**ENSET-M 2021-2022**

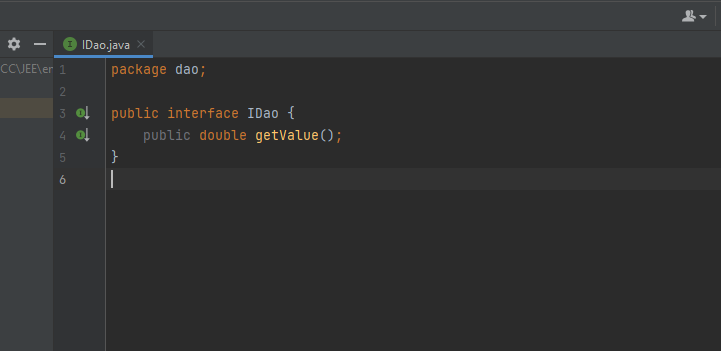
**COMPTE RENDU**

**TP1 JEE : Inversion de contrôle et Injection des dépendances**

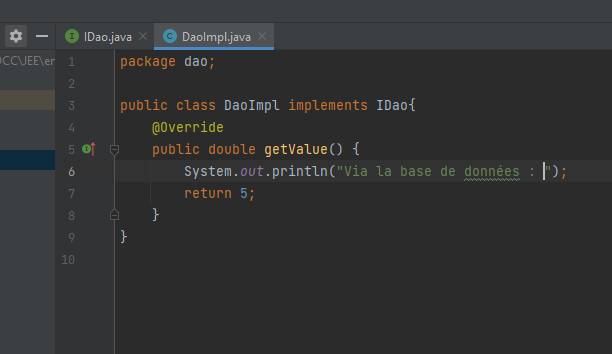
Par : **Assimi DIALLO** Encadrant : **M. YOUSSFI**

Ceci est le premier TP du module Architectures Distribuées JEE et Middlewares. Il nous permettra de faire une prise de contact avec l’inversion de contrôle et l’injection des dépendances et ça sera aussi l’occasion pour une première utilisation du framework Spring.

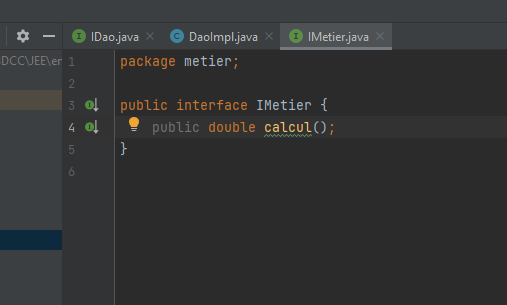
1. Création de l’interface IDao



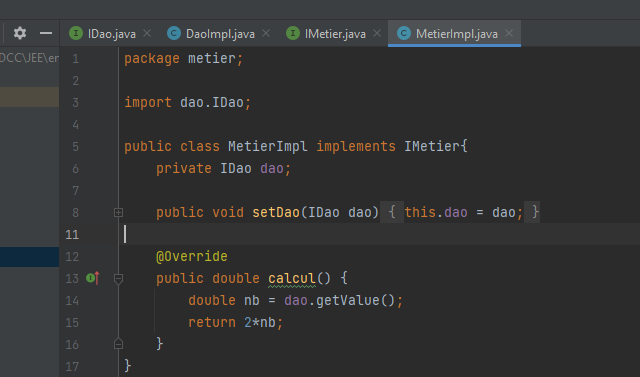
1. Création d’une implémentation de l’interface IDao



