**Practical 2**

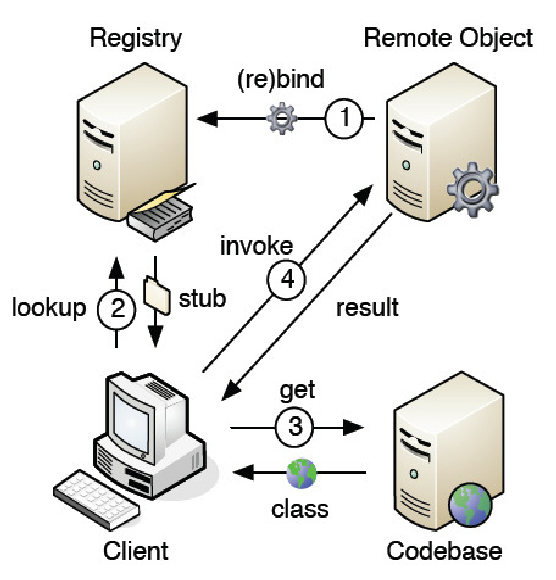
**Aim:**

2.2 Write RMI application where client will input two matrices from User and One Server will Multiply them and Second Server will add given matrices. Client will print results on its output screen. (Matrix can be initialized or taken from user).

**Description:**

RMI (Remote Method Invocation)

The **RMI** (Remote Method Invocation) is an API that provides a mechanism to create distributed application in java. The RMI allows an object to invoke methods on an object running in another JVM.



**Program code:**

**RemoteMultiple:-**

**import** java**.**rmi**.\*;**

public interface RemoteMultiple **extends** Remote

**{**

public int**[][]** multi**(**int**[][]** a**,** int**[][]** b**)** **throws** Exception**;**

**}**

**RemoteAdd:-**

**import** java**.**rmi**.\*;**

public interface RemoteAdd **extends** Remote

**{**

public int**[][]** add**(**int**[][]** a**,**int**[][]** b**)** **throws** Exception**;**

**}**

**ServerMultiply:-**

**import** java**.**rmi**.\*;**

**import** java**.**rmi**.**server**.\*;**

**import** java**.**lang**.**String**;**

public class ServerMultiply **extends**

UnicastRemoteObject **implements** RemoteMultiple**{**

public ServerMultiply**()throws** Exception**{**

**super();**

**}**

public int**[][]** multi**(**int**[][]** a**,** int**[][]** b**)** **{**

int rowsInA **=** a**.**length**;**

int columnsInA **=** a**[**0**].**length**;** // same as rows in B

int columnsInB **=** b**[**0**].**length**;**

int**[][]** c **=** **new** int**[**rowsInA**][**columnsInB**];**

**for** **(**int i **=** 0**;** i **<** rowsInA**;** i**++)** **{**

**for** **(**int j **=** 0**;** j **<** columnsInB**;** j**++)** **{**

**for** **(**int k **=** 0**;** k **<** columnsInA**;** k**++)** **{**

c**[**i**][**j**]** **=** c**[**i**][**j**]** **+** a**[**i**][**k**]** **\*** b**[**k**][**j**];**

