GOA COLLEGE OF ENGINEERING

"Bhausaheb Bandodkar Technical Education Complex"

Experiment No: 3

Remote Method Invocation
Aim: To implement Remote Method Invocation.

Theory:

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.

The RMI provides remote communication between the applications using two objects stub and skeleton. RMI uses stub and skeleton object for communication with the remote object. A **remote object** is an object whose method can be invoked from another JVM. Let's understand the stub and skeleton objects:

Stub

The stub is an object, acts as a gateway for the client side. All the outgoing requests are routed through it. It resides at the client side and represents the remote object. When the caller invokes method on the stub object, it does the following tasks:

1. It initiates a connection with remote Virtual Machine (JVM),
2. It writes and transmits (marshals) the parameters to the remote Virtual Machine (JVM),
3. It waits for the result
4. It reads the return value or exception, and
5. It finally, returns the value to the caller.

Skeleton

The skeleton is an object, acts as a gateway for the server-side object. All the incoming requests are routed through it. When the skeleton receives the incoming request, it does the following tasks:

1. It reads the parameter for the remote method
2. It invokes the method on the actual remote object, and
3. It writes and transmits (marshals) the result to the caller.

Program:
//server code

```
import java.rmi.*;
import java.rmi.registry.*;


public class MyServer{


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

```
Adder stub=new AdderRemote();
Naming.rebind("rmi://localhost:5000/white",stub);
System.out.println("Server Started");
}catch(Exception e){System.out.println(e);}
}}

//client code
import java.rmi.*;

public class MyClient{

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

Adder stub=(Adder)Naming.lookup("rmi://localhost:5000/white");
System.out.println(stub.add(9,9));

}catch(Exception e){System.out.println(e);}
}}

//adder remote code
import java.rmi.*;
import java.rmi.server.*;

public class AdderRemote extends UnicastRemoteObject implements Adder{

AdderRemote()throws RemoteException{
super();
}

public int add(int x,int y){return x+y;}
```

}

Output:

```
C:\Users\deeprajb\Dropbox\dos tutorials\expt3>javac *.java
C:\Users\deeprajb\Dropbox\dos tutorials\expt3>rmic AdderRemote
C:\Users\deeprajb Dropbox\dos tutorials\expt3>rmiregistry 5000
C:\Users\deeprajb\Dropbox\dos tutorials\expt3>java MyServer
C:\Users\deeprajb\Dropbox\dos tutorials\expt3>java MyClient
18
```

Conclusion: Remote Method Invocation experiment was implemented and executed successfully.