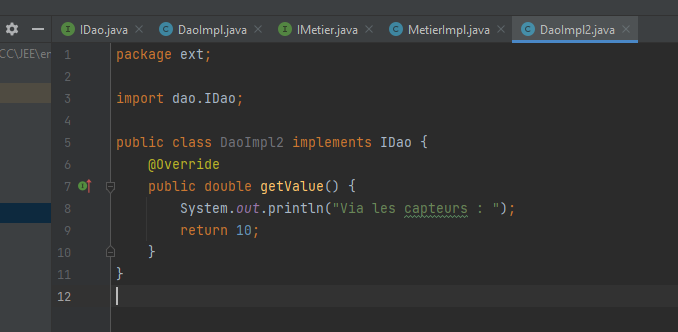
1. Création de l'interface IMetier

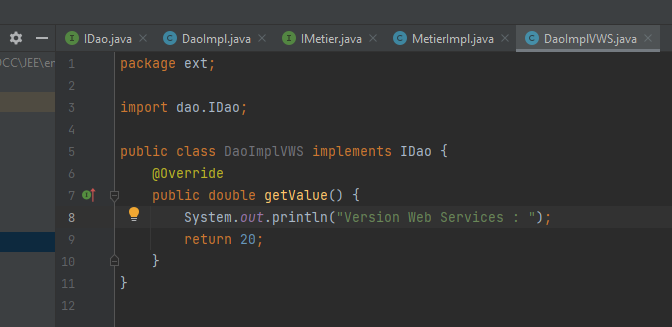


1. Création d’une implémentation de cette interface en utilisant le couplage faible



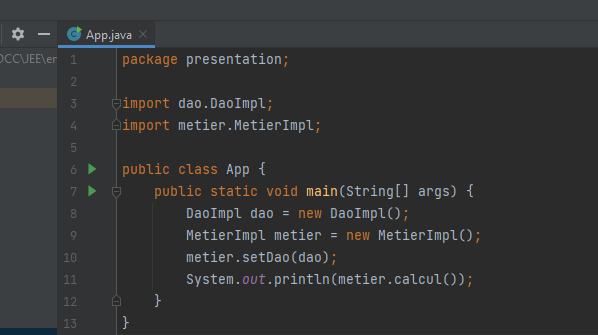
Nous allons créer 2 autres implémentations de l’interface IDao. Nous les mettrons dans un autre package que nous allons nommer **ext**.

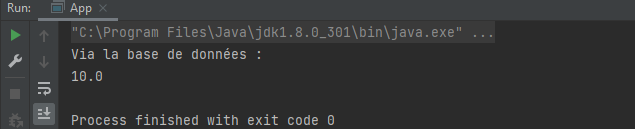




1. Faisons l'injection des dépendances :
2. Par instanciation statique

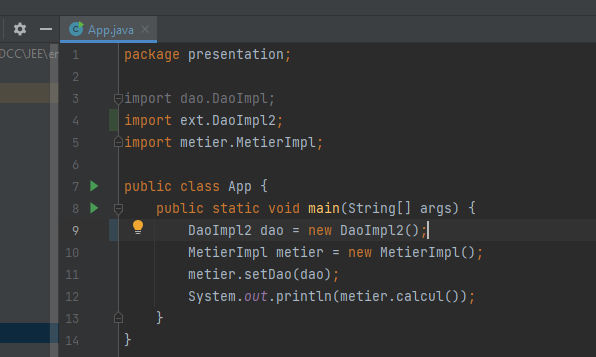
Pour cela nous allons créer une classe **App** dans un package **Présentation**, classe dans laquelle nous allons utiliser la méthode **main.**

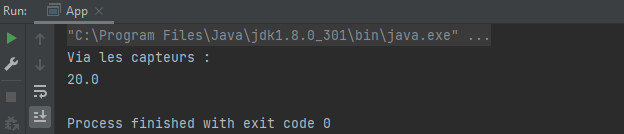




Ici si nous choisissons d’utiliser une des versions, des implémentations de l’interface IDao, nous devons modifier la ligne 8. Le code n’est donc pas fermé à la modification bien qu’ouvert à l’extension.

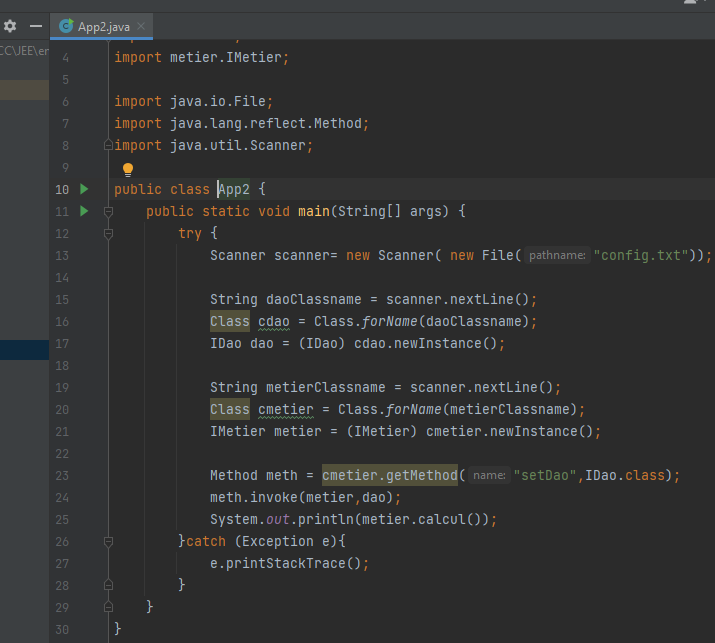
Comme on peut le voir ci-dessous :

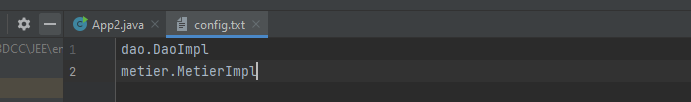




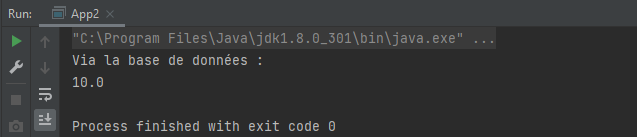
1. Par instanciation dynamique

Nous allons donc passer par une instanciation dynamique pour rendre le code fermé à la modification. Pour cela nous allons créer un fichier de configuration dans lequel nous pourrons préciser l’implémentation IDao que nous souhaitons utiliser. Nous allons définir une classe **App2** dans laquelle nous allons utiliser la méthode **main.**

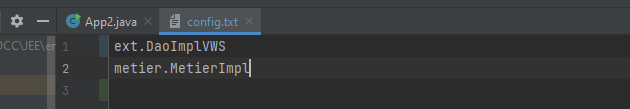


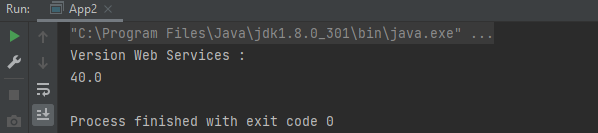


Il s’affichera :



Modifier le fichier de configuration **config.txt** pour utiliser la version Services Web :

