

Rapport N1:

Objectifs:

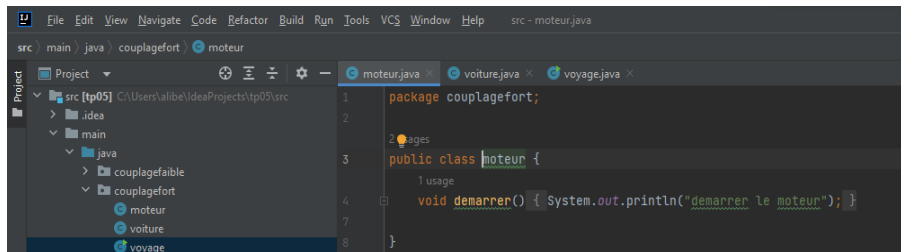
- 1) Différence entre couplage faible et couplage fort en Java
- 2) le role de

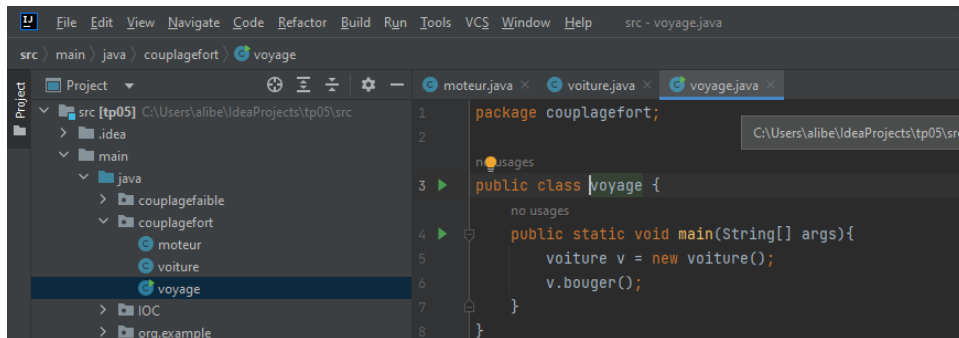
1) Différence entre couplage faible et couplage fort en Java

Couplage fort:

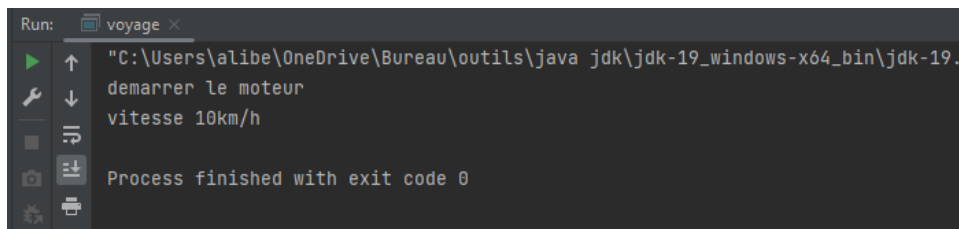
- Un objet fortement couplé est un objet qui a besoin de connaître les autres objets et est généralement très dépendant les uns des autres.
- La modification d'un objet dans une application fortement couplée nécessite souvent de modifier d'autres objets.

EXEMPLE :



