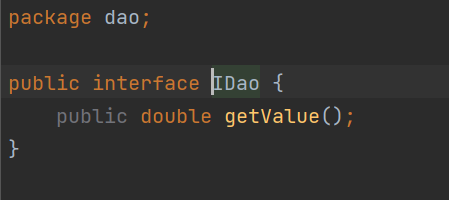
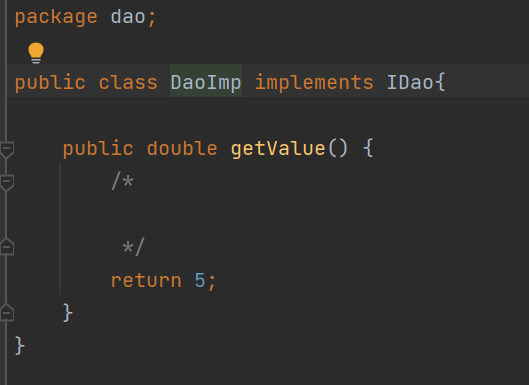
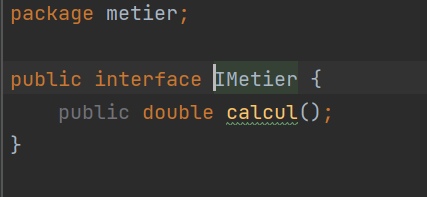
# Compte rendu

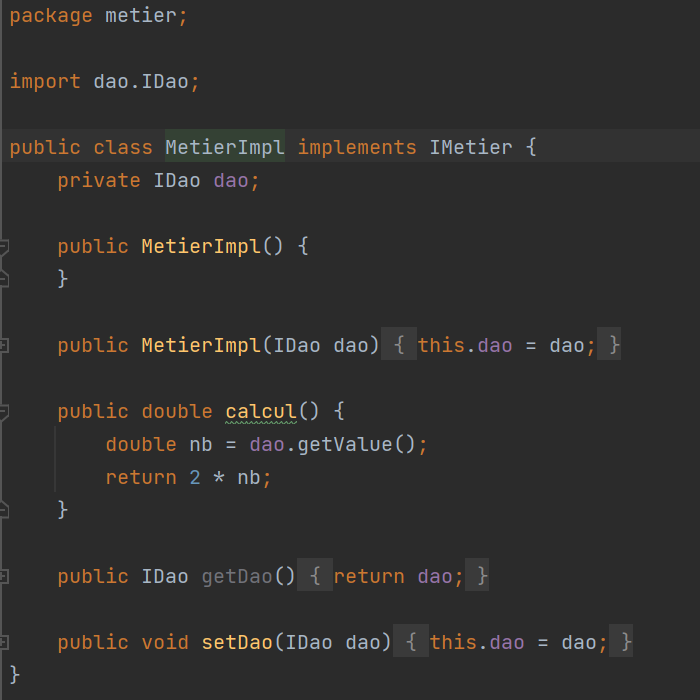
# Inversion de contrôle et Injection des dépendances

Tout d’abord, on a crée la couche dao dont on a crée l’interface IDao qui contient une méthode GetValue(), puis on a crée une implémentation de cette interface, qui fait retourner à chaque fois 5, et en suite la couche metier dont on a crée l’interface IMetier qui contient la méthode calcul(), puis on a crée une implémentation de cette interface qui contient une dépendance de la couche dao, précisément de l’interface IDao.

