Ecole Normale Supérieure de l'Enseignement Technique Mohammedia

Université Hassan II de Casablanca

ENSET

ا لمدرسة العليا لأساتذة التعليم التقني المحمدية

جامعة الحسن الثاني بالدار البيضاء

DEPARTEMENT MATHEMATIQUES ET INFORMATIQUE

Compte rendu

- Travaux Pratique : 1- « Architecture JEE »

INVERSION DE CONTROLE ET INJECTION DES DEPENDANCES

GLSID II

Réalisé par : Ikram BERRADI

1. Créer l'interface IDao

```java
package dao;

import org.springframework.stereotype.Component;

@Component("dao")
public class DaoImp implements Dao {

    @Override
    public double getData() {
        //dévlopeur
        //se coecter a la bd pour récuperer la donnée
        System.out.println("version-1");
        double temp= Math.random()*40;

        return temp;
    }
}
```

2. Créer une implémentation de cette interface

```java
package dao;
import org.springframework.stereotype.Component;
@Component("dao")
public class DaoImp implements Dao {

    @Override
    public double getData() {
        //dévlopeur
        //se coecter a la bd pour récuperer la donnée
        System.out.println("version-1");
        double temp= Math.random()*40;

        return temp;
    }
}
```

3. Créer l'interface IMetier

```java
package dao;
//consepeur
public interface Dao {
    public double getData();//methode abstaite
}
```

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

```java
package metier;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import dao.Dao;
@Component
public class MetierImp implements Metier{
    //Couplage Faible
    @Autowired
    private Dao dao=null;
   /* public MetierImp(Dao dao) {
        this.dao = dao;
    }*/
    @Override
    public double calcul() {
        double data=dao.getData();
        double res=data*223*Math.cos(data);
        return res;
    }
    //pour injecter dans la variable dao un objet d'une clase qui impliment l' interface Dao
    public void setDao(Dao dao) { this.dao = dao; }
}
```

ext.DaoImpV2 :

```java
package ext;
import dao.Dao;
import org.springframework.stereotype.Component;
@Component("dao2")
public class DaoImpV2 implements Dao {
    @Override
    public double getData() {
        System.out.println("version-2");

        double temp= Math.random()*40;

        return temp;    }
}
```

5. Faire l'injection des dépendances :
   a. Par instanciation statique
   Présentation :

```java
package présentation;
import dao.DaoImp;
import metier.MetierImp;
public class Présentation_V1 {
        public static void main(String[] args) {
            // injection des dependense par instaciation statique =>new
            //en utiliser setr
            MetierImp metier=new MetierImp();
            DaoImp dao=new DaoImp();
            metier.setDao(dao); //metierImp-->DaoImp
            System.out.println("instaciation statique=>"+metier.calcul());}
}
```

Exécution :

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
version-1
instaciation statique=>-505.5601388443662

Process finished with exit code 0
```

b. Par instanciation dynamique

Présentation :

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import java.util.Scanner;
public class Présentation_V2 {
    public static void main(String[] args) throws FileNotFoundException, ClassNotFoundException, InstantiationE
       /*ovrire une ficher*/
        Scanner scanner=new Scanner(new File( pathname: "src/config.txt"));
      /* /*lire premier line de fichier txt*/
       String daoClassName=scanner.nextLine();
        /* chercher si la classe  il existe et charger au mémoire si non un erreue class not fonde exeption  */
       Class cDao=Class.forName(daoClassName);
        //cherche un constrecteur son parametre =erreur INstasiatuionexseption
       Dao dao= (Dao) cDao.newInstance();//demader de crées un object
       System.out.println(dao.getData());
        String metierClassName=scanner.nextLine();
        Class cMetier=Class.forName(metierClassName);
        Metier metier=(Metier)cMetier.newInstance();
//creé un objet métthode pur stoker dans laquelle  l'objet de la class dao
        Method methode=cMetier.getMethod( name: "setDao",Dao.class);
        //executer la méthode
        methode.invoke(metier,dao);
        System.out.println("instaciation dynamique=>"+metier.calcul());
    }}
```

Fichier config.txt :

```
dao.DaoImp
metier.MetierImp
```

Exécution :

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
version-1
instaciation dynamique=>-679.5077461335962

Process finished with exit code 0
```

c. En utilisant le Framework Spring

- Version XML

Presentation

```
import metier.Metier;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class PresentationSpringXml {
    public static void main(String[] args)  {
        ApplicationContext context=new ClassPathXmlApplicationContext( configLocation: "config.xml");
        Metier metier=(Metier) context.getBean( s: "metier");
        System.out.println("SpringVersion-XML=>"+metier.calcul());

    }
}
```

Fichier config.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/be
<bean id="dao" class="ext.DaoImpV2"></bean>
    <bean id="metier" class="metier.MetierImp">
        <property name="dao" ref="dao"></property>
    </bean>

</beans>
```

Exécution :

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
version-2
SpringVersion-XML=>2653.673860701793

Process finished with exit code 0
```

- Version annotations

Présentation :

```java
import metier.Metier;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;


public class PresentationStringAnotation {
    public static void main(String[] args) {
    ApplicationContext context=new AnnotationConfigApplicationContext( ...basePackages: "dao2","metier","ext");
    Metier metier=context.getBean(Metier.class);
    System.out.println("SpringVersion Annotation=>"+metier.calcul());
}}
```

Exécution :

```
"C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" ...
version-2
SpringVersion Annotation=>-1948.4517434806628

Process finished with exit code 0
```