

# TP Corba

Master MaDSI

---

*“L'architecture CORBA (Common Object Request Broker Architecture) est une norme développée par OMG (Object Management Group) pour assurer l'interopérabilité entre les objets distribués. CORBA est la première solution middleware au monde permettant l'échange d'informations, indépendamment des plateformes matérielles, des langages de programmation et des systèmes d'exploitation. CORBA est essentiellement une spécification de conception pour un ORB (Object Request Broker), un ORB fournissant le mécanisme nécessaire pour que les objets distribués puissent communiquer entre eux, que ce soit localement ou sur des périphériques distants, écrits sous différents langages de programmation ou situés à différents endroits d'un réseau.”*

\*\*\*

Dans ce TP sur CORBA, nous allons apprendre à créer:

- Objet distant simple.
- Serveur pour instancier (créer) et lier un objet distant.
- Client pour appeler un objet à distance

Vous devez installer le kit de développement Java (JDK) ainsi qu'un éditeur de programmation (IDE) tel que Eclipse. Pour plus d'informations sur l'installation de JDK et d'Eclipse, utilisez le lien suivant :

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.

## Côté serveur

### Étape 1

Créer un nouveau projet Java avec Eclipse (ou NetBeans ou un autre éditeur de votre choix) et appelez-le: CorbaAdditionServer-> Cliquez sur Terminer une fois terminé.

### Étape 2

Sous le projet CorbaAdditionServer, créer un nouveau fichier appelé Addition.idl. Copiez le code ci-dessous dans le nouveau fichier.

module AdditionApp

{

interface Addition

{

long add(in long a,in long b);

oneway void shutdown();

};

};

### Étape 3

Ouvrez votre console CMD, changer le répertoire (cd) dans le dossier src à l'emplacement du projet.

### Étape 4

Compilez le fichier idl en utilisant la commande idlj comme ci-dessous:

```
idlj -fall Addition.idl
```

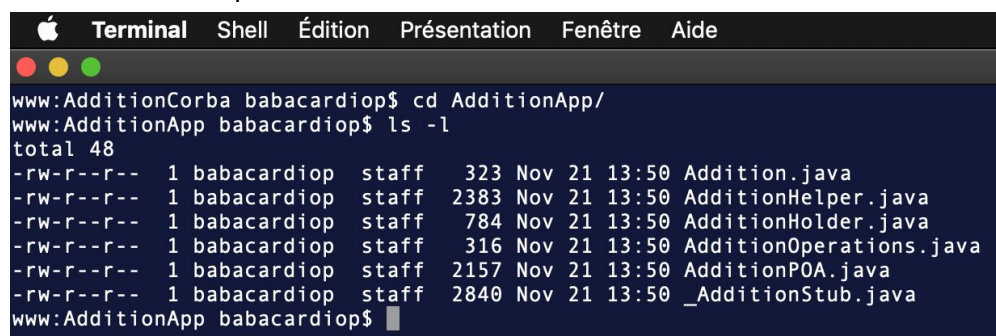
Vous verrez que dans le dossier src, un nouveau répertoire appelé **AdditionApp** est créé avec un certain nombre de fichiers Java.

Entrer dans le répertoire **AdditionApp** et afficher son contenu.

Commandes :

```
cd AdditionApp
ls -l
```

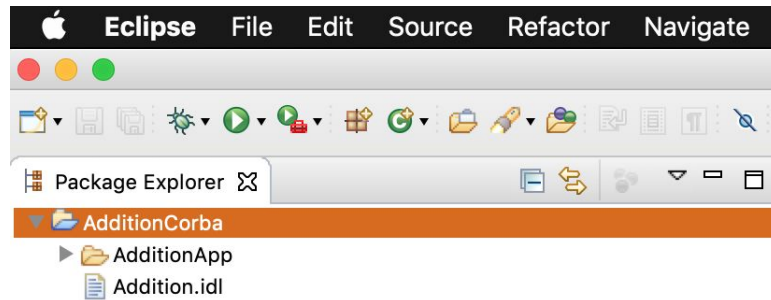
Voici le contenu du répertoire.



```
Terminal  Shell  Édition  Présentation  Fenêtre  Aide
www:AdditionCorba babacardiop$ cd AdditionApp/
www:AdditionApp babacardiop$ ls -l
total 48
-rw-r--r--  1 babacardiop  staff   323 Nov 21 13:50 Addition.java
-rw-r--r--  1 babacardiop  staff  2383 Nov 21 13:50 AdditionHelper.java
-rw-r--r--  1 babacardiop  staff   784 Nov 21 13:50 AdditionHolder.java
-rw-r--r--  1 babacardiop  staff   316 Nov 21 13:50 AdditionOperations.java
-rw-r--r--  1 babacardiop  staff  2157 Nov 21 13:50 AdditionPOA.java
-rw-r--r--  1 babacardiop  staff  2840 Nov 21 13:50 _AdditionStub.java
www:AdditionApp babacardiop$
```

