Assignment No.1

Implement multi-threaded client/server Process communication using RMI.

# Problem Statement:

To develop any distributed application through implementing client-server communication programs based on Java RMI .

|  |  |  |
| --- | --- | --- |
| **Code:** |  |  |
| **1.** | **AddServerIntf.java** |
|  | import java.rmi.\*;  public interface AddServerIntf extends Remote {  double add(double d1, double d2) throws RemoteException;  } |
| **2.** | **AddServerImpl.java** |
|  | import java.rmi.\*; import java.rmi.server.\*;  public class AddServerImpl extends UnicastRemoteObject implements AddServerIntf {  public AddServerImpl() throws RemoteException {  }  public double add(double d1, double d2) throws RemoteException return d1 + d2;  }  } | { |
| **3.** | **AddServer.java** |  |
|  | import java.net.\*; import java.rmi.\*; public class AddServer {  public static void main(String args[]) { try {  AddServerImpl addServerImpl = new AddServerImpl(); Naming.rebind("AddServer", addServerImpl);  }  catch(Exception e) { System.out.println("Exception: " + e);  }  }  } |  |
| **4.** | **AddClient.java** |  |

import java.rmi.\*; public class AddClient {

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

String addServerURL = "rmi://" + args[0] + "/AddServer"; AddServerIntf addServerIntf = (AddServerIntf)Naming.lookup(addServerURL); System.out.println("The first number is: " + args[1]); double d1 = Double.valueOf(args[1]).doubleValue(); System.out.println("The second number is: " + args[2]); double d2 = Double.valueOf(args[2]).doubleValue();

System.out.println("The sum is: " + addServerIntf.add(d1, d2));

}

catch(Exception e) { System.out.println("Exception: " + e);

}

}

}

# OUTPUT:





