

UE systèmes à objets répartis
Examen du 24 Mars 2014
Le barème est indicatif

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

Exercice sur CORBA/Java : Serveur de données 3D (7 points)

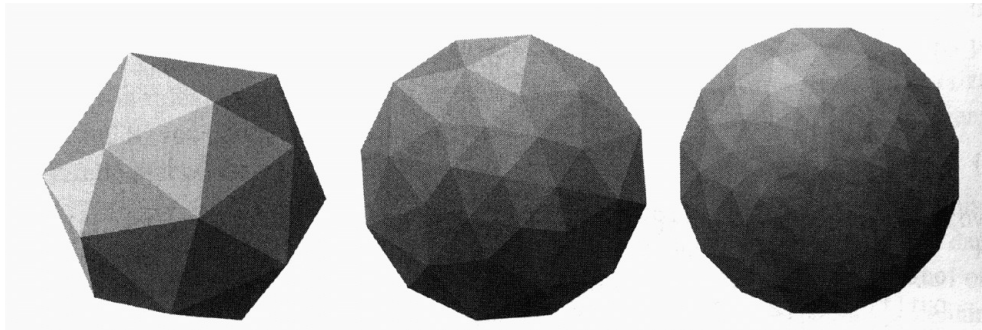


FIGURE 1 – Approximation d'un sphère par un polyèdre

On souhaite implanter un serveur de données qui mémorise des informations sur des objets en 3 dimensions. Ce serveur doit permettre à tout client d'accéder à des descriptions d'objets 3D ou polyèdre.

La Figure 1 montre plusieurs exemples d'objet 3D. Chaque objet (ou polyèdre) est constitué d'un ensemble de polygones. Chaque polygone est identifié par un nom unique. Les polygones sont des triangles. On rappelle qu'un triangle est défini par 3 points sur un plan. Chaque point d'un triangle est défini par trois coordonnées : x (l'abscisse) , y (l'ordonnée) et z (la profondeur).

On souhaite réaliser cette application en CORBA. L'interface du serveur mémorisant les polygones est la suivante :

```
module dessin_3d {  
  
    struct point {  
        long x;  
        long y;  
        long z;  
    };  
  
    interface polygone {  
        readonly attribute string nom_polygone;  
  
        attribute point point1;  
        attribute point point2;  
        attribute point point3;  
    };  
};
```

```

exception non_trouve{};

interface polyedre {

    readonly attribute long nombre_de_polygones;

    polygone ajouter_polygone(in string nom, in point p1, in point p2, in point p3);

    void rechercher_polygone(inout polygone p, in string nom)
        raises (non_trouve);
};
};

```

Question 1 : 5 points

Proposez une classe d'implémentation en Java pour chaque interface IDL du module *dessin_3d*; c-a-d les interfaces *polygone* et *polyedre*. Les méthodes :

- *ajouter_polygone* instancie un objet CORBA de type *polygone*, puis, insère cet objet dans une structure de données de votre choix et enfin renvoie au client la référence de l'objet CORBA ainsi créé.
- *rechercher_polygone* examine la structure de données qui mémorise les instances de type *polygone*. Elle recherche l'instance demandée par le client, puis, lui renvoie une référence sur l'objet ainsi trouvé. Pour retrouver le polygone, on utilise son nom : ainsi le second argument de *rechercher_polygone* est le nom du polygone recherché par le client.

Question 2 : 1 point

A partir de l'interface IDL ci-dessus, on va générer les entités suivantes : *polyedreHolder.java*, *polyedreHelper.java*, *polyedrePOA.java* et *_polyedreStub.java*.

Expliquez à quoi elles servent.

Question 3 : 1 point

Pourquoi un objet instancié à partir de la classe d'implémentation de *polygone* ne peut pas être passée en argument d'une méthode IDL ?