

Corso di Sistemi Distribuiti
Prof. Rocco Aversa
Prova scritta novembre 2015

Scrivere utilizzando Java RMI un'applicazione in cui viene implementato un servizio che effettui l'inversione delle colonne di una matrice di interi. Il codice del client dovrà effettuare le seguenti operazioni:

- Chiedere al server di attivare il servizio di inversione e di registrarlo temporaneamente al Registry con una stringa scelta dal client
- Assicuratosi che il servizio sia disponibile e registrato, il client utilizza il servizio;
- Chiedere al server di cancellare la registrazione del servizio.

Illustrare brevemente i passi che devono essere fatti per rendere operativa l'applicazione distribuita.

Memorandum delle principali classi e metodi necessari alla scrittura del codice:

The `java.rmi.Remote` interface serves to identify all remote interfaces; all remote objects must directly or indirectly implement this interface.

Implementation classes can implement any number of remote interfaces and can extend other remote implementation classes like `java.rmi.server.UnicastRemoteObject`

A `RemoteException` is the common superclass for a number of communication-related exceptions that may occur during the execution of a remote method call. Each method of a remote interface, an interface that extends `java.rmi.Remote`, must list `RemoteException` in its throws clause.

Della classe `Naming` :

```
public static void (re)bind(String name,  
                           Remote obj)  
    throws RemoteException,  
           MalformedURLException
```

Rebinds the specified name to a new remote object. Any existing binding for the name is replaced.

```
public static Remote lookup(String name)  
    throws NotBoundException,  
           MalformedURLException,  
           RemoteException
```

Returns a reference, a stub, for the remote object associated with the specified `name`.