Practical 7(RMI palindrome)

**Palindrome.java -**

import java.rmi.\*;

interface Palindrome extends Remote

{

public int palin(String a) throws RemoteException;

}

**PalinClass.java -**

import java.rmi.\*;

import java.lang.\*;

import java.rmi.server.\*;

public class PalinClass extends UnicastRemoteObject implements Palindrome

{

public PalinClass () throws RemoteException { }

public int palin(String a) throws RemoteException

{

System.out.println("Hello");

StringBuffer str = new StringBuffer(a);

String str1 = str.toString();

System.out.println("Print : " + str1.toString());

StringBuffer str2 = str.reverse();

System.out.println("Print : " + str2.toString());

int b = str1.compareTo(str2.toString());

System.out.println("Print : " + b);

if (b == 0)

return 1;

else

return 0;

}

}

**rmiserver.java -**

import java.io.\*;

import java.rmi.\*;

import java.net.\*;

public class rmiserver

{

public static void main(String args[]) throws Exception

{

try

{

PalinClass twox = new PalinClass ();

Naming.bind("palin", twox);

System.out.println("Object registered");

}

catch(Exception e)

{

System.out.println("Exception" + e);

}

}

}

**rmiclient.java -**

import java.rmi.\*;

import java.util.Scanner;

public class rmiclient {

public static void main(String args[]) {

try {

Palindrome onex = (Palindrome) Naming.lookup("palin");

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string to check if it's a palindrome: ");

String inputString = scanner.nextLine();

int m = onex.palin(inputString);

System.out.println("Result: " + m);

if (m == 1) {

System.out.println("The given string is a Palindrome");

} else {

System.out.println("The given string is not a Palindrome");

}

scanner.close();

} catch (Exception e) {

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

}

}

}