

Corso di Sistemi Distribuiti
Prof. Rocco Aversa
Prova scritta febbraio 2017

Scrivere, utilizzando Java RMI, un **server** che offre il servizio **trasposta** di una matrice quadrata di interi. Il servizio trasposta ha come parametro di ingresso il nome del file dove è memorizzata la matrice ed il nome del servizio remoto che effettua la **lettura della matrice** (String **nome_file**).

Il servizio **lettura della matrice**, però viene attivato solo su richiesta attraverso un altro servizio remoto **registra_servizio**(String **nome_servizio**).

Scrivere il codice di un **client** che utilizza il servizio **trasposta**.

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`.