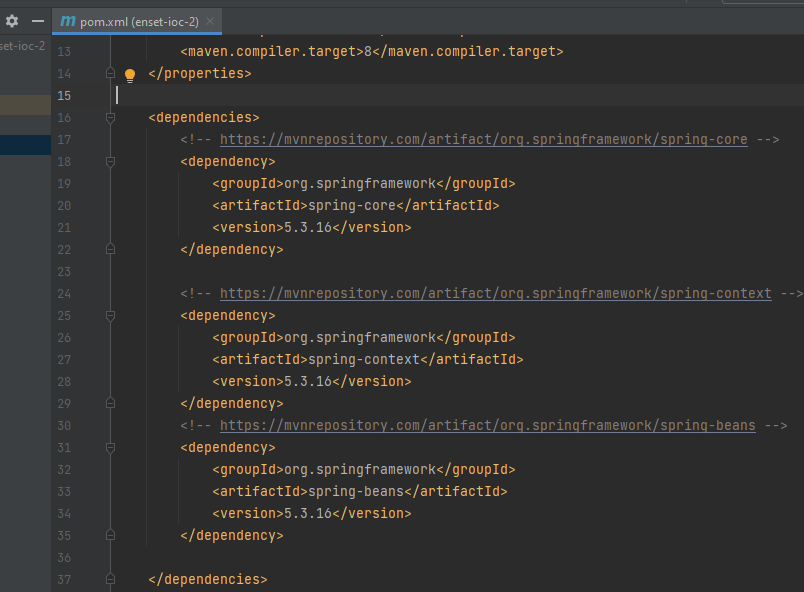




Nous venons ainsi de faire l’injection des dépendances par instanciation statique puis par instanciation dynamique.

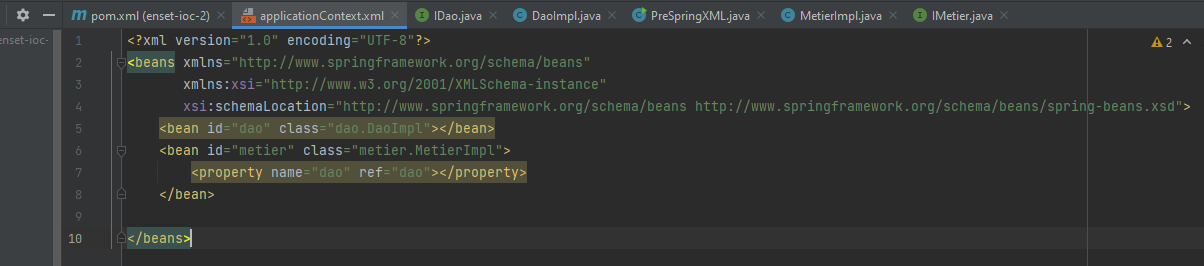
1. En utilisant le Framework Spring

Ici nous allons créer un nouveau projet avec les mêmes packages, classes et interfaces que le précèdent projet. La différence est juste qu’ici nous allons créer un projet **Maven** pour pouvoir ajouter les dépendances nécessaires à l’utilisation du framework **Spring**. Ces dépendances sont ajoutées dans le fichier pom.xml.

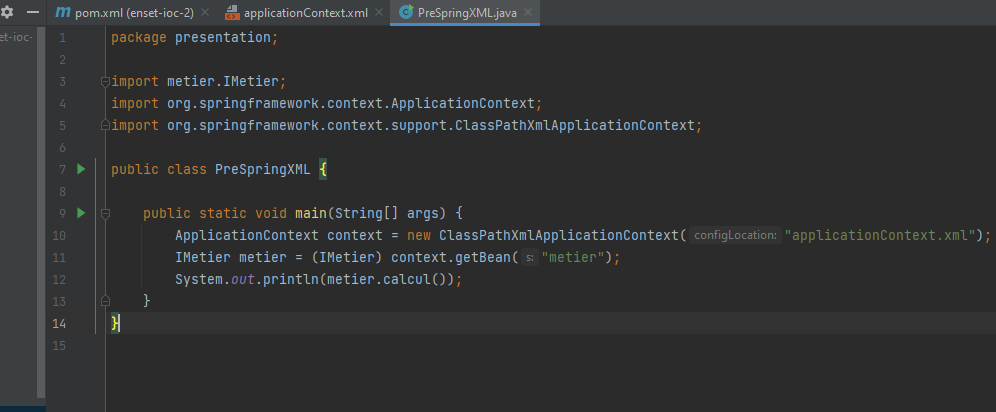


Nous allons faire l’injection des dépendances de 2 façons :

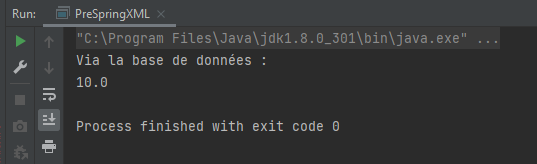
- Version XML

On crée un fichier XML (Spring Config) dans le dossier **resources** et on aura dans le fichier :