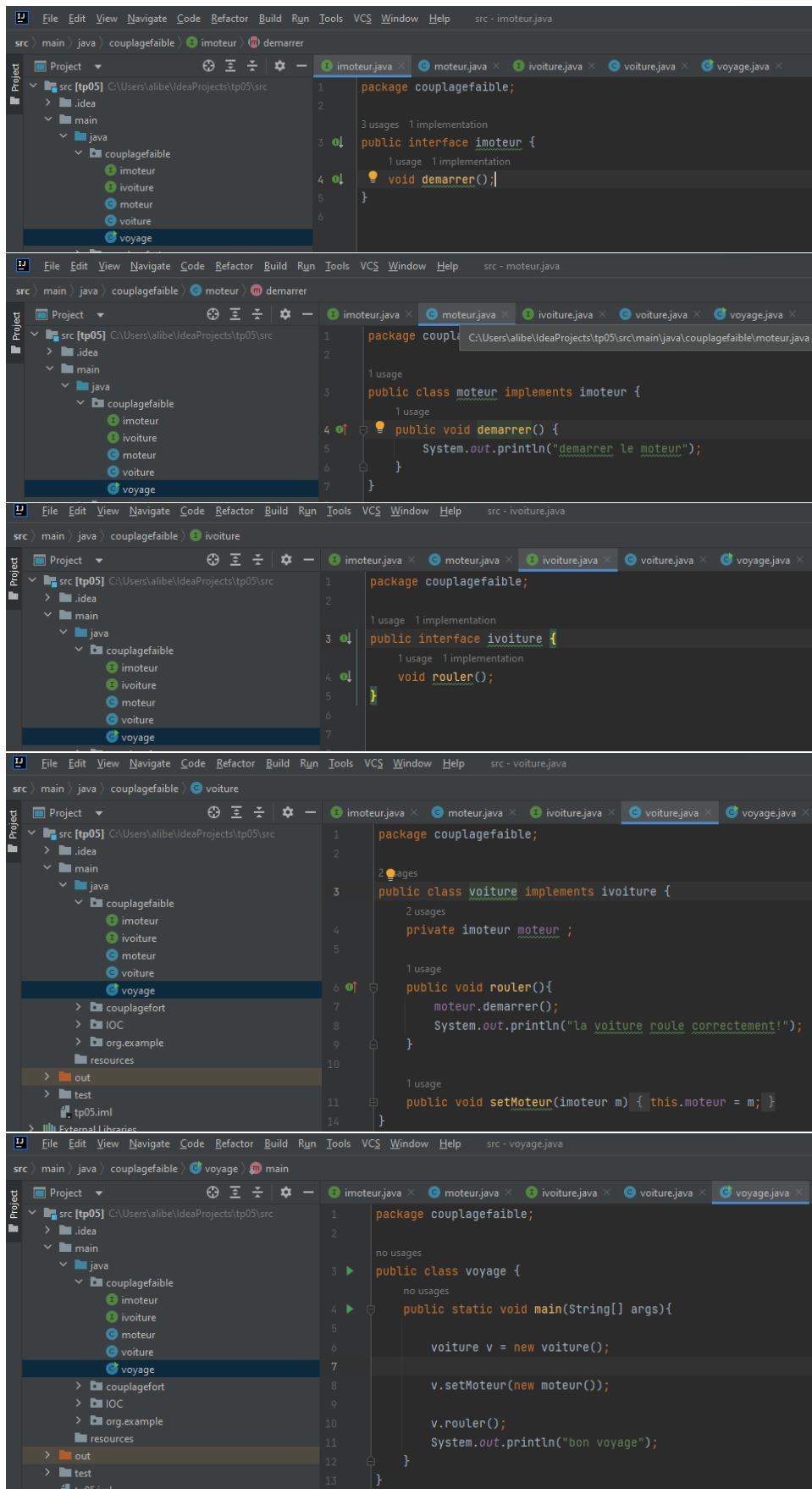


Sortie:

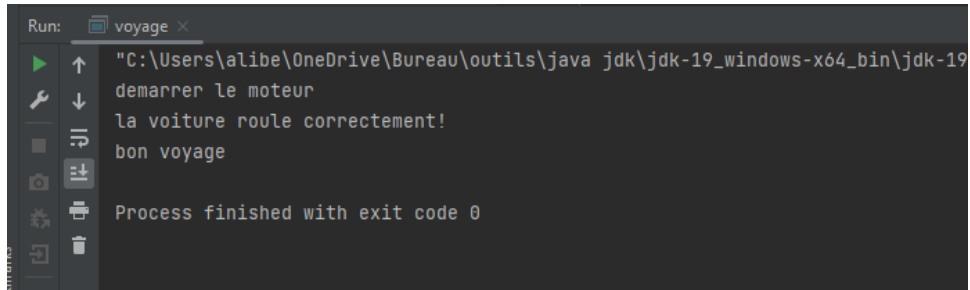


Couplage faible:

- Le couplage faible permet de réduire les interdépendances entre les composants d'un système dans le but de réduire le risque que les changements dans un composant nécessitent des changements dans tout autre composant.
- Le couplage faible est un concept destiné à augmenter la flexibilité du système, à le rendre plus maintenable et à rendre l'ensemble du framework plus stable.



