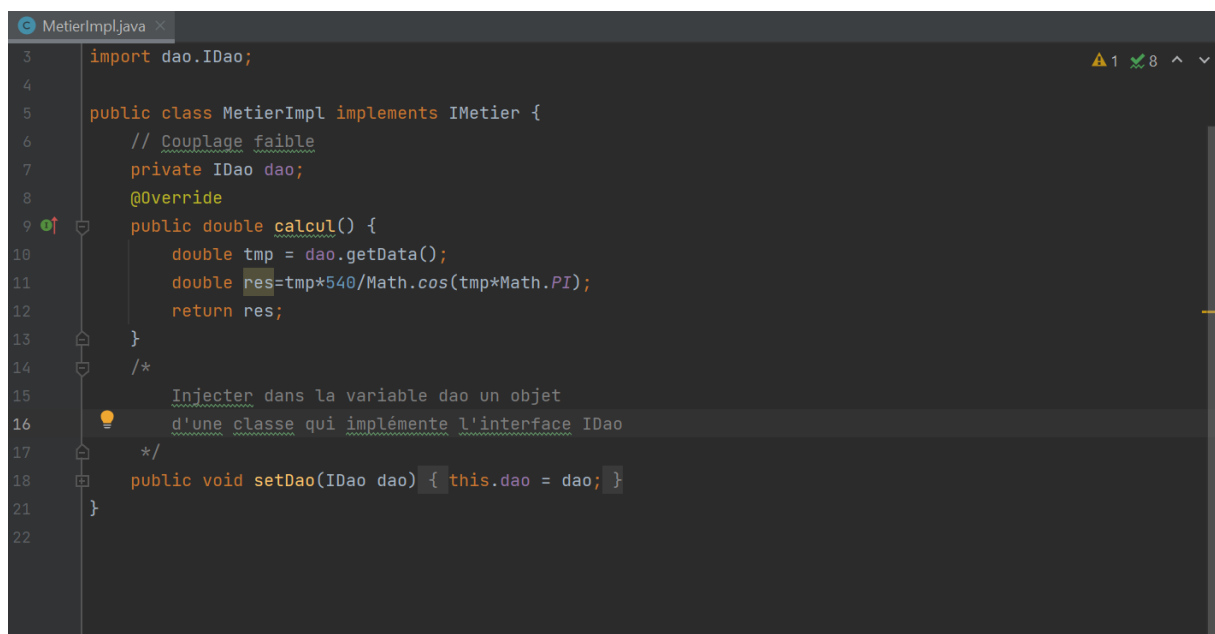
3. Créer l'interface IMetier





```
1 package metier;
2
3 public interface IMetier {
4     double calcul();
5 }
6
7
8
```

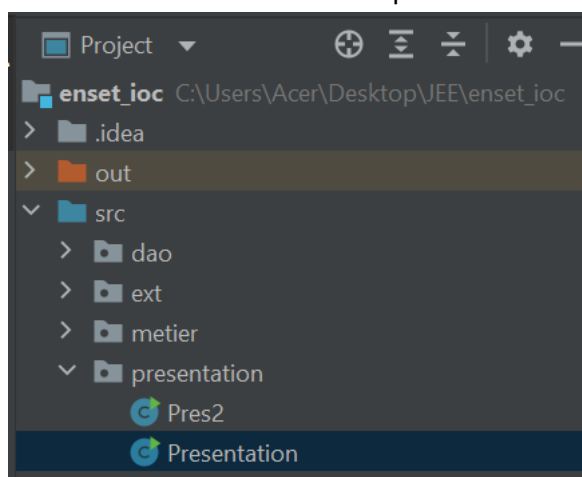
4. Créer une implémentation de cette interface en utilisant le couplage faible



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

5. Faire l'injection des dépendances :

a. Par instantiation statique



```
MetierImpl.java x Presentation.java x
6
7 public class Presentation {
8     public static void main(String[] args) {
9
10         // Injection des dépendances par instantiation statique => new //
11         /*
12         DaoImpl dao = new DaoImpl();
13         MetierImpl metier = new MetierImpl();
14         metier.setDao(dao);
15         */
16         DaoImpl2 dao = new DaoImpl2();
17         MetierImpl metier = new MetierImpl();
18         metier.setDao(dao);
19         System.out.println("Résultat="+metier.calcul());
20     }
21 }
22
```

