



TP N^o 3

—
Architectures des Systèmes Distribués
Systèmes Distribués avec CORBA
—

Le but de ce TP est de mettre en place une application distribuée qui se base sur le protocole CORBA.

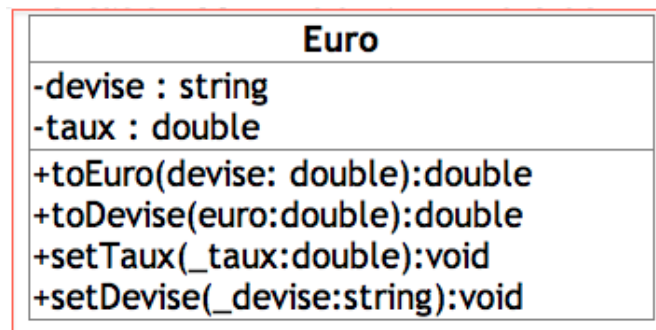
Exercice 1.

Reprendre l'exemple de l'application répartie **Hello World** proposer dans le cours.

Question 1.1. Proposer un jeux de testes.

Exercice 2.

On se propose de faire une implémentation CORBA d'un objet Euro défini par le diagramme UML suivant :



Question 2.1. écrire la description **IDL** du convertisseur et effectuer la projection en Java.

Cette projection crée un répertoire contenant un répertoire convertisseur correspondant au module défini en IDL. Ce répertoire contient les fichiers générés suivants :

- **Euro.java** : définition de l'interface Euro.
- **EuroHelper.java** : méthodes pour faciliter l'utilisation des objets (lecture d'un objet, conversion avec la méthode narrow, ...Euro).

- **EuroHolder.java** : méthodes pour prendre en charge le passage de paramètres avec les méthodes de l'objet Euro.
- **EuroOperations.java** : interface présentant les méthodes de l'objet Euro (Ce dernier hérite de cette interface).
- **_EuroStub.java** : stub pour l'objet Euro.

Question 2.2. Définir la classe de l'objet que nous voulons mettre à disposition des clients.

- On implémente l'objet serveur en utilisant la classe **EuroPOA**.
- Créez une nouvelle classe **EuroPOAImpl** (et donc un nouveau fichier EuroPOAImpl.java).

Question 2.3. Réaliser l'implémentation du serveur CORBA.

- On définit ici une application principale contenant un objet Euro.
- Créez une nouvelle classe **ServerPOA** (et donc un nouveau fichier ServerPOA.java).

Question 2.4. Réaliser l'implémentation du client CORBA.

- Afin de récupérer la référence sur l'objet utilisé, le client va utiliser le fichier ObjectId créé par le serveur CORBA.
- Créez une nouvelle classe **Client** (et donc un nouveau fichier Client.java).