TP1: Inversion de Controle
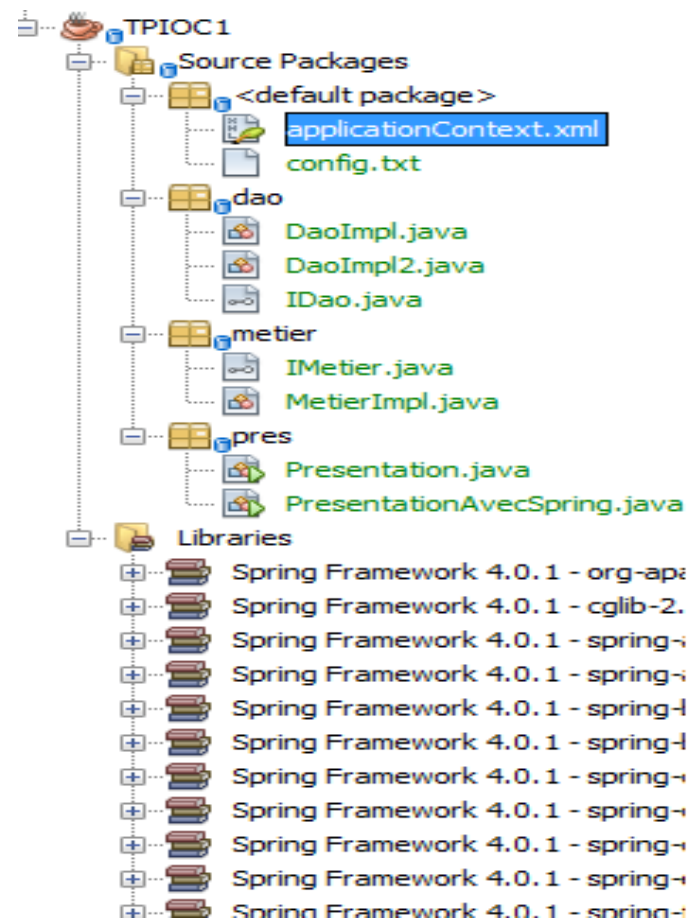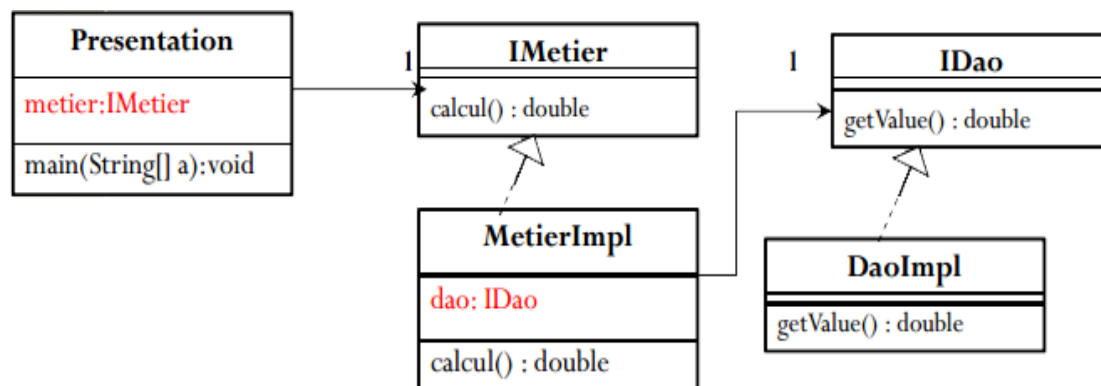
Ayoub MOUSTAID - GLSID2
Année universitaire : 2015/2016

