

Exercice 1 – Généralités CORBA

1) Dans les approches RPC (et donc comme celle utilisée dans CORBA) on utilise des souches ou talons (ou encore en anglais stubs, skeletons,...). Quels sont les rôles assignés aux souches ?

Transposer les donner dans un codage commun (assurer l'interopérabilité). Il assurent aussi une partie de la communication réseau.

2) CORBA indique, dans ces objectifs, gérer l'interopérabilité entre différentes plates-formes matérielles/logicielles. Indiquez ce qui dans CORBA assure cette interopérabilité.

Indication : comment un client en C++ avec un ORB Corba (par exemple le produit Orbacus de la société IONA) sur un Windows peut-il dialoguer avec un serveur en Java avec un ORB Borland Visibroker en Unix.

Le bus principalement, conjointement avec les stubs etc...

Exercice 2 – Langage IDL

1) Le langage IDL CORBA est un langage qui ne possède pas d'instructions de traitement. Pourquoi ?

Car c'est un langage de description d'interface, c'est-à-dire de description de la partie publique des objets.

2) Un compilateur C++ génère du code chargeable et exécutable. Qu'est ce que génère un compilateur IDL ?

Un compilateur IDL génère du code dans un langage cible (C++ si le client doit être écrit en C++ par exemple).

3) Le code IDL suivant définit quelques éléments pour une interface d'accès à des fonctions de comptabilité.

```
module CORBACompta {
    struct Compte {
        attribute string IdentCompte ;
        attribute string TypeCompte ;
        attribute Client client ;
    };
    struct Client {
        attribute string NomClient ;
    };
    typedef sequence<Compte> SeqCompte ;
    typedef sequence<Client> SeqClient ;

    interface GestionFamilledeCompte {
        Compte obtenirCompte (in string IdentCompte) ;
        SeqClient recherchepartypeCompte (in string TypeCompte);
        SeqCompte rechercheparnomclient (in string NomClient);
    };
};
```

Commentez cette définition. Quels sont les variables, les types et les opérations prévues à ce niveau d'implantation ?

Deux structures (Compte et Client), deux types listes (SeqCompte et SeqClient), une interface contenant trois méthodes, ayant toutes un seul paramètre, passé en mode entrée.

Exercice 3 – Architecture CORBA

1) Quel est de rôle de l'annuaire d'interfaces (*interface repository*) dans CORBA ?

Faire un dépôt d'interface, que le client pourra interroger pour ensuite communiquer avec le véritable objet existant à travers le bus.

2) Même question pour l'annuaire d'implémentations (*implementation repository*).

C'est un dépôt d'implémentation utilisé par le bus lui-même, permettant à CORBA d'instancier les objets correspondant à la demande d'un client.

3) CORBA réalise pour un utilisateur des fonctions habituelles dans un protocole réseau de multiplexage à l'émission et de démultiplexage à la réception. Comment est assurée cette fonction de multiplexage/démultiplexage par CORBA ?

*Gérer plusieurs langages (au niveau client et serveur).
Ce sont les stubs qui en ont le rôle.*

Exercice 4 – Corba et l'approche objet

1) CORBA se place dans le cadre des approches objets. L'une des caractéristiques majeure de l'approche objet est l'encapsulation (en anglais '*information hiding*'). En quoi CORBA assure-t-il le respect du principe d'encapsulation ?

On ne décrit, dans l'IDL, que les parties publiques. On peut ensuite utiliser du code et donc des objets non publics à partir de ces objets publics.

2) L'un des éléments considérés comme très important dans le développement de l'approche objet a été la possibilité de réutilisation de code par héritage. Comment l'héritage est-il traité dans CORBA ?

L'héritage est présent dans le langage IDL.