**}**

**}**

**}**

**return** c**;**

**}**

**}**

**ServerMultiply2:-**

**import** java**.**rmi**.\*;**

**import** java**.**net**.\*;**

public class ServerMultiply2

**{**

public static void main**(**String args**[])**

**{**

**try**

**{**

ServerMultiply s**=new** ServerMultiply**();**

Naming**.**rebind**(**"SERVICE2"**,**s**);**

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

**}**

**catch(**Exception e**)**

**{**

System**.**out**.**println**(**e**.**getMessage**());**

**}**

**}**

**}**

**ServerAddition:-**

**import** java**.**rmi**.\*;**

**import** java**.**rmi**.**server**.\*;**

**import** java**.**lang**.**String**;**

public class ServerAddition **extends**

UnicastRemoteObject **implements** RemoteAdd**{**

public ServerAddition**()throws** Exception**{**

**super();**

**}**

public int**[][]** add**(**int**[][]** a**,**int**[][]** b**){**

int rows**=** a**.**length**;**

int columns **=** a**[**0**].**length**;** // same as rows in B

int**[][]** c **=** **new** int**[**rows**][**columns**];**

**for** **(**int i **=** 0**;** i **<** rows**;** i**++)** **{**

**for** **(**int j **=** 0**;** j **<** columns**;** j**++)** **{**

c**[**i**][**j**]** **=** a**[**i**][**j**]** **+** b**[**i**][**j**];**

**}**

**}**

**return** c**;**

**}**

**}**

**ServerAdd2:-**

**import** java**.**rmi**.\*;**

**import** java**.**net**.\*;**

public class ServerAdd2

**{**

public static void main**(**String args**[])**

**{**

**try**

**{**

ServerAddition s**=new** ServerAddition**();**

Naming**.**rebind**(**"SERVICE1"**,**s**);**

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

**}**

**catch(**Exception e**)**

**{**

System**.**out**.**println**(**e**.**getMessage**());**

**}**

**}**

**}**

**Client2:-**

**import** java**.**rmi**.\*;**

**import** java**.**io**.\*;**

**import** java**.**util**.\*;**

public class Client2**{**

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

**try{**

Scanner s**=new** Scanner**(**System**.**in**);**

System**.**out**.**println**(**"Enrollment No:130050131070"**);**

System**.**out**.**print**(**"Enter number of rows in A: "**);**

int rowsInA **=** s**.**nextInt**();**

System**.**out**.**print**(**"Enter number of columns in A / rows in B: "**);**

int columnsInA **=** s**.**nextInt**();**

System**.**out**.**print**(**"Enter number of columns in B: "**);**

int columnsInB **=** s**.**nextInt**();**

int**[][]** a **=** **new** int**[**rowsInA**][**columnsInA**];**

int**[][]** b **=** **new** int**[**columnsInA**][**columnsInB**];**

int**[][]** c**;**

System**.**out**.**println**(**"Enter matrix A"**);**

**for** **(**int i **=** 0**;** i **<** a**.**length**;** i**++)** **{**

**for** **(**int j **=** 0**;** j **<** a**[**0**].**length**;** j**++)** **{**

a**[**i**][**j**]** **=** s**.**nextInt**();**

**}**

**}**

System**.**out**.**println**(**"Enter matrix B"**);**

**for** **(**int i **=** 0**;** i **<** b**.**length**;** i**++)** **{**

**for** **(**int j **=** 0**;** j **<** b**[**0**].**length**;** j**++)** **{**

b**[**i**][**j**]** **=** s**.**nextInt**();**

**}**

**}**

System**.**out**.**println**(**"enter your opion:\n 1.add \n 2.Multi"**);**

int z**;**

z**=**s**.**nextInt**();**

String ip**=**"rmi://localhost/SERVICE"**+**z**;**

**if(**z**==**2**){**

RemoteMultiple s1**=(**RemoteMultiple**)**Naming**.**lookup**(**ip**);**

c**=** s1**.**multi**(**a**,**b**);**

**}**

**else{**

RemoteAdd s2**=(**RemoteAdd**)**Naming**.**lookup**(**ip**);**

c**=**s2**.**add**(**a**,**b**);**

**}**

**for** **(**int i **=** 0**;** i **<** c**.**length**;** i**++)** **{**

**for** **(**int j **=** 0**;** j **<** c**[**0**].**length**;** j**++)** **{**

System**.**out**.**print**(**c**[**i**][**j**]+”\t”);**

**}**

System**.**out**.**println**();**

**}**

**}**

**catch(**Exception e**){**

System**.**out**.**println**(**e**.**getMessage**());**

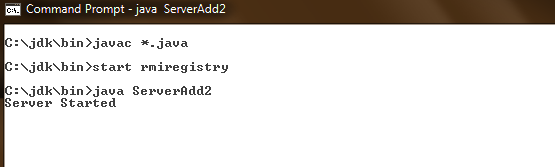
e**.**printStackTrace**();**

**}**

**}**

**}**

**Input Output:**

****

