



TP4 Compte rendu

Module GL et IL

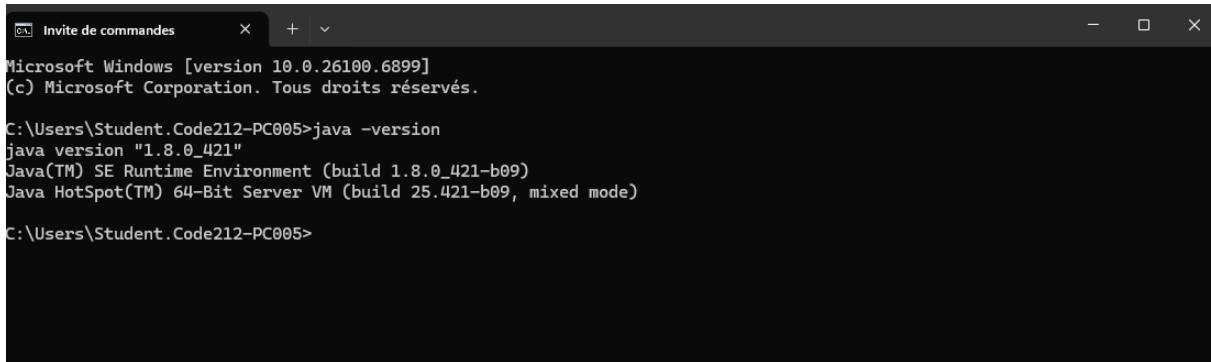
Filière : Ingénierie logicielle

Niveau : Deuxième année

Réalisé par : Moutaoikil Mohamed

Configurations :

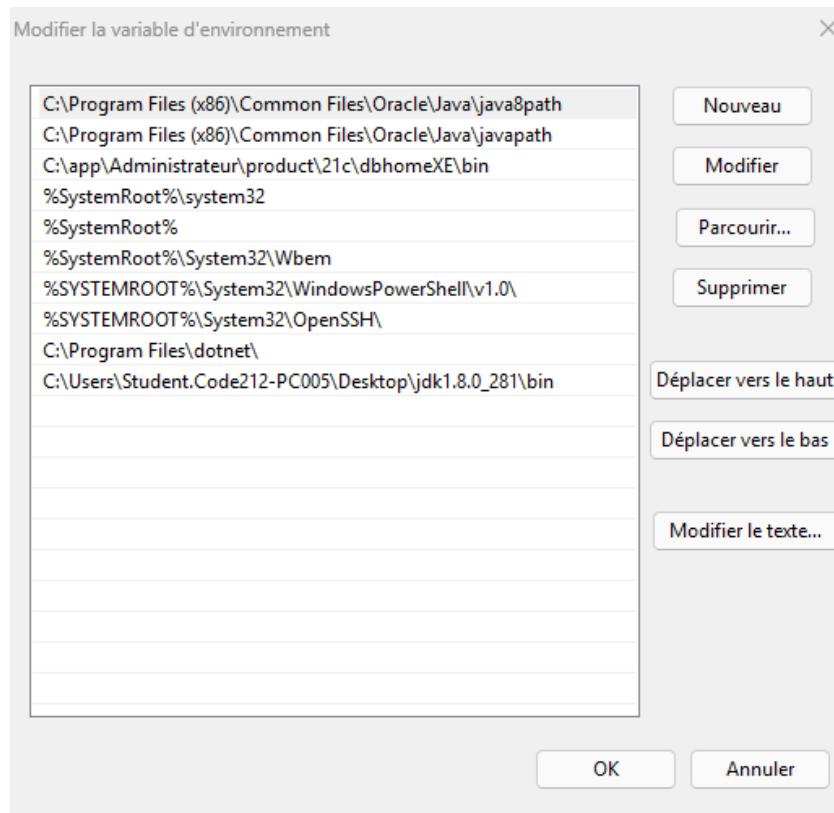
le jdk 1.8 est déjà bien installé et configuré dans la machine , la version est adapté et le variable d' environnement est réglé.



```
Microsoft Windows [version 10.0.26100.6899]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Student.Code212-PC005>java -version
java version "1.8.0_421"
Java(TM) SE Runtime Environment (build 1.8.0_421-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.421-b09, mixed mode)

C:\Users\Student.Code212-PC005>
```



