LAB:NO:06

OBJECT: TO GET FAMILIAR WITH CODE OF “RMI-IIOP”.

CLASS TASK: HelloWorld Application using IIOP.

Program: Interface.java:

import java.rmi.Remote;

public interface HelloInterfaceIIOP extends Remote{

public String sayHello(String from)throws java.rmi.RemoteException;

}

RemoteObjectClass.java:

import javax.rmi.PortableRemoteObject;

import java.rmi.RemoteException;

public class HelloImplIIOP extends PortableRemoteObject implements HelloInterfaceIIOP{

public HelloImplIIOP() throws RemoteException{

super();

}

public String sayHello(String from)throws RemoteException{

return "hello from"+from+"!!";

}

}

Server.java:

import javax.naming.InitialContext;

import javax.naming.Context;

public class HelloServerIIOP{

public static void main(String args[]){

try{

HelloImplIIOP helloRef=new HelloImplIIOP();

Context initialNamingContext=new InitialContext();

initialNamingContext.rebind("hello service",helloRef);

System.out.println("hello server is ready...");

}

catch(Exception e){

System.out.println(e);

}}}

Client.java:

import java.net.MalformedURLException;

import javax.rmi.\*;

import java.rmi.\*;

import java.util.Vector;

import javax.naming.\*;

import java.io.\*;

import java.util.Scanner;

public class sentClientIIOP {

public static void main( String args[]){

Context ic;

Object objref;

sentInterfaceIIOP hi;

String inputString;

Scanner sc = new Scanner(System.in);

System.out.println("Enter Input String :");

inputString = sc.nextLine();

try {

ic = new InitialContext();

objref = ic.lookup("ReverseSentence");

System.out.println("Client: Obtained a reference of server.");

hi = (sentInterfaceIIOP) PortableRemoteObject.narrow(

objref, sentInterfaceIIOP.class);

System.out.println("Server Reply: to client");

System.out.println(hi.reverseSent( inputString ));

}

catch( Exception e ) {

System.err.println( "Exception " + e + "Caught" );

e.printStackTrace( );

return;

}}}

Output:



TASK: Create RMI IIOP application in which take a sentence at runtime on the client application and let the server return its reverse.  
E.g. input could be: subject is distributed computing   
output should be: computing distributed is subject.

Program: Interface.java:

import java.rmi.Remote;

public interface sentInterfaceIIOP extends Remote {

public String reverseSent( String sent ) throws java.rmi.RemoteException;

}

RemoteObjectClass.java:

import javax.rmi.PortableRemoteObject;

import java.rmi.RemoteException;

public class sentImplIIOP extends PortableRemoteObject implements sentInterfaceIIOP{

public sentImplIIOP() throws RemoteException{

super();

}

public String reverseSent( String sent) throws RemoteException{

String[] words = sent.split("\\s");

String outputString="";

for (int i = words.length-1; i >= 0;i--){

outputString = outputString + words[i] +" ";

}

return outputString;

}

}

Server.java:

import javax.naming.InitialContext;

import javax.naming.Context;

public class sentServerIIOP {

public static void main(String args[]) {

try {

sentImplIIOP sentRef = new sentImplIIOP();

Context initialNamingContext = new InitialContext();

initialNamingContext.rebind("ReverseSentence", sentRef );

System.out.println("Hello!!! Server: Ready...");

} catch (Exception e) {

System.out.println("Trouble: " + e);

e.printStackTrace();

}}}

Client.java:

import java.net.MalformedURLException;

import javax.rmi.\*;

import java.rmi.\*;

import java.util.Vector;

import javax.naming.\*;

import java.io.\*;

public class HelloClientIIOP{

public static void main(String args[]){

Context ic;

Object objref;

HelloInterfaceIIOP hi;

try{

ic =new InitialContext();

objref=ic.lookup("hello service");

System.out.println("client obt a ref to hello server");

hi=(HelloInterfaceIIOP)PortableRemoteObject.narrow(

objref,HelloInterfaceIIOP.class);

System.out.println(hi.sayHello("mars"));

}

catch(Exception e){

System.out.println(e);

return;

}}}

Output:



