Experiment No. 5

Aim of the Experiment: Develop an RMI application which accepts a string or a number and checks that string or number is palindrome or not.

Objective:

To Develop program using RMI application

Course Outcome Addressed: CO5

Theory:

Palindrome number:

A **palindrome number** is a number that is same after reverse. For example 545, 151, 34543, 343, 171, 48984 are the palindrome numbers. It can also be a string like LOL, MADAM etc.

Remote method invocation(RMI):

Remote method invocation(RMI) allow a java object to invoke method on an object running on another machine. RMI provide remote communication between java program. RMI is used for building distributed application.

Concept of RMI application:

A RMI application can be divided into two part,Client program and Server program. A Server program creates some remote object, make their references available for the client to invoke method on it. A Client program make request for remote objects on server and invoke method on them. Stub and Skeleton are two important object used for communication with remote object.

Stub:

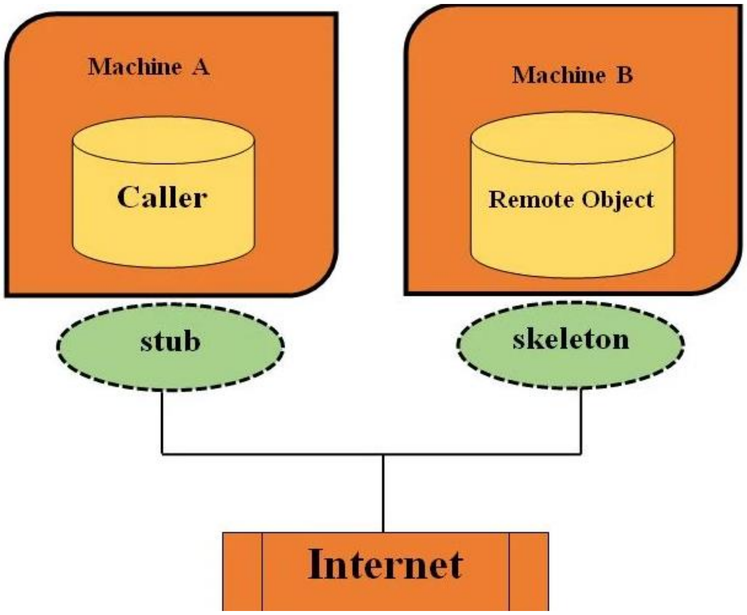
In RMI, a stub is an object that is used as a Gateway for the client-side. All the outgoing request are sent through it. When a client invokes the method on the stub object following things are performed internally:

1. A connection is established using Remote Virtual Machine.
2. It then transmits the parameters to the Remote Virtual Machine. This is also known as Marshals
3. After the 2nd step, it then waits for the output.
4. Now it reads the value or exception which is come as an output.
5. At last, it returns the value to the client.

Skeleton

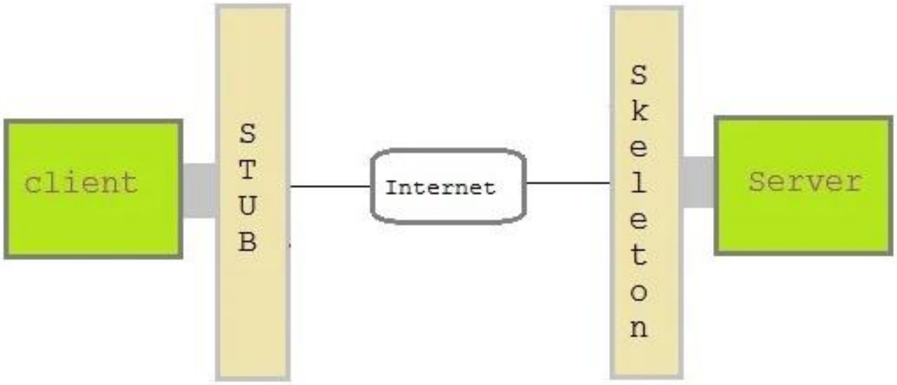
In RMI, a skeleton is an object that is used as a Gateway for the server-side.All the incoming request are sent through it. When a Server invokes the method on the skeleton object following things are performed internally:

1. All the Parameters are read for the remote method.
2. The method is invoked on the remote object.
3. It then writes and transmits the parameters for the result. This is also known as Marshals.



