

UE SOR, Examen du 23 Mars 2015  
Partie CORBA/RMI/.NET  
Attention : document recto-verso  
Le barème est indicatif

Singhoff Frank  
singhoff@univ-brest.fr  
C-202

**Exercice 1 : implantation d'un service permettant d'accéder à une bibliothèque avec CORBA/Java (7 points)**

On souhaite implanter un service d'accès à une bibliothèque dont l'interface IDL est en annexe de ce sujet. Pour ce faire, on suppose que le serveur propose deux interfaces IDL : les interfaces IDL *ouvrage* et *bibliotheque*.

L'interface *ouvrage* ne comporte que des attributs qui stockent les différentes informations pour chaque ouvrage. Les données associées à chaque ouvrage sont le titre de l'ouvrage, le type d'ouvrage (ex : livre, DCD, ...) ainsi qu'un ensemble de mots clefs qui caractérisent le contenu de l'ouvrage.

Les méthodes accessibles par l'interface IDL *bibliotheque* sont les suivantes :

– La méthode *ajouter\_ouvrage*.

Cette méthode permet à un client d'ajouter un ouvrage dans la bibliothèque. Pour ce faire, le client doit fournir le titre de l'ouvrage. Les autres informations de l'ouvrage sont renseignées ultérieurement avec les attributs de l'interface *ouvrage*. Si un client tente d'ajouter un ouvrage dont le titre est déjà enregistré, l'objet CORBA doit lever l'exception *ouvrage\_deja\_existant*.

– La méthode *rechercher\_ouvrage*.

Cette méthode permet d'obtenir, sous la forme d'une séquence, toutes les références d'objet des ouvrages qui correspondent à une requête. La requête en question consiste à sélectionner parmi tous les ouvrages de la bibliothèque les ouvrages dont un des mots clefs correspond au mot passé en argument de la méthode *rechercher\_ouvrage*. Si aucun titre n'a été sélectionné, l'objet CORBA doit lever l'exception *pas\_d\_ouvrage*.

**Question 1 : 2 points**

A partir du fichier IDL en annexe, le compilateur IDL va générer différents fichiers Java. Citez les fichiers Java générés pour le type *ouvrage* et expliquez leur utilité.

**Question 2 : 5 points**

Proposez une classe d'implémentation en Java pour chaque interface IDL du module *bib*.

## Exercice 2 : Questions concernant .NET (6 points)

- Question 1 : En C#, expliquez pourquoi il est parfois nécessaire d'utiliser l'instruction *ref*.
- Question 2 : Expliquez ce qu'est un objet de type `MARSHALBYREFOBJECT`.
- Question 3 : Avec .NET, quelle est la durée de vie d'un objet réparti ?
- Question 4 : La technologie à objets répartis .NET est-elle transparente à la localisation ? Justifiez votre réponse.
- Question 5 : Dans l'interface d'un objet réparti .NET, expliquez pourquoi il est parfois nécessaire de compléter la déclaration d'une classe avec l'instruction `[SERIALIZABLE]`.
- Question 6 : Expliquez la signification de l'instruction `[WEBMETHOD(ENABLESESSION = TRUE)]`.

## Annexe

```
module bib {  
  
    enum famille {revue, livre, cd, dvd};  
    typedef sequence <string> liste_string;  
  
    interface ouvrage {  
        readonly attribute string titre;  
        attribute liste_string mots_clefs;  
        attribute famille famille_ouvrage;  
    };  
  
    typedef sequence <ouvrage> liste_d_ouvrage;  
    exception pas_d_ouvrage{};  
    exception ouvrage_deja_existant{};  
  
    interface bibliotheque {  
        ouvrage ajouter_ouvrage (in string titre)  
            raises (ouvrage_deja_existant);  
        void rechercher_ouvrage (inout liste_d_ouvrage l, in string mot)  
            raises (pas_d_ouvrage);  
    };  
};
```