

Programación de Objetos Distribuidos

Trabajo Práctico 03 - RMI Definiciones

Ejercicio 1

Indicar cual es la función de los siguientes elementos en un ambiente RMI:

1. Interfaz remota
2. Servant
3. Stub
4. Client
5. Server
6. Servidor de Nombres

Ejercicio 2

Justificar si es verdadero o falso que la siguiente interfaz puede ser utilizada para un servicio remoto.

```
public interface Service extends Remote {  
    double getMediana(int[] elements);  
    int[] getModa(int[] elements);  
    double getPromedio(int[] elements);  
}
```

Ejercicio 3

Utilizando las clases implementadas en el ejercicio 05 de la clase "RMI definiciones" realice las siguientes operaciones y en cada caso describa y explique el resultado obtenido (cada ítem es independiente de los otros):

1. Levantar rmiregistry y servicio. Bajar el rmiregistry solamente e intentar conectarse con el cliente.
2. Levantar rmiregistry y servicio. Bajar el servicio solamente e intentar conectarse con el cliente. Describa y explique el resultado obtenido.
3. Levantar rmiregistry y servicio. Modificar el cliente para que en un ciclo infinito llame un método del servicio cada 1 segundo. Baje el proceso del servidor de nombres, describa y explique el resultado obtenido.
4. Repitiendo el caso anterior, baje el proceso servidor, describa y explique el resultado obtenido.
5. Copie las clases servant y server (no modificar más que el nombre de las clases) de manera de poder levantar ambos servicios juntos (el original y el copiado) con el mismo nombre. Conectar al cliente e indicar a cuál de los dos servidores se está conectando.
6. ¿Qué sucede si a los servers del punto anterior se cambia el comando rebind() por bind() y se ejecutan? ¿Proponga cuándo conviene usar rebind()?

Ejercicio 4

1. Programar un servidor RMI que contenga las siguientes operaciones:
 - **ping**: operación sin parámetros que contesta inmediatamente con la cadena: "pong"
 - **time**: operación sin parámetros que retorna la hora del servidor.
 - **echo**: operación que recibe una cadena e inmediatamente la retorna.
 - **hello**: operación que recibe una cadena (pensando que es un nombre) y retorna el texto "hello <nombre>".
 - **fortune**: operación sin parámetros que retorna una cadena de texto, seleccionando la misma aleatoriamente de una lista de textos que recibe durante la construcción el servant (los textos que son frases tipo fortune cookies) pueden estar "cableados" en el código de la clase servidor).
2. Realizar un cliente para poder probar las operaciones.