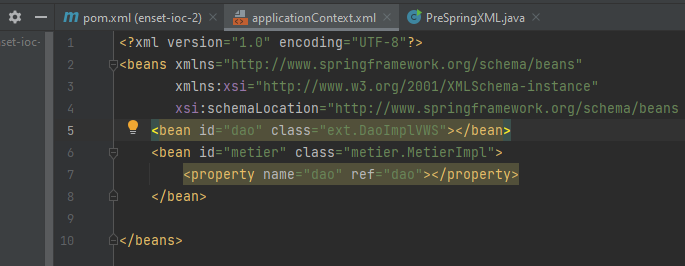
Dans une classe **PreSpringXML** qui contient une méthode **main**, on aura



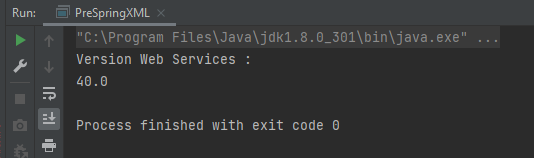
Si on exécute on obtient :



Pour utiliser une autre version, on modifie dans le fichier XML :

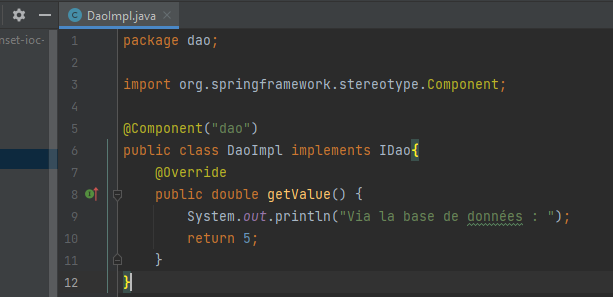


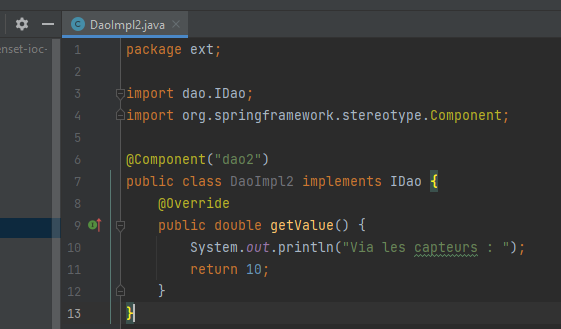
Quand on exécute après cette modification :

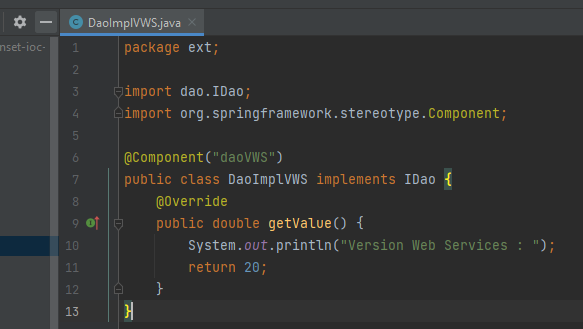


- Version annotations

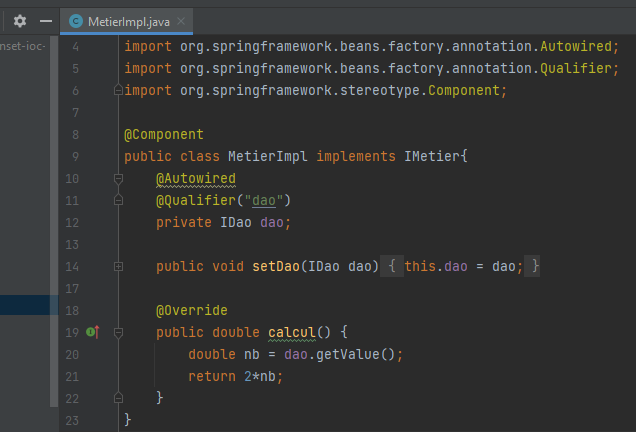
Ici on ajoute l’attribut @Component dans les implémentations de l’interface IDao, comme suit :

