import java.net.MalformedURLException; import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

public class ClientProgram {

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

LocateRegistry.createRegistry(8899);

Registry registry = LocateRegistry.getRegistry(8899);

PalinInterface p = (PalinInterface) registry.lookup("localhost/abc");

System.out.println

("Palindrome String is RACECAR");

boolean b = p.checkPalindrome("RACECAR"); if(b)

System.out.println("String is a Palindrome !!!");

else

System.out.println("String is not a Palindrome.");

} catch (Exception e) {

e.printStackTrace();

}

}

}

import java.net.MalformedURLException; import java.rmi.\*;

import java.rmi.registry.LocateRegistry; import java.rmi.registry.Registry;

public class ServerProgram {

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

PalinInterface p = new PalinImplement(); LocateRegistry.createRegistry(7788);

Registry registry = LocateRegistry.getRegistry(7788); registry.bind("localhost/abc", p);

} catch (Exception e) {

e.printStackTrace();

}

System.out.println("Server Started");

}

}

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

class PalinImplement extends UnicastRemoteObject implements PalinInterface {

protected PalinImplement() throws RemoteException { super();

}

public boolean checkPalindrome(String str) { String revstr = "";

int l = str.length(); int i;

for(i=l-1;i>=0;i--) {

revstr+=str.charAt(i);

}

if(str.compareTo(revstr)==0) return true;

else

return false;

}

}

interface

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface PalinInterface extends Remote {

public boolean checkPalindrome(String str) throws RemoteException;

}