Sortie:

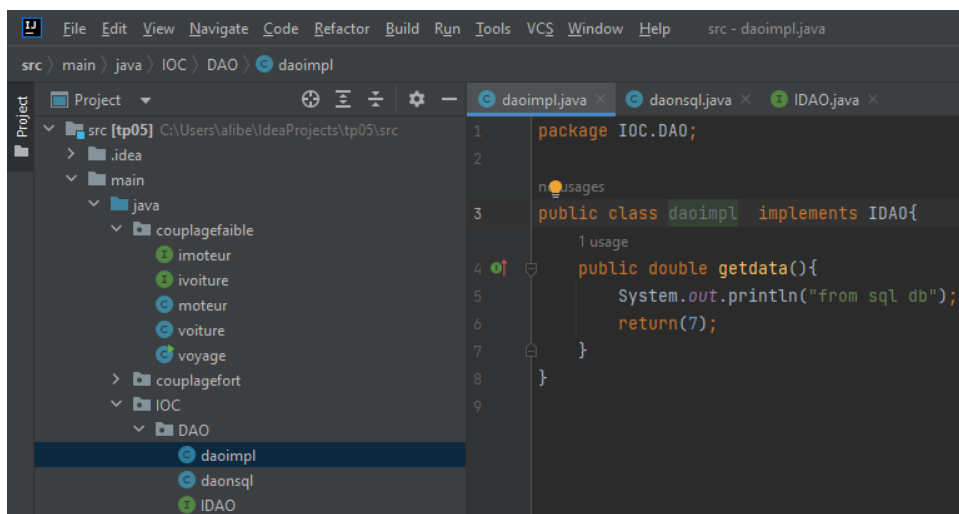


```
Run: voyage x
"C:\Users\alibe\OneDrive\Bureau\outils\java\jdk\jdk-19_windows-x64_bin\jdk-19
demarrer le moteur
la voiture roule correctement!
bon voyage
Process finished with exit code 0
```

2) Injection des dépendances:

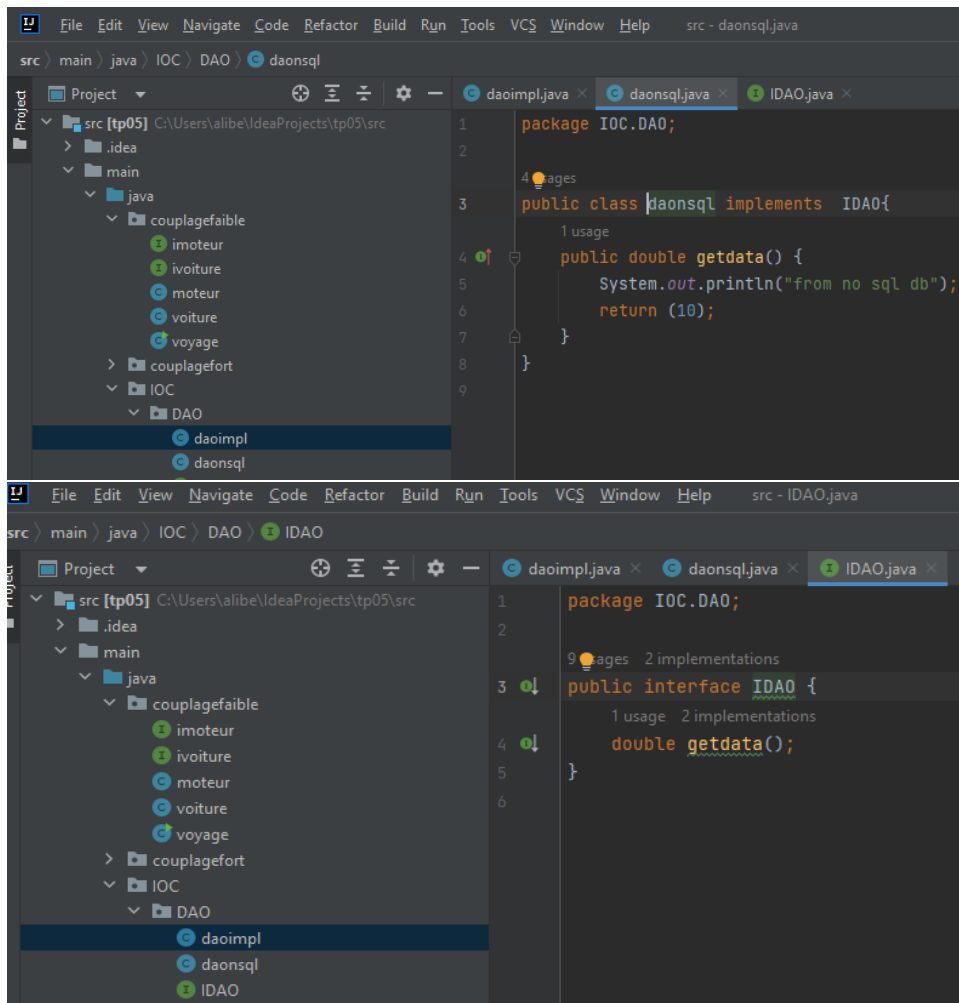
a) Instanciation statique:

DAO:

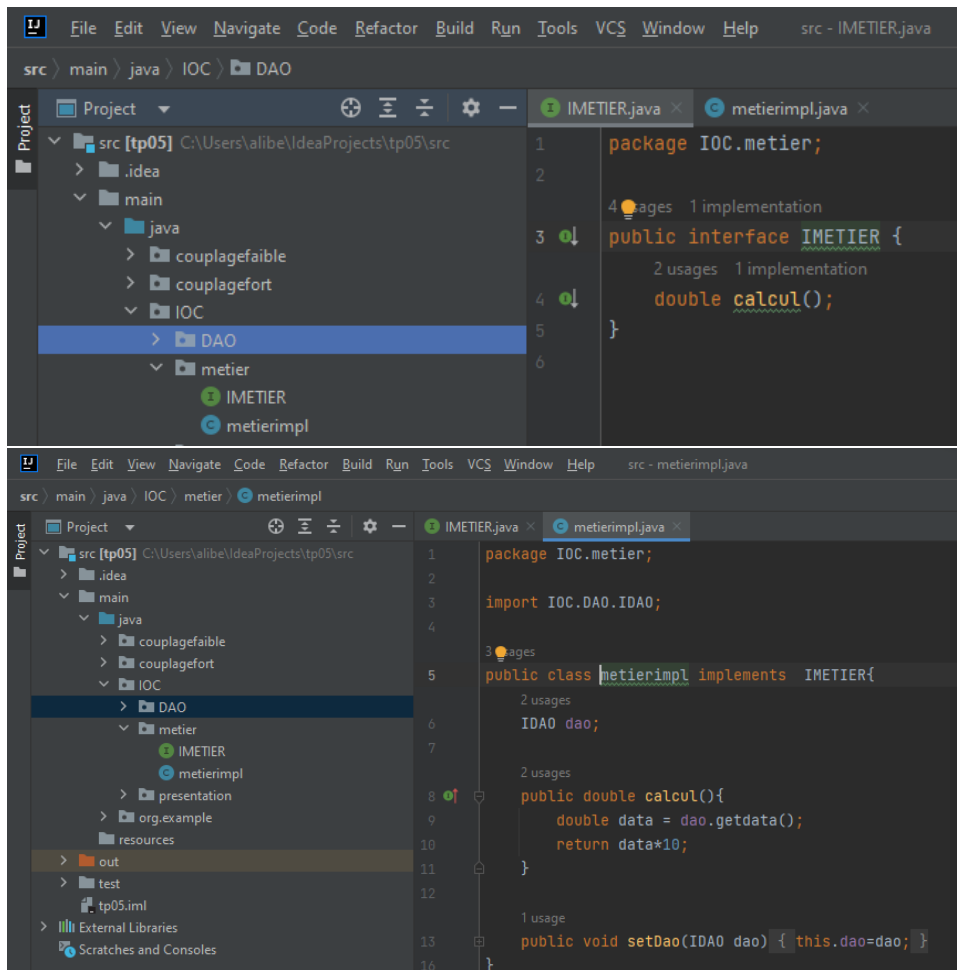


```
src | main | java | IOC | DAO | daoimpl
src | main | java | IOC | DAO | daonsql
src | main | java | IOC | DAO | IDAO

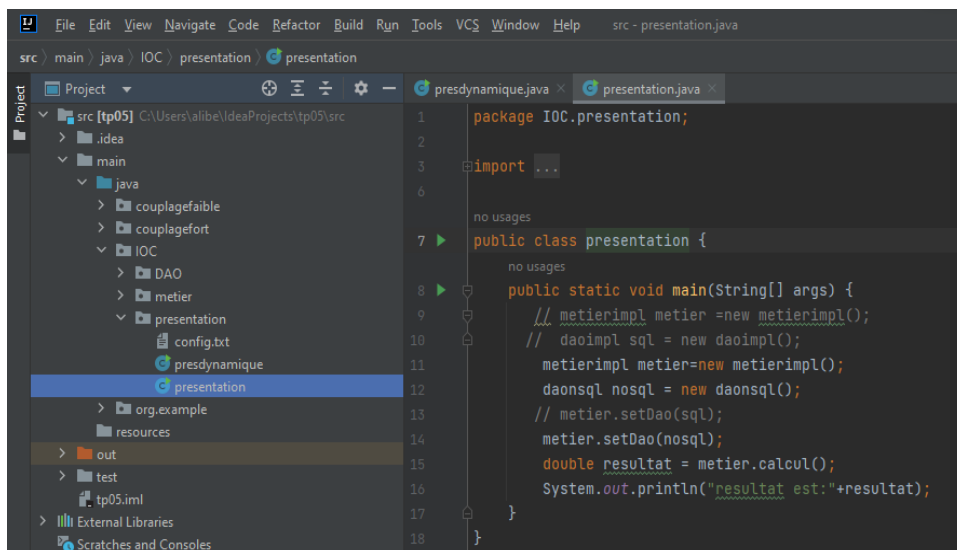
1 package IOC.DAO;
2
3 public class daoimpl implements IDAO{
4     public double getdata(){
5         System.out.println("from sql db");
6         return(7);
7     }
8 }
9
```



