

UE systèmes à objets et services répartis
Examen du 29 mars 2018
Le barème est indicatif
Session 1
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)

Vous travaillez dans une entreprise qui propose de réaliser pour ses clients des applications réparties avec 3 technologies différentes : les sockets TCP/IP programmées en C, Java RMI ou CORBA.

Donner et expliquer les critères qui vous conduiront à choisir les sockets, Java RMI ou CORBA pour la réalisation d'une application donnée.

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 des données météo.

Chaque objet de type *localisation* mémorise les informations d'une ville. Au sein du serveur, l'interface *base_donnees* est instanciée une fois et permet d'enregistrer les données météo. L'interface *base_donnees* permet d'associer à une ville plusieurs données météo. Chaque donnée météo est stockée dans la structure IDL *donnee*.

La méthode *creation_localisation* permet d'ajouter une ville dans le serveur et de retourner au client une référence sur un objet CORBA représentant cette ville.

Par la suite, l'utilisateur peut ajouter différentes données pour la ville associée à l'objet *localisation* grâce à la méthode *ajouter_donnee*. L'ensemble des données pour une ville peut être mémorisé au sein du serveur dans un tableau ou tout autre collection Java.

Enfin, la méthode *consulter_donnee* permet de récupérer toutes les données d'une ville via le paramètre *resultat* à partir du nom de la ville.

Question 1

Proposer une mise en oeuvre pour les interfaces *localisation* et *base_donnees*.

Question 2

Soit le client suivant :

```
package meteo;

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 *base_donnees*. Complétez le client CORBA ci-dessus afin d'enregistrer pour la ville de Brest les deux données définies ci-dessous :

- vitesse_vent = 20; orientation_vent=sud; heure = "12h30"
- vitesse_vent = 30; orientation_vent=nord; heure = "10h30"

On suppose que le client doit également demander la création de l'objet *localisation* associé à Brest.

Annexe

```
module meteo {  
  
    enum vent {sud, nord, est, ouest};  
  
    struct donnee {  
        long vitesse_vent;  
        vent orientation_vent;  
        string heure;  
    };  
  
    typedef sequence <donnee> ensemble_donnees;  
  
    interface localisation {  
        readonly attribute string nom_ville;  
        attribute ensemble_donnees les_donnees;  
        void ajouter_donnee(in donnee d);  
    };  
  
    interface base_donnees {  
        void creation_localisation(in string nom_ville, inout localisation l);  
        void consulter_donnee(in string nom_ville, inout ensemble_donnees resultat);  
    };  
};
```