Stub and Skeleton

Stub act as a gateway for Client program. It resides on Client side and communicate with Skeleton object. It establish the connection between remote object and transmit request to it. Skeleton object resides on server program. It is responsible for passing request from Stub to remote object.



Creating a Simple RMI application involves following steps:-

- Define a remote interface.
- Implementing remote interface.
- create and start remote application
- create and start client application

Algorithm:

SERVER SIDE:

```
Step 1: Start
Step 2: Define the class rmiserver
Step 3: Create the object twox in try
Step 4: Register the object twox
Step 5: Display the exception in catch
Step 6: Stop
```

CLIENT SIDE:

```
Step 1: Start
Step 2: Define the class rmiclient
Step 3: Initialize the string s1 in try
Step 4: Create and Initialize the object onex
Step 5: Assign the value to m by calling the method palin
Step 6: Check whether the string is palindrome or not
Step 7: Display whether the string is palindrome or not
Step 8: Display the exception in catch
Step 9: Stop
```

SOURCE CODE:

one.java

```
import java.rmi.*;
interface one extends Remote
{

    public int palin(String a) throws RemoteException;
}
```

two.java

```
import java.rmi.*;
import java.lang.*;
import java.rmi.server.*;
public class two extends UnicastRemoteObject implements one
{
    public two() 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

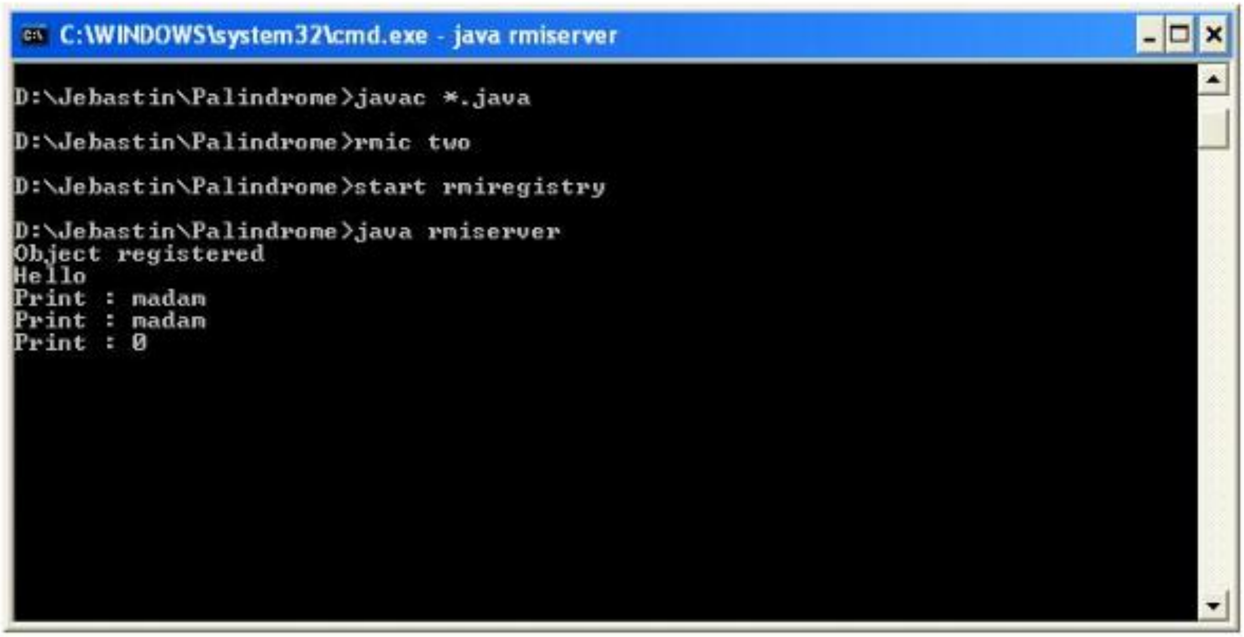
```
import java.io.*;
import java.rmi.*;
import java.net.*;
public class rmiserver
{
    public static void main(String args[]) throws Exception
    {
        try
        {
            two twox = new two();
            Naming.bind("palin", twox);
            System.out.println("Object registered");
        }
        catch(Exception e)
        {
            System.out.println("Exception" + e);
        }
    }
}
```

rmiclient.java