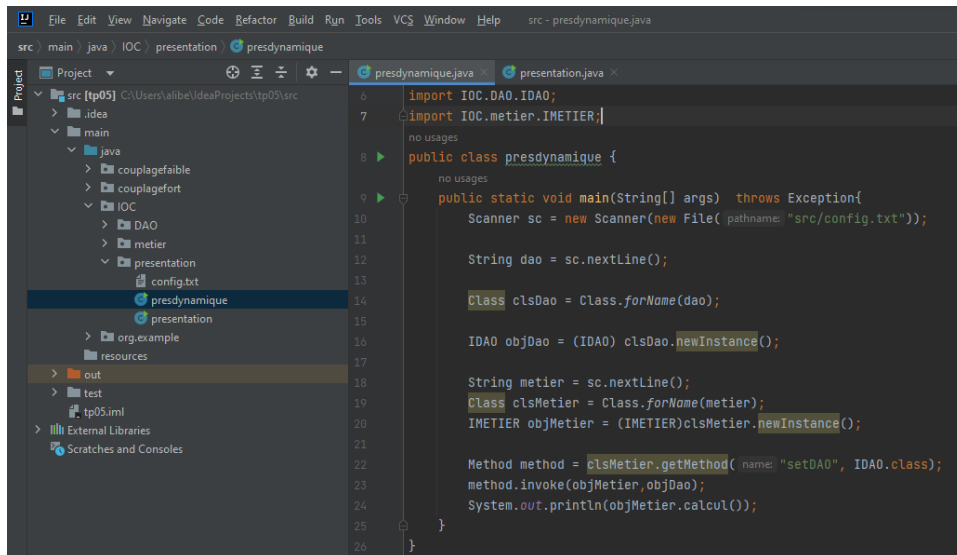
METIER:



PRESENTATION:



a) Instanciation dynamique:



The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a package hierarchy: src/main/java/IOC/presentation. The code editor displays the following Java code:

```
6 import IOC.DAO.IDAO;  
7 import IOC.metier.IMETIER;  
8  
9 public class presdynamique {  
10     public static void main(String[] args) throws Exception{  
11         Scanner sc = new Scanner(new File( pathname: "src/config.txt"));  
12  
13         String dao = sc.nextLine();  
14  
15         Class clsDao = Class.forName(dao);  
16  
17         IDAO objDao = (IDAO) clsDao.newInstance();  
18  
19         String metier = sc.nextLine();  
20         Class clsMetier = Class.forName(metier);  
21         IMETIER objMetier = (IMETIER)clsMetier.newInstance();  
22  
23         Method method = clsMetier.getMethod( name: "setDAO", IDAO.class);  
24         method.invoke(objMetier,objDao);  
25         System.out.println(objMetier.calcul());  
26     }  
27 }
```