### Étape 5

Rafraîchir la liste des éléments du projet créé en faisant click droit sur le répertoire source de votre projet sur Eclipse, et cliquer sur “Refresh”.



## Étape 6

Sous le projet CorbaAdditionServer, créer une nouvelle classe appelée AdditionObj. Copiez le code ci-dessous dans la nouvelle classe.

```
import AdditionApp.*;

import org.omg.CosNaming.*;

import org.omg.CosNaming.NamingContextPackage.*;

import org.omg.CORBA.*;

import org.omg.PortableServer.*;

import org.omg.PortableServer.POA;

import java.util.Properties;

class AdditionObj extends AdditionPOA {

    private ORB orb;

    public void setORB(ORB orb_val) {

        orb = orb_val;

    }
```

```
// implement add() method
```

```
public int add(int a, int b) {
```

```
    int r=a+b;
```

```
    return r;
```

```
}
```

```
// implement shutdown() method
```

```
public void shutdown() {
```

```
    orb.shutdown(false);
```

```
}
```

```
}
```

## Étape 7

Sous le projet CorbaAdditionServer, créer une nouvelle classe appelée StartServer. Copier le code ci-dessous dans la nouvelle classe.

```
import AdditionApp.*;
```

```
import org.omg.CosNaming.*;
```

```
import org.omg.CosNaming.NamingContextPackage.*;
```

```
import org.omg.CORBA.*;
```

```
import org.omg.PortableServer.*;
```

```
import org.omg.PortableServer.POA;
```

```
import java.util.Properties;
```

```
public class StartServer {
```

```
    public static void main(String args[]) {
```

```
        try{
```

```
// create and initialize the ORB //// get reference to rootpoa & activate the POAManager
```

```
ORB orb = ORB.init(args, null);
```

```
POA rootpoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
```

```
rootpoa.the_POAManager().activate();
```

```
// create servant and register it with the ORB
```

```
AdditionObj addobj = new AdditionObj();
```

```
addobj.setORB(orb);
```

```
// get object reference from the servant
```

```
org.omg.CORBA.Object ref = rootpoa.servant_to_reference(addobj);
```

```
Addition href = AdditionHelper.narrow(ref);
```

```
org.omg.CORBA.Object objRef = orb.resolve_initial_references("NameService");
```

```
NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);
```

```
NameComponent path[] = ncRef.to_name( "ABC" );
```

```
ncRef.rebind(path, href);
```

```
System.out.println("Addition Server ready and waiting ...");
```

```
// wait for invocations from clients
```

```
for (;;) {
```

```
    orb.run();
```

```
}
```

```
}
```

```
catch (Exception e) {
```

```
    System.err.println("ERROR: " + e);
```

```
    e.printStackTrace(System.out);
```

```
}
```

```
System.out.println("HelloServer Exiting ...");
```

```
}
```

```
}
```