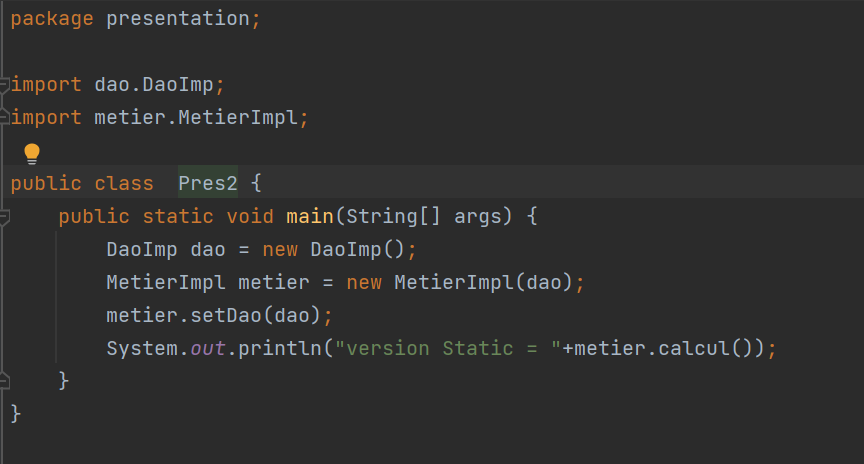




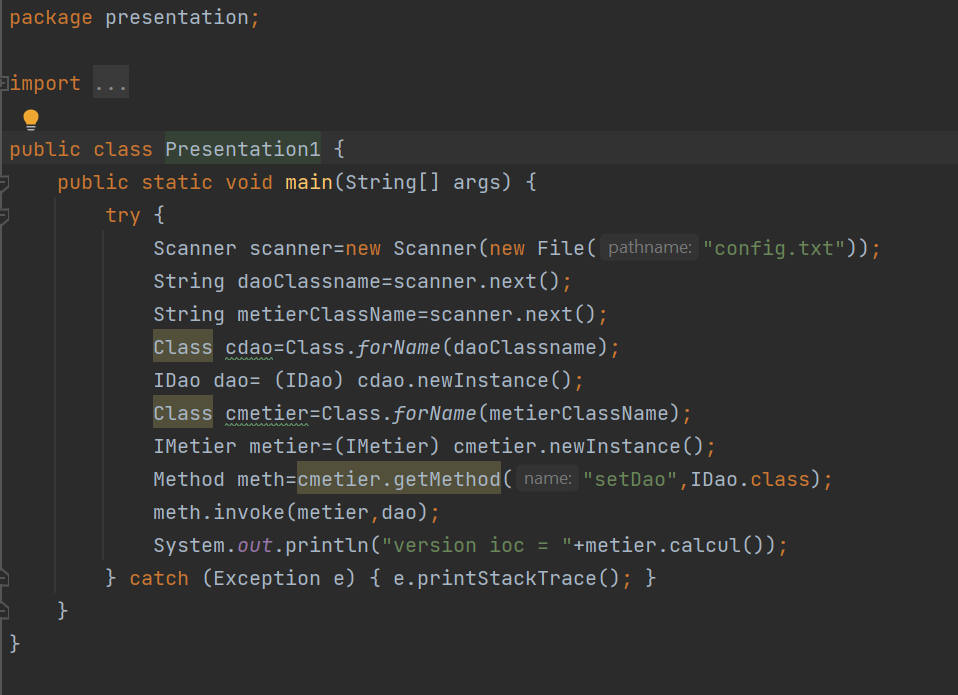


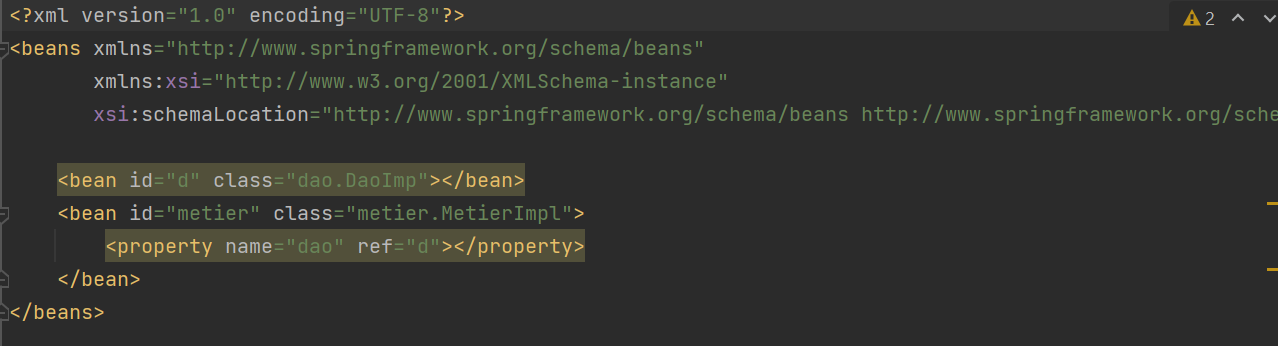


Et pour faire l’injection des dépendances, on va commencer par l’instanciation statique avec la méthode new (), crée une méthode dao et and méthode metier, puis associer ces deux objets. Puis on passe à l’instanciation dynamique, en commençant par créer un fichier de configuration, dans lequel on déclare le nom de la classe qu’on veut utiliser.