Création des applications Client / Serveur

La première étape est de créer le fichier IDL

On remplit le fichier idl par ce contenu et on compile après

The screenshot shows the Eclipse IDE interface. On the left, the Explorer view displays a project named 'TP4' with a folder 'HelloWorldApp' containing files 'HelloWorldApp.java' and 'HelloWorld.idl'. The 'HelloWorld.idl' file is selected and highlighted in blue. On the right, the editor view shows the contents of the 'HelloWorld.idl' file:

```
1 module HelloWorldApp{
2     interface Hello{
3         attribute string HelloMessage;
4     };
5 };
```

2- Compilation du fichier IDL

Après la compilation on peut visualiser les fichiers suivants est apparu dans le dossier HelloWorldApp , on compile avec la commande suivant :

```
C:\ Invité de commandes x + v
Microsoft Windows [version 10.0.26100.6899]
c) Microsoft Corporation. Tous droits réservés.

:C:\Users\Student.Code212-PC005>jdk --version
jdk' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.

:C:\Users\Student.Code212-PC005>java --version
nrecognized option: --version
rror: Could not create the Java Virtual Machine.
rror: A fatal exception has occurred. Program will exit.

:C:\Users\Student.Code212-PC005>java -version
ava version "1.8.0_421"
ava(TM) SE Runtime Environment (build 1.8.0_421-b09)
ava HotSpot(TM) 64-Bit Server VM (build 25.421-b09, mixed mode)

:C:\Users\Student.Code212-PC005>cd C:\Users\Student.Code212-PC005\Desktop\GénieLogiciel\TP4\HelloWorldApp

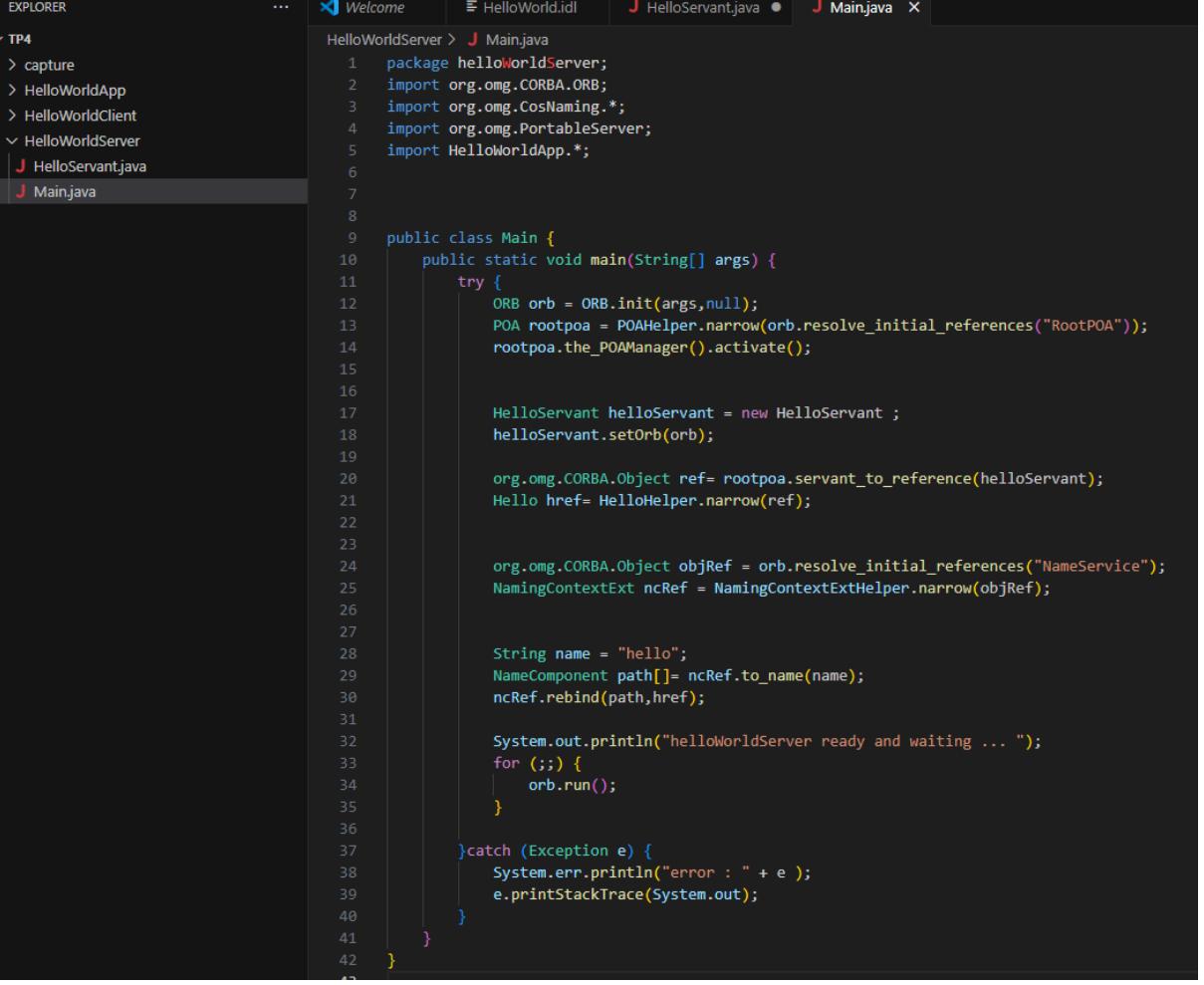
:C:\Users\Student.Code212-PC005\Desktop\GénieLogiciel\TP4\HelloWorldApp>idlj -fall HelloWorld.idl
:C:\Users\Student.Code212-PC005\Desktop\GénieLogiciel\TP4\HelloWorldApp>
```