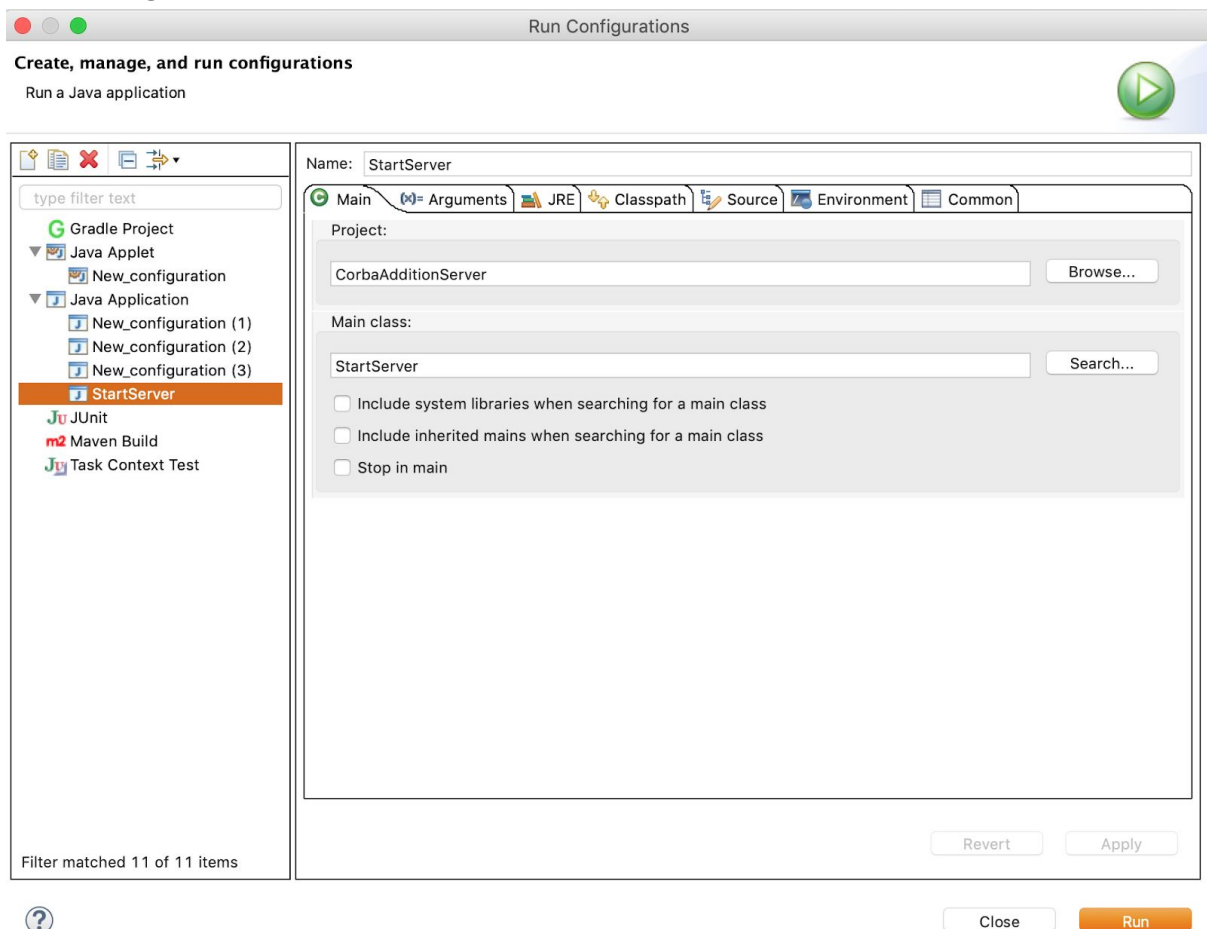
## Étape 8

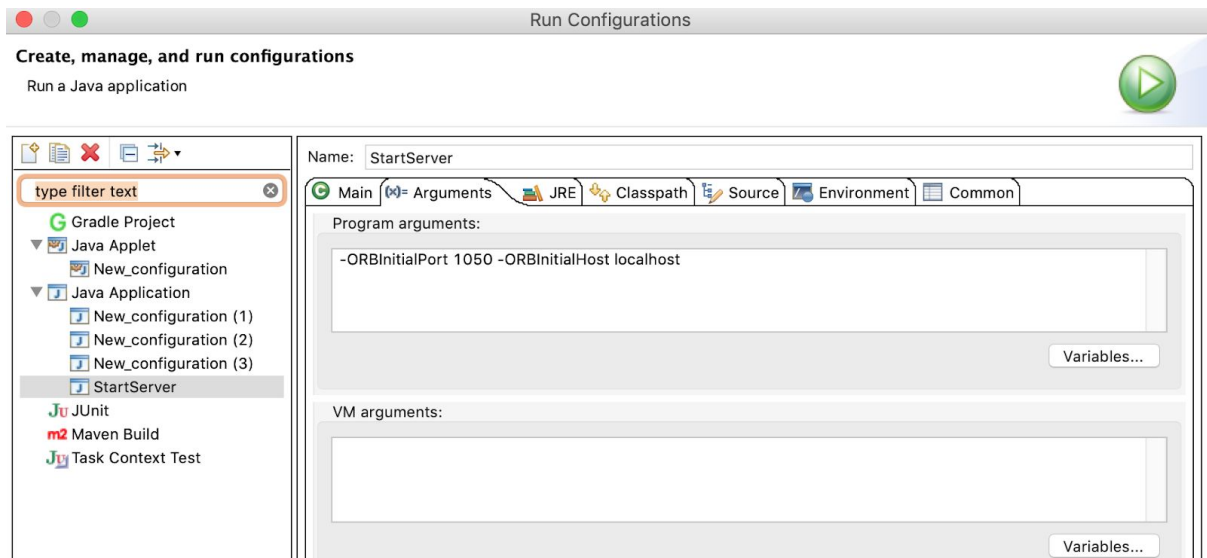
Ouvrir une nouvelle fenêtre de la console CMD, et taper la commande suivante pour démarrer l'ORB.

**start orbd -ORBInitialPort 1050**

## Étape 9

Sous Eclipse maintenant, cliquer sur le projet CorbaAdditionServer, puis cliquer sur **Run -> Run Configurations** comme indiqué ci-dessous:





## Étape 10

Cliquez sur **Arguments** et tapez les arguments suivants dans les arguments du programme. Cliquez sur **Appliquer** ou “*Apply*” une fois terminé:

**-ORBInitialPort 1050 -ORBInitialHost localhost**

Voici le résultat obtenu si tout se passe bien



## Côté client

### Étape 1

Créer un nouveau projet Java avec Eclipse et appeler-le: CorbaAdditionClient-> Cliquer sur “Finish” une fois terminé.

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: CorbaAdditionClien

☒ Use default location

Location: /Users/babacardiop/Documents/REST\_Tutorial/CorbaAdditionClien Browse...

JRE

☒ Use an execution environment JRE: JavaSE-1.8

☐ Use a project specific JRE: Java SE 8 [1.8.0\_91]

☐ Use default JRE (currently 'Java SE 8 [1.8.0\_91]') [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets New...

Working sets: Select...

? < Back Next > Cancel Finish

## Étape 2

Cliquer avec le bouton droit sur Package AdditionApp sous l'ancien projet coté serveur (CorbaAdditionServer), puis cliquer sur Copier.

Ensuite, faire un clic droit sur **src** sous CorbaAdditionClient, puis cliquer sur Coller.

## Étape 3

Sous le projet CorbaAdditionClient, créer une nouvelle classe appelée StartClient. Copier le code ci-dessous dans la nouvelle classe.

```
import AdditionApp.*;

import org.omg.CosNaming.*;

import org.omg.CosNaming.NamingContextPackage.*;

import org.omg.CORBA.*;
```



```

import java.io.*;

import java.util.*;

public class StartClient {

    /**
     * @param args the command line arguments
     */

    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);

            Addition addobj = (Addition) AdditionHelper.narrow(ncRef.resolve_str("ABC"));

            Scanner c=new Scanner(System.in);

            System.out.println("Welcome to the addition system:");

            for(;;){

                System.out.println("Enter a:");

                String aa = c.nextLine();

                System.out.println("Enter b:");

                String bb = c.nextLine();

                int a=Integer.parseInt(aa);

                int b=Integer.parseInt(bb);

                int r=addobj.add(a,b);

                System.out.println("The result for addition is : "+r);

                System.out.println("-----");

            }

        }
    }
}