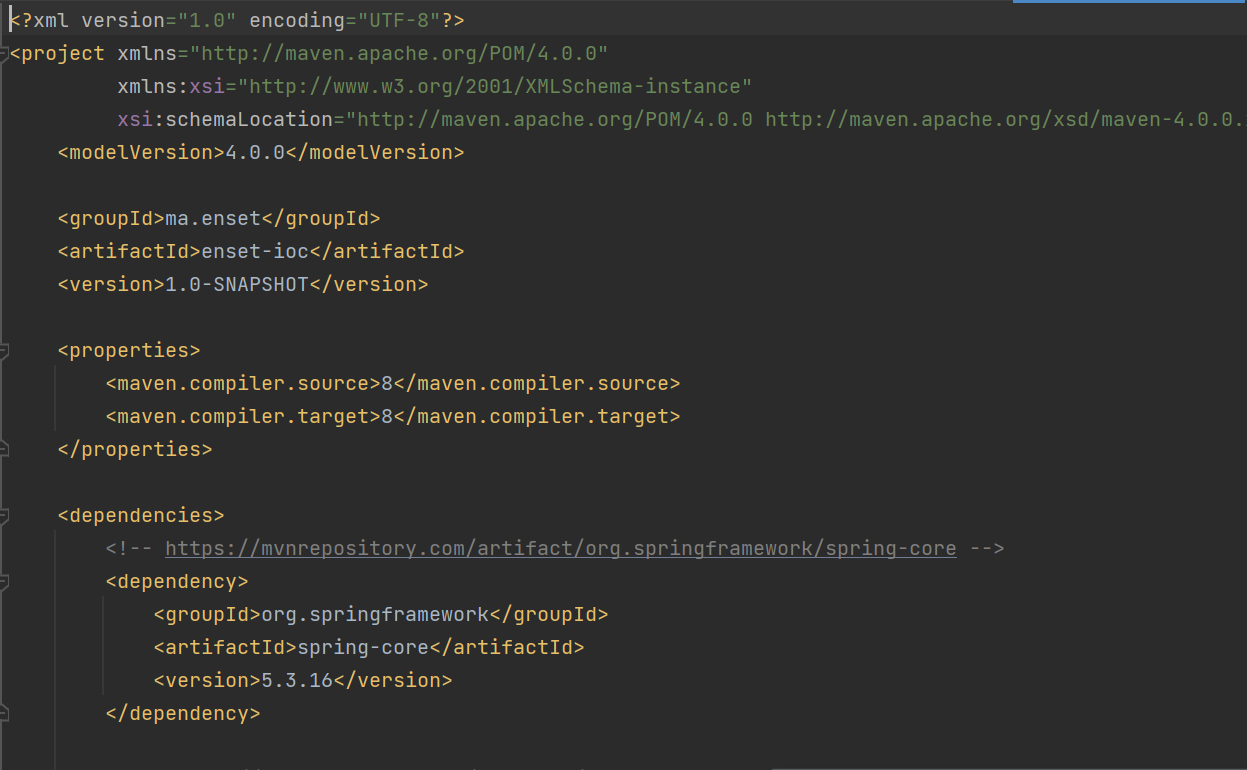


Après, l’application va lire le fichier texte configuration, et puis lire la première ligne du fichier texte, et trouvé le nom de la classe, et la chargé en mémoire, et crée une instance de cette classe, et la même chose pour la deuxième ligne. Et pour faire appel à une méthode dynamique, on crée l’objet méthode, en spécifiant le nom de la méthode, puis on fait appel à la méthode invoke.



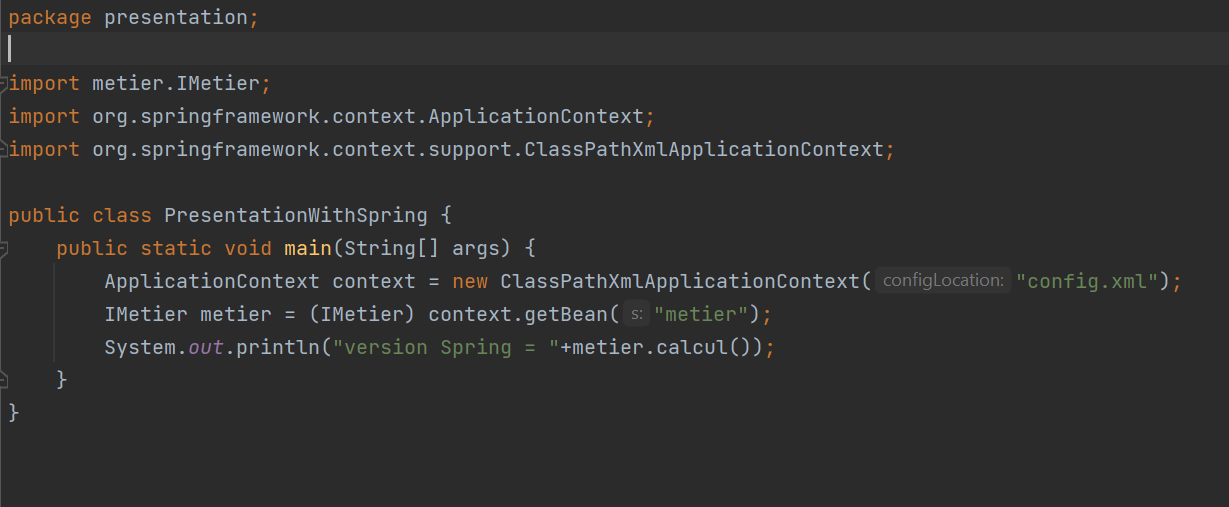
Pour la version XML, on va crée tout d’abord un fichier xml dans le dossier ressources 

Mais avant ça on crée tout d’abord le fichier pom.xml qui est une initialisation de projet Maven, en installant tout les maven spring depandencies qu’on est besoin





Et après on injecte les depandencies dans le code java

Et dés la fin, on va créer une version de l’application présentation spring version annotations, et il faut démarrer l’application avec ma méthode applicationcontext().