```
import java.io.*;
import java.rmi.*;
import java.net.*;
public class rmiclient
{
    public static void main(String args[]) throws Exception
    {
        try
        {
            String s1 = "rmi://localhost/palin";
            one onex = (one)Naming.lookup(s1);
            int m = onex.palin("madam");
            System.out.println("Print : " + m);
            if (m == 1)
            {
                System.out.println("The given string is a Palindrome");
            }
            else
            {
                System.out.println("The given string is not a Palindrome");
            }
        }
        catch (Exception e)
        {
            System.out.println("Exception" + e);
        }
    }
}
```

OUTPUT:

SERVER SIDE:



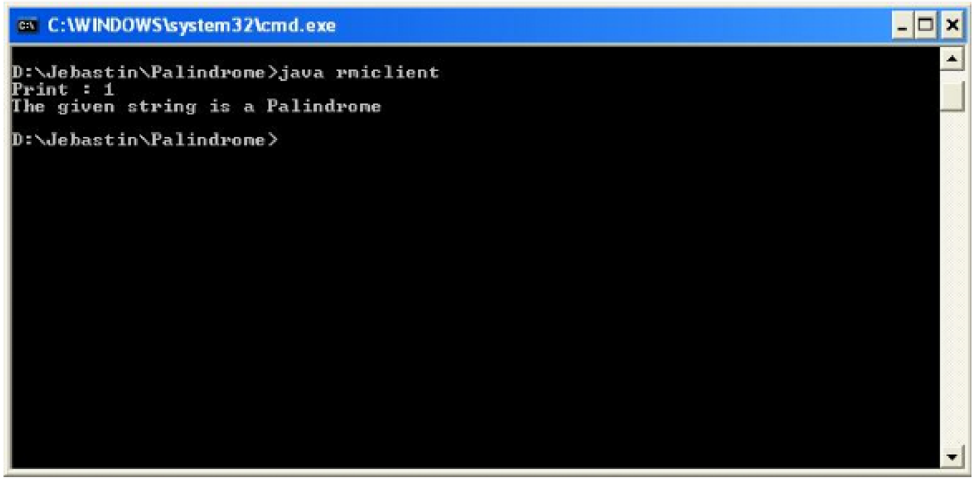
```
D:\Jebastin\Palindrome>javac *.java

D:\Jebastin\Palindrome>rmic two

D:\Jebastin\Palindrome>start rmiregistry

D:\Jebastin\Palindrome>java rmiserver
Object registered
Hello
Print : madam
Print : madam
Print : 0
```

CLIENT SIDE:



```
D:\Jebastin\Palindrome>java rmiclient
Print : 1
The given string is a Palindrome

D:\Jebastin\Palindrome>
```

Conclusion:

Thus the Java program for the implementation of Remote Method Invocation for palindrome was performed and the output was verified.

References:

Herbert Schildt , “Java : The Complete Reference” Tata McGraw-Hill (7th Edition).

Questions:

1. What is Java Remote Method Invocation (RMI)?
2. What is RMI remote object?
3. How does RMI communicate with the remote object?
4. Explain Different layers of RMI architecture.
5. What Are The Different Types Of Classes That Are Used In RMI?