b. Par instantiation dynamique

```
Project
enset_ioc C:\Users\Acen\Desktop\VEE\enset_ioc
> idea
> out
> src
> dao
> ext
> metier
> presentation
Pres2
Presentation
config.txt
configV1.txt
configV2.txt
enset_iocimpl
External Libraries
Scratches and Consoles

Pres2.java
11 public class Pres2 {
12     public static void main(String[] args) throws Exception {
13         Scanner scanner = new Scanner(new File( pathname: "config.txt"));
14
15         String daoClassName = scanner.nextLine();
16         Class cDao = Class.forName(daoClassName);
17         IDao dao = (IDao) cDao.newInstance();
18
19         String metierClassName=scanner.nextLine();
20         Class cMetier = Class.forName(metierClassName);
21         IMetier metier=(IMetier) cMetier.newInstance();
22
23         Method method=cMetier.getMethod( name: "setDao",IDao.class);
24         //metier.setDao(dao);
25         method.invoke(metier,dao);
26
27         System.out.println("Resultat=>"+metier.calcul());
28     }
29 }
30 }
31
```

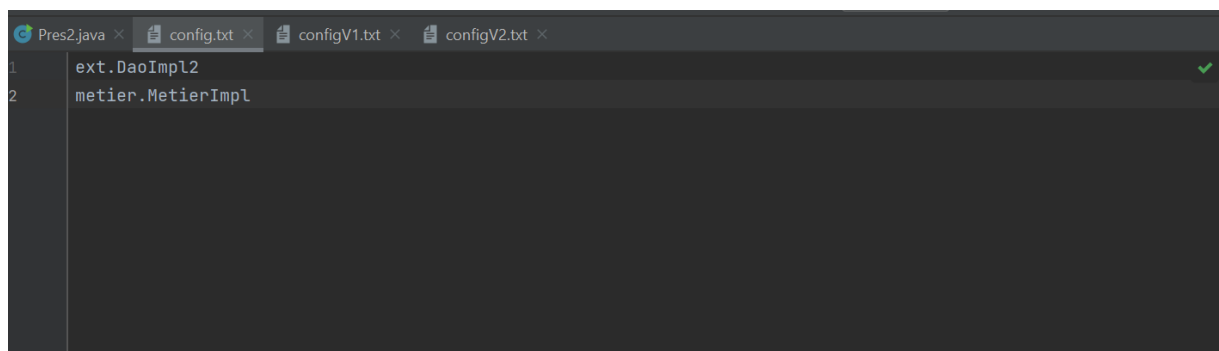
Fichier config.txt :

```
Pres2.java x config.txt x configV1.txt x
1 dao.DaoImpl
2 metier.MetierImpl
```



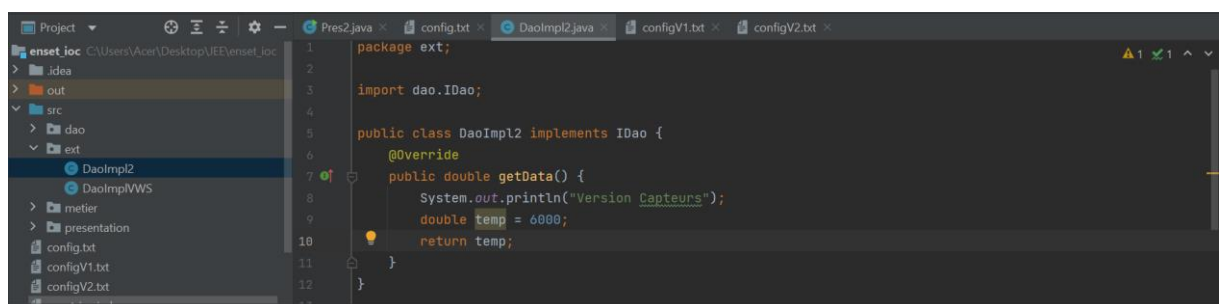
```
Run: Pres2 x
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
version base de données
Resultat=>-16134.117374357747
Process finished with exit code 0
```

Fichier config.txt : DaoImpl2

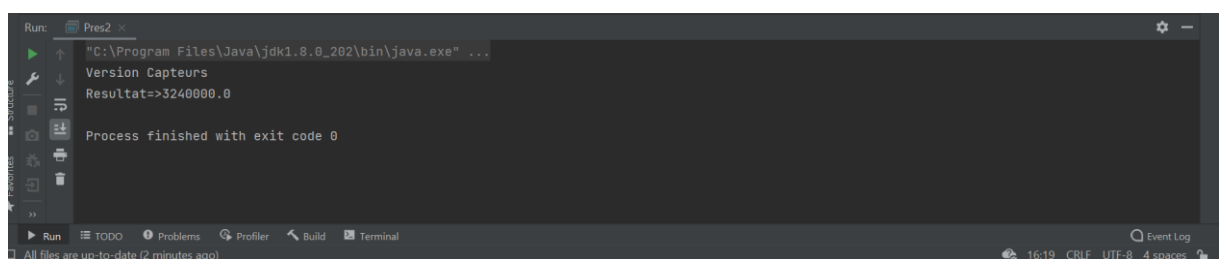


```
Pres2.java x config.txt x configV1.txt x configV2.txt x
1 ext.DaoImpl2
2 metier.MetierImpl
```