```

```

    }

    catch (Exception e) {

        System.out.println("Hello Client exception: " + e);

        e.printStackTrace();

    }

}

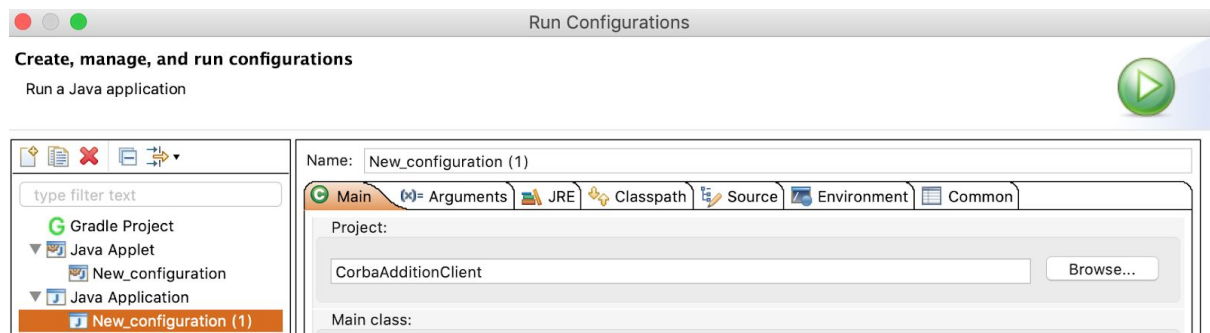
}

}

```

## Étape 4

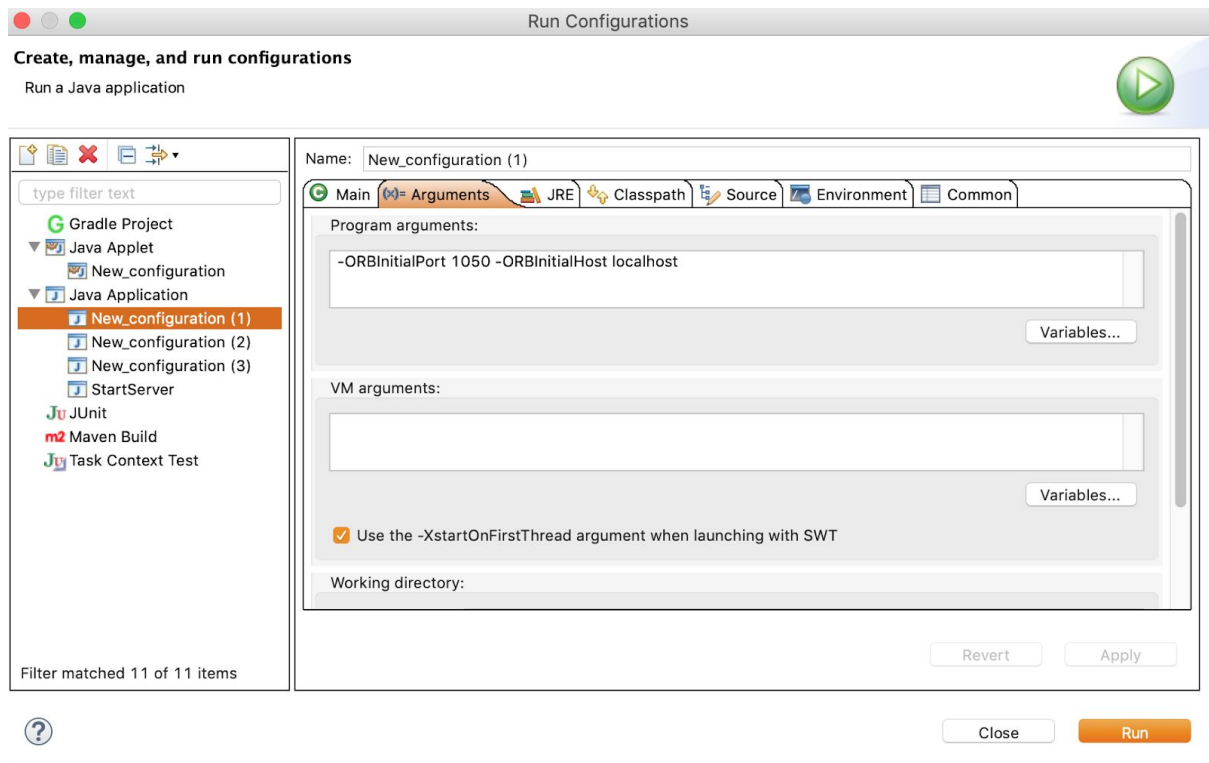
Cliquez sur **Run -> Run Configuration**, tout en sélectionnant le StartClient pour le CorbaAdditionClient.



## Étape 5

Cliquer sur Arguments, puis sous les arguments du programme, taper ce qui suit:

**-ORBInitialPort 1050 -ORBInitialHost localhost**



Cliquez sur Appliquer, puis sur Exécuter. C'est tout pour le client.

Voici un exemple de test pour le client.