Structure

- **IDE : Netbeans - Architecture**

```
TPIOC1
  Source Packages
    <default package>
      applicationContext.xml
      config.txt
    dao
      DaoImpl.java
      DaoImpl2.java
      IDao.java
    metier
      IMetier.java
      MetierImpl.java
    pres
      Presentation.java
      PresentationAvecSpring.java
  Libraries
    Spring Framework 4.0.1 - org-apa
    Spring Framework 4.0.1 - cglib-2.
    Spring Framework 4.0.1 - spring-
    Spring Framework 4.0.1 - spring-
    Spring Framework 4.0.1 - spring-
    Spring Framework 4.0.1 - spring-
    Spring Framework 4.0.1 - spring-
    Spring Framework 4.0.1 - spring-
    Spring Framework 4.0.1 - spring-
    Spring Framework 4.0.1 - spring-
    Spring Framework 4.0.1 - spring-
```

- **Diagramme de classes**

Presentation
- metier:IMetier
- main(String[] a):void

IMetier
- calcul() : double

IDao
- getValue() : double

MetierImpl
- dao: IDao
- calcul() : double

DaoImpl
- getValue() : double

Implémentation

1.Couche DAO

- **Interface IDao**

```
package dao;

public interface IDao {

    double getTemperature();

}
```

- **Classe DaoImpl**

```
package dao;

public class DaoImpl implements IDao {

    @Override

    public double getTemperature() {

        double val=24;

        return val;

    }

}
```

- **Classe DaoImpl2**

```
package dao;

public class DaoImpl2 implements IDao {

    @Override

    public double getTemperature() {

        double val=50;

        return val;
```

```
}}
```

2. Couche Metier

- **Interface IMetier**

```
package metier;

public interface IMetier {

   double calcule();

}
```

- **Classe MetierImpl**

```
package metier;

import dao.IDao;

public class MetierImpl implements IMetier{

   private IDao dao;

   @Override

   public double calcule() {

      double val=dao.getTemperature();

      return val*10;

   }

   public  void setDao(IDao dao)

   {

   this.dao=dao;

   }

}
```

3. Couche Presentation

- **Presentation (version instantiation statique)**

```
package pres;

import dao.IDao;

import metier.IMetier;
```

```
import java.io.File;

import java.lang.reflect.Method;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;


public class Presentation {

    public static void main(String[] args) {

        try {

            DaoImpl dao= new DaoImpl();

            MetierImpl metier=new MetierImpl();

            metier.setDao(dao);

            System.out.println(metier.calcule());

        } catch (Exception ex) {

            Logger.getLogger(Presentation.class.getName()).log(Level.SEVERE, null, ex);

        }

    }
```

- **Classe Presentation (version instantiation dynamique)**

```java
package pres;

import dao.IDao;

import metier.IMetier;

import java.io.File;

import java.lang.reflect.Method;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Presentation {

    public static void main(String[] args) {

        try {

            Scanner scanner =new  Scanner(new File("config.txt"));

            String daoClassName=scanner.next();

            String metierClassName=scanner.next();

            Class cDao=Class.forName(daoClassName);

            IDao dao=(IDao)cDao.newInstance();//doit avoir un constructeur par defaut

            Class cMetier=Class.forName(metierClassName);

            IMetier metier=(IMetier)cMetier.newInstance();

            Method m=cMetier.getMethod("setMetier", new Class[]{IDao.class});

            m.invoke(metier,new Object[]{dao});

        } catch (Exception ex) {

            Logger.getLogger(Presentation.class.getName()).log(Level.SEVERE, null, ex);

        }
```

```
    }
```

- **Classe PresentationAvecSpring (version Spring)**

```java
package pres;

import metier.IMetier;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class PresentationAvecSpring {

   public static void main(String[] args) {

      ClassPathXmlApplicationContext context= new ClassPathXmlApplicationContext(new
String[]{"applicationContext.xml"});

      IMetier metier=(IMetier) context.getBean("metier");

      System.out.println(metier.calcule());

   }}
```

- **Fichier config.txt**

```
dao.DaoImpl

metier.MetierImpl
```

- **applicationContext.xml**

```xml
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd

">

  <bean id="d" class="dao.DaoImpl"></bean>

  <bean id="metier" class="metier.MetierImpl">
```

```
        <property name="dao" ref="d"></property>

</bean></beans>
```

- **Resultat**
  (Choix du classe DaoImpl1)

  (Choix du classe DaoImpl2)