

UE systèmes à objets et services répartis
Examen du 29 juin 2018
Le barème est indicatif
Session 2
Tous documents autorisés

Singhoff Frank
singhoff@univ-brest.fr
C-203

NB : ce document contient 2 feuilles recto/verso.

Exercice 1 : question de cours (3 points)

En CORBA, une référence d'objet peut être échangée entre clients et serveurs. Dans ce cas, la référence d'objet doit être sérialisée. Pourquoi ? Quelles sont les primitives CORBA permettant de sérialiser et désérialiser une référence d'objet CORBA ?

Exercice 2 : IDL vers Java (3 points)

L'annexe de ce document contient un fichier IDL. Chaque interface IDL de ce fichier va conduire à une génération de code. Donner la liste des fichiers qui seront générés et indiquer en quelques mots la fonction de chacun.

Exercice 3 : CORBA/Java (7 points)

On souhaite implanter les interfaces IDL en annexe. Ces interfaces offrent un moyen pour stocker les informations concernant les réservations d'un ensemble d'hôtel.

Chaque objet de type *hotel* mémorise les informations d'un hôtel. Au sein du serveur, l'interface *centrale* est instanciée une fois et permet d'enregistrer les données des hôtels. L'interface *centrale* permet d'instancier un objet CORBA pour chaque hôtel géré par l'application. La méthode *creation_hotel* ajoute un hôtel dans le serveur et retourne au client une référence sur l'objet CORBA représentant l'hôtel.

Par la suite, l'utilisateur peut ajouter différentes réservations pour l'hôtel associée à l'objet *hotel* grâce à la méthode *ajouter_reservation*. Chaque réservation est stockée dans la structure IDL *reservation*. Les données associées aux réservations pour un hôtel donné sont mémorisées au sein du serveur dans un tableau ou tout autre collection Java de votre choix.

Question 1

Proposer une mise en oeuvre pour les interfaces *hotel* et *centrale*.

Question 2

Soit le client suivant :

```
package hotels;

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

public class Client {
    public static void main(String[] args) throws IOException {

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

        String ior = "IOR:12A2D53EF4C39A9D23.....";
        ...
    }
}
```

On suppose que la variable *ior* référence un objet CORBA de type *centrale*. Complétez le client CORBA ci-dessus afin d'enregistrer pour l'hôtel dont le nom est "Hotel du centre" les deux réservations suivantes :

- Une chambre *single* pour 4 nuits.
- Une chambre *king* pour 2 nuits.

On suppose que le client doit également demander la création de l'objet *hotel* associé à "Hotel du centre".

Annexe

```
module hotels {

    enum type_chambre {single, king, appartement};

    struct reservation {
        type_chambre une_chambre;
        long nombre_nuit;
    };

    typedef sequence <reservation> reservations;

    interface hotel {
        readonly attribute string nom_hotel;
        readonly attribute reservations demandes;
        void ajouter_reservation(in reservation d);
    };

    interface centrale {
        void creation_hotel(in string nom_hotel, inout hotel h);
    };
};
```