The screenshot shows the Eclipse IDE interface. On the left, the Explorer view displays a project named 'TP4' with a folder 'HelloWorldApp' containing files '_HelloStub.java', 'Hello.java', 'HelloHelper.java', 'HelloHolder.java', 'HelloOperations.java', and 'HelloPOA.java'. The file '_HelloStub.java' is selected and highlighted in red. On the right, the editor view shows the contents of the 'HelloWorld.idl' file, which is identical to the one shown in the first screenshot.

3- Cration du serveur

On crer un module Client avec une classe main qui va tre remplie par le contenu suivant :

Fichier main et HelloServant



The screenshot shows an IDE interface with the following details:

- EXPLORER** pane on the left lists the project structure:
 - TP4
 - > capture
 - > HelloWorldApp
 - > HelloWorldClient
 - >HelloWorldServer
 - J HelloServant.java
 - J Main.java
- MAIN** pane on the right displays the content of `Main.java`:

```
1 package helloWorldServer;
2 import org.omg.CORBA.ORB;
3 import org.omg.CosNaming.*;
4 import org.omg.PortableServer;
5 import HelloWorldApp.*;
6
7
8
9 public class Main {
10     public static void main(String[] args) {
11         try {
12             ORB orb = ORB.init(args,null);
13             POA rootpoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
14             rootpoa.the_POAManager().activate();
15
16
17             HelloServant helloServant = new HelloServant ;
18             helloServant.setOrb(orb);
19
20             org.omg.CORBA.Object ref= rootpoa.servant_to_reference(helloServant);
21             Hello href= HelloHelper.narrow(ref);
22
23
24             org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
25             NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);
26
27
28             String name = "hello";
29             NameComponent path[] = ncRef.to_name(name);
30             ncRef.rebind(path,href);
31
32             System.out.println("helloWorldServer ready and waiting ... ");
33             for (;;) {
34                 orb.run();
35             }
36
37         }catch (Exception e) {
38             System.err.println("error : " + e );
39             e.printStackTrace(System.out);
40         }
41     }
42 }
```

3- Cration du Client:

On crer un module Serveur, et un fichier main et class Serveur qui ont tre remplie par le contenu suivant :

Fichier Main

```
package helloWorldClient;

import org.omg.CORBA.ORB;
import org.omg.CosNaming.*;
import HelloWorldApp.*;

public class Main {
    static Hello hello;

    public static void main(String[] args) {
        try {
            ORB orb = ORB.init(args, null);

            org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
            NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);
            String name = "Hello";
            hello = HelloHelper.narrow(ncRef.resolve_str(name));
            System.out.println("Obtained a handle on server object : " + hello );
            System.out.println(hello.HelloMessage());
        } catch(Exception e) {
            System.out.println("ERROR : " + e);
            e.printStackTrace(System.out);
        }
    }
}
```

6- Démarrage du service ORB et lancement de deux Programmes:

on lance le port par la commande ci dessous , après on test les deux programme et comme résultat le serveur est en attente

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Student.Code212-PC005\Desktop\GénieLogiciel\TP4> &
005\AppData\Roaming\Code\User\workspaceStorage\916b54928e052f782
helloWorldServer ready and waiting ...
```