Exemple CORBA
http://java.sun.com/docs/books/tutorial/idl/hello/index.html

Avant toute chose, il faut initialiser java : **`. /home/brdewach/pub/initjava`**

1. Serveur

Une specification de l'interface serveur se fait à l'aide du langage IDL dont voici un exemple simple :

```
module HelloApp {
  interface Hello {
    string sayHello();
  };
};
```

Pour le transformer en classes Java : **`idlj -fall -oldImplBase <fichier>.idl`**
Ceci créera pour l'exemple plus haut les fichiers HelloHelper.java HelloHolder.java _HelloImplBase.java Hello.java HelloOperations.java et _HelloStub.java.

Le serveur doit alors donner une implémentation de cette interface :

```
class HelloServant extends _HelloImplBase {
  public String sayHello() {
    return "\nHello world!!\n";
  }
}
```

Pour activer cette implémentation, le serveur doit le connecter à 'lORB avec :

```
// Create and initialize the ORB
ORB orb = ORB.init(args, null);
// Create the servant and register it with the ORB
HelloServant helloRef = new HelloServant();
orb.connect(helloRef);
```

Pour *publier* cette implémentation, le serveur peut utiliser le naming service :

```
// Get the root naming context
org.omg.CORBA.Object objRef =
orb.resolve_initial_references("NameService");
NamingContext ncRef = NamingContextHelper.narrow(objRef);
// Bind the object reference in naming
NameComponent nc = new NameComponent("Hello", "");
NameComponent path[] = {nc};
ncRef.rebind(path, helloRef);
```

Puis la tâche du serveur consiste à attendre les appels venant de l'ORB avec :

```
java.lang.Object sync = new java.lang.Object();
synchronized(sync){
  sync.wait();
}
```

2. Client

Le client, utilise le même méchanisme pour obtenir une référence vers l'interface *publié* par le serveur

```
// Resolve the object reference in naming
NameComponent nc = new NameComponent("Hello", "");
NameComponent path[] = {nc};
Hello helloRef = HelloHelper.narrow(ncRef.resolve(path));
```

Le client peut alors simplement faire un appel à `helloRef.sayHello()`.

3. Naming Service

Pour que le naming service fonctionne, il faut le lancer avec :

```
tnameserv -ORBInitialPort 2000 &
```

3. Démarrer l'application

```
java HelloServer -ORBInitialPort 2000
java HelloClient -ORBInitialPort 2000
```