



CORSO PAS Laboratorio di RETI

Nicola Drago
nicola.drago@univr.it
Dipartimento di Informatica
Università di Verona



Definizione di un'interfaccia

E' necessario DEFINIRE l'interfaccia DEI metodi che sono resi disponibili

```
import java.rmi.Remote;  
import java.rmi.RemoteException;  
  
public interface ISquareRoot extends  
    Remote  
{  
    double    calculateSquareRoot(double  
aNumber) throws RemoteException;  
}
```



Definizione del server

Viene definito il codice lato server, ovvero chi esporta il servizio sarà una classe

```
import java.net.MalformedURLException;  
import java.rmi.server.UnicastRemoteObject;  
import java.rmi.Naming;  
import java.rmi.RemoteException;  
import java.io.*;  
  
public class RMISquareRootServer  
    extends UnicastRemoteObject  
    implements ISquareRoot  
{
```



Definizione del server

Un eventuale costruttore e il METODO REMOTO

```
public RMISquareRootServer() throws
FileNotFoundException, IOException, RemoteException
{
    //Costruttore code
}

public double calculateSquareRoot(double aNumber)
{
    return Math.sqrt( aNumber);
}
}
```



Istanziamento del SERVER - Servizio

Serve l'istanziamento dell'oggetto server (servirà un main)

```
import java.net.MalformedURLException;
import java.rmi.server.UnicastRemoteObject;
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.io.*;

public class RMITestSquareRoot
{
    public static void main(String[] args)
    {
        try
```



Istanziamento del SERVER - Servizio

L'oggetto va registrato nel "RMIREGISTRY" da questo momento sarà disponibile.

```
try
{
    ISquareRoot server = new RMISquareRootServer();
    Naming.rebind("//localhost/RMISquareRoot", server);
}
catch (RemoteException e) {e.printStackTrace();}
catch (MalformedURLException e) {e.printStackTrace();}
catch (FileNotFoundException e) {e.printStackTrace();}
catch (IOException e) {e.printStackTrace();}
}
}
```



Utilizzo dell'oggetto remoto il client

In un qualunque programma, su un qualunque PC può essere richiamato il metodo remoto

```
import java.rmi.Naming;  
import java.rmi.NotBoundException;  
import java.rmi.RemoteException;  
import java.net.MalformedURLException;  
  
public class RMISquareRootClient  
{  
    public static void main(String[] args) {  
        int x = Integer.parseInt(args[0]);  
    }  
}
```



Utilizzo dell'oggetto remoto il client

Viene richiesto in rete tramite RMIREGISTRY (Naming.lookup) l'oggetto remoto, dopodichè può essere richiamato il metodo.

```
try  
{  
    ISquareRoot squareServer = (ISquareRoot)  
        Naming.lookup ("rmi://localhost/RMISquareRoot");  
  
    double result = squareServer.calculateSquareRoot(x) ;  
    System.out.println(result);  
}  
catch (NotBoundException e) {e.printStackTrace(); }  
catch (RemoteException e) {e.printStackTrace(); }  
catch (MalformedURLException e) { e.printStackTrace(); }  
}
```



L'utilizzo

- *Come prima cosa è necessario eseguire (rimane residente fino al riavvio) il comando "RMIREGISTRY" di fatto il server dei nomi per gli oggetti RMI, a lui vanno registrati gli oggetti a lui, andranno richiesti gli oggetti.*
- *Va mandato in esecuzione RMITestSquareRoot che si preoccupa di registrare la classe su RMIREGISTRY*
- *Da questo momento in poi può essere istanziato e utilizzato l'oggetto remoto*



La Struttura