Ext.DaoImpl2.java :



```
Project | Pres2.java x config.txt x DaoImpl2.java x configV1.txt x configV2.txt x
1 package ext;
2
3 import dao.IDao;
4
5 public class DaoImpl2 implements IDao {
6     @Override
7     public double getData() {
8         System.out.println("Version Capteurs");
9         double temp = 6000;
10        return temp;
11    }
12 }
```



```
Run: Pres2 x
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
Version Capteurs
Resultat=>3240000.0
Process finished with exit code 0
```

Fichier config.txt : DaoImplVWS

```
Pres2.java x DaoImplVWS.java x config.txt x
1 ext.DaoImplVWS
2 metier.MetierImpl
```

ext.DaoImplVWS.java :

```
Pres2.java x DaoImplVWS.java x
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 90;
10    }
11 }
12 }
```

```
Run: Pres2 x
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
Version web service
Resultat=>48600.0
Process finished with exit code 0
```

c. En utilisant le Framework Spring - Version XML

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
enstet-joc-2 - ApplicationContext.xml
Project
  enstet-joc-2 [enstet-joc-2] C:\Users\Acer\Desktop\enstet-joc-2
  src
    main
      java
        dao
        ext
        metier
        presentation
      resources
        ApplicationContext.xml
      test
      target
      config.txt
      pom.xml
  External Libraries
    < 1.8 > C:\Program Files\Java\jdk1.8.0_202
    Maven: org.springframework:spring-aop:5.3.16
    Maven: org.springframework:spring-beans:5.3.16
    Maven: org.springframework:spring-context:5.3.16
  Run: PresSpringXML 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 http://www.springframework.org/schema/beans.xsd">
  5     <!--<bean id="dao" class="dao.DaoImpl"></bean-->
  6     <!--<bean id="dao" class="ext.DaoImpl2"></bean-->
  7     <bean id="dao" class="ext.DaoImplVWS"></bean>
  8     <bean id="metier" class="metier.MetierImpl">
  9         <property name="dao" ref="dao"></property>
  10    </bean>
  11 </beans>
```

The screenshot shows an IDE with two tabs: `ApplicationContext.xml` and `PresSpringXML.java`. The `PresSpringXML.java` file contains the following code:

```
1 package presentation;
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("ApplicationContext.xml");
10        IMetier metier = (IMetier) context.getBean("metier");
11        System.out.println("Resultat=>" + metier.calcul());
12    }
13 }
14
```

Below the code editor, the `Run` window shows the execution of `PresSpringXML`. The command used is `"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...`. The output is `Version web service` and `Resultat=>48600.0`. The process finished with exit code 0.

- Version annotations

The screenshot shows an IDE with four tabs: `DaoImpl.java`, `MetierImpl.java`, `PresSpringAnnotation.java`, and `ApplicationContext.xml`. The `DaoImpl.java` file contains the following code:

```
4
5 @Component("dao")
6 public class DaoImpl implements IDao {
7     @Override
8     public double getData() {
9         /*
10         Se connecter à la BD pour récupérer la température
11         */
12         System.out.println("version base de données");
13         double temp = Math.random()*40;
14         return temp;
15     }
16 }
17
```

=> Via Autowired

The screenshot shows an IDE with the `MetierImpl.java` file. The code is as follows:

```
7 @Component("metier")
8 public class MetierImpl implements IMetier {
9     // Couplage faible
10    @Autowired
11    private IDao dao;
12    @Override
13    public double calcul() {
14        double tmp = dao.getData();
15        double res=tmp*540/Math.cos(tmp*Math.PI);
16        return res;
17    }
18 }
```

=> Via Constructeur

```
6
7  @Component("metier")
8  public class MetierImpl implements IMetier {
9      // Couplage faible
10     // @Autowired
11     private IDao dao;
12
13     public MetierImpl(IDao dao) {
14         this.dao = dao;
15     }
16
```

