**JAC444 - Lecture 11**

Remote Method Invocation

Segment 2 - Develop RMI Application

# Remote Method Invocation

**In this lesson you will be learning about:**

* Designing RMI application
* Developing distributed object defined by RMI interfaces
* Designing and developing RMI Server
* Designing and developing a RMI Client
* Deploying and running the RMI system

# Building Calculator RMI System

Design and implement Java RMI Calculator interfaces

**2**

**1**

Develop Java code implementing classes defined by RMI Calculator interfaces

Develop code for Java RMI Calculator server

Develop code for Java RMI Calculator client program

**5** Install and run RMI Calculator system

**3**

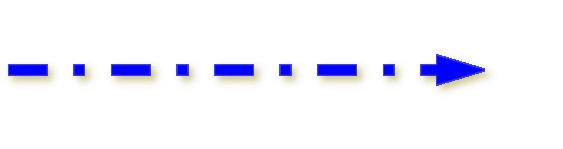
**4**

# Naming Remote Objects

How could a client find an RMI remote server (service)?

* RMI System includes a simple service called the *RMI Registry*: **rmiregistry**
* On a server machine, a server program creates a remote service and register it in the RMI registry.
* On the client side, RMI Registry is accessed through the class Naming***.*** The static method lookup(String url)is the method a client uses it to query a registry.
* The method returns a remote reference to the service object. The URL parameter of a lookup method takes the form:

**rmi://<host\_name>[:<service\_port>]/<service\_name>**

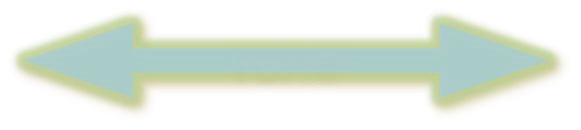
**Overview of RMI**



RMI Client



Stub



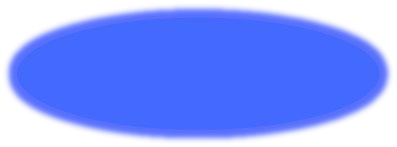
TCP/IP

Method call

Parameters

Return value

Look up server



RMI

Registry

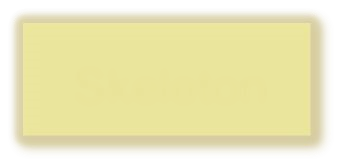


RMI Client Machine

RMI Server Machine



RMI Server



Skeleton

# Calculator Interfaces

**1**

Interface defines all of the remote features offered by the server – **Calculator.java**

**public interface Calculator extends java.rmi.Remote {**

**public long add(long a, long b)**

**throws java.rmi.RemoteException;**

**public long sub(long a, long b)**

**throws java.rmi.RemoteException;**

**public long mul(long a, long b)**

**throws java.rmi.RemoteException;**

**public long div(long a, long b)**

**throws java.rmi.RemoteException;**

**}**

# Calculator Implementation Class

The implementation of the interface for the remote service.

**CalculatorImpl.java**

**public class CalculatorImpl extends java.rmi.server.UnicastRemoteObject**

**implements Calculator {**

**2**

**/**/Implementations must have an explicit constructor //in order to declare the RemoteException exception  **public CalculatorImpl() throws java.rmi.RemoteException { super(); }**

**public long add(long a, long b)throws java.rmi.RemoteException { return a + b; } ...**

**}**

# Calculator RMI Server

The class CalculatorServer.java is a server class that has only constructor.

# Calculator RMI Client

RMI Client: **CalculatorClient.java**

import java.rmi.Naming; import java.rmi.RemoteException; import java.net.MalformedURLException; import java.rmi.NotBoundException; public class CalculatorClient {

public static void main(String[] args) {

try {

|  |  |
| --- | --- |
| **Calculator c =** |  |
| **(Calculator)Naming.lookup("rmi://localhost/CalculatorService");** | |

System.out.println( **c.sub(4, 3)** );

} catch (MalformedURLException murle) {

System.out.println(murle);

} catch (RemoteException re) {

System.out.println(re);

} catch (NotBoundException nbe) {

System.out.println(nbe);

}

}

}

# Running Calculator RMI System

* Run the *Registry*. You must be in the directory that contains the classes you have developed. From there, enter the following rmiregistry

**5**

* Start the RMI calculator server hosting the Calculator service java CalculatorServer
* Run the RMI calculator client program

java CalculatorClient

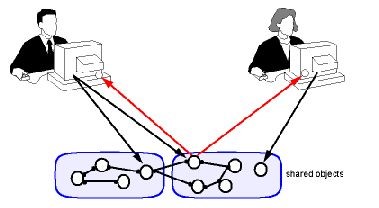
# RMI over IIOP

Common Object Request Broker Architecture CORBA

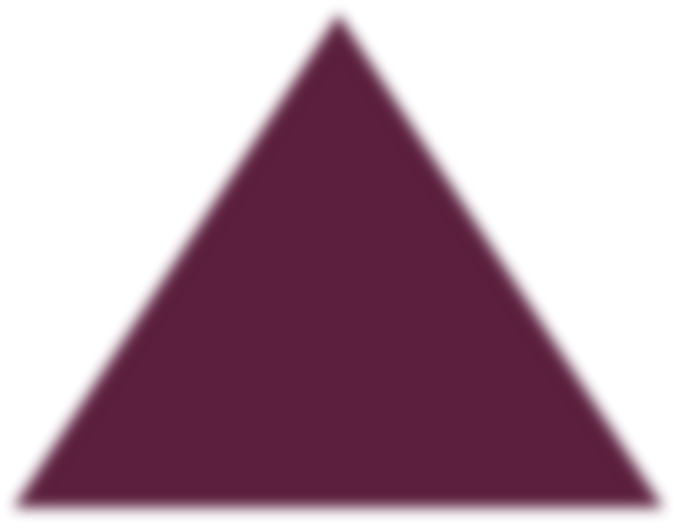
Remote Method Invocation (RMI)

over Internet Inter-Orb Protocol (IIOP)

## access distributed objects on the Internet



# Run RMI System



**Method**

RMI

**Client**

**Registry**

RMI

**Bind**

**1**

**2**

**Lookup**

**3**

**Invocation**

RMI

**Server**

# Conclusion

**After completion of this lesson you should know:**

* How to design distributed applications using RMI.
* How to develop Java RMI programs.
* How to deploy applications using RMI tools.



**Jordan Anastasiade – Java Programming Language Course 13**