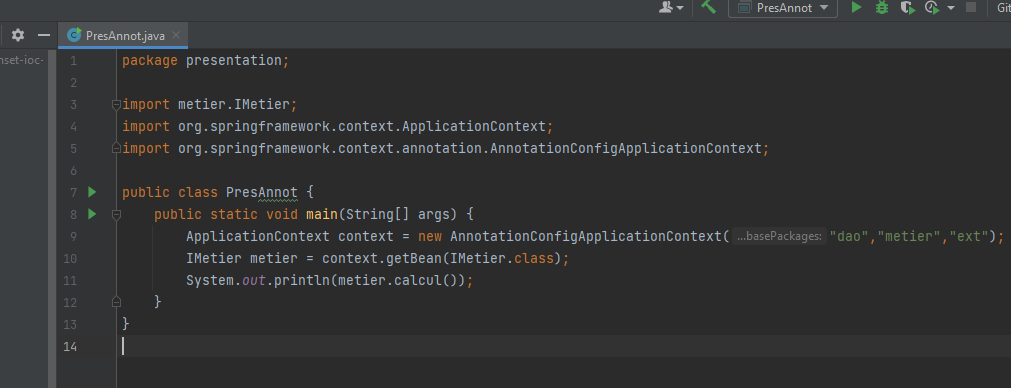




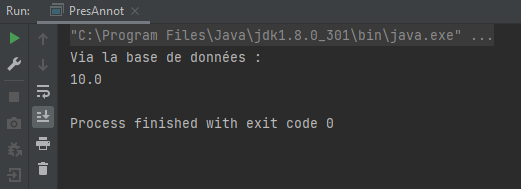


Dans l’implémentation MetierImpl de l’interface on aura :

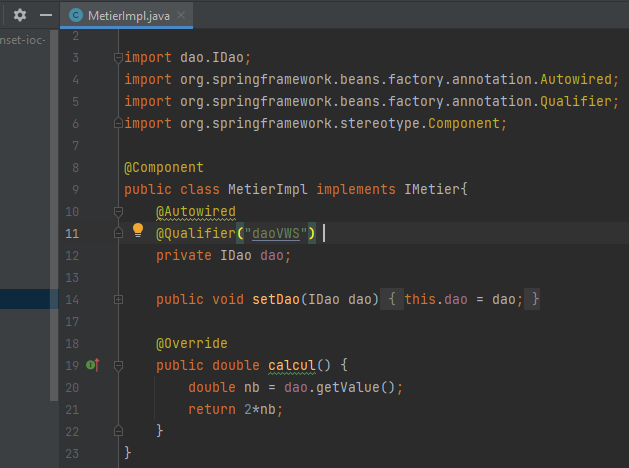


Dans l’attribut @Qualifier on précise l’implémentation IDao que nous souhaitons utiliser. Ici on utilise la version base de données. On va donc créer une classe **PresAnnot** va nous permettre d’exécuter notre programme.

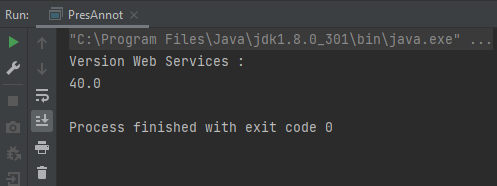
Quand on exécute le programme, on a :



Modifions l’implémentation IDao à utiliser dans @Qualifier de MetierImpl et utilisons la version Web Services. On aura :



Quand on exécute, on obtient :



Nous venons ainsi de faire l’injection des dépendances de deux manières différentes en utilisant le framework Spring.