Camlin Page
Date / /

Assignment No 3

Q. Create program based on RMI using swing components

* Remote Method Invocation (RMI)

It is a mechanism that allows an object residing in one system to access an object running on another JVM.

RMI is used to build distributed applications, it provides remote communication between java program. It is provided in the package "Java.rmi"

Architecture - In RMI application, we write two program, a server program and a client program the server program is created and referenced of that object is made available for the client. The client program requests the remote object on the server and tries to invoke it's methods.

Working - When the client makes call to the remote object ~~obj~~ it is recieved by the stub which eventually pases his request to the RRL. The client - ~~server~~side ~~request~~ receives request which invokes a method called invoke() of the object remoteRof. It passes the request to RRL on the server-side.

The RRL on the server side passes the request to the skeleton which finally invokes the required object on server. The resuet is passed all the way back to the client.

Scanned with CamScanner

Client Program :

```
import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton; import

javax.swing.JFrame; import

javax.swing.JLabel; import

javax.swing.JTextField; import

java.rmi.NotBoundException; import

java.rmi.RemoteException; import

java.rmi.registry.LocateRegistry; import

java.rmi.registry.Registry; import

java.util.Scanner; import

java.awt.Color;


class Swing extends JFrame implements ActionListener {


   JButton jb1;

   JTextField jt1;

   JLabel lbl, lb3;


   Swing() {


            lb3 = new JLabel("Enter number");

      lb3.setForeground(Color.WHITE);

lb3.setBounds(160, 50, 150, 60);        add(lb3);


      jt1 = new JTextField();

jt1.setBounds(160, 100, 150, 30);

add(jt1);
```

```
        lbl   =   new   JLabel("Result   :");
lbl.setForeground(Color.WHITE);
lbl.setBounds(160, 140, 150, 30);
        add(lbl);



        jb1 = new JButton("Check");
jb1.setBounds(160, 200, 100, 30);         add(jb1);

        jb1.addActionListener(this);

        setLayout(null);         setSize(600, 400);
setVisible(true);         setTitle("Prime Number");
getContentPane().setBackground(Color.BLACK);
    }

    public void actionPerformed(ActionEvent e)
{
               try
               {
               Registry reg = LocateRegistry.getRegistry("localhost",3333);
        prime pd = (prime)reg.lookup("Hii Server");
int a = Integer.parseInt(jt1.getText());

        String c;

        if (e.getSource().equals(jb1)) {             c =
pd.prime(a);             lbl.setText("Result : "+
String.valueOf(c));
lbl.setForeground(Color.WHITE);
        }
              }catch(RemoteException p)
              {
                            System.out.println("Exception"+e);
              }catch(NotBoundException q)
              {
                            System.out.println("Exception"+e);
              }
    }

    public static void main(String args[]) throws RemoteException ,NotBoundException
    {
        Swing t = new Swing();


    }
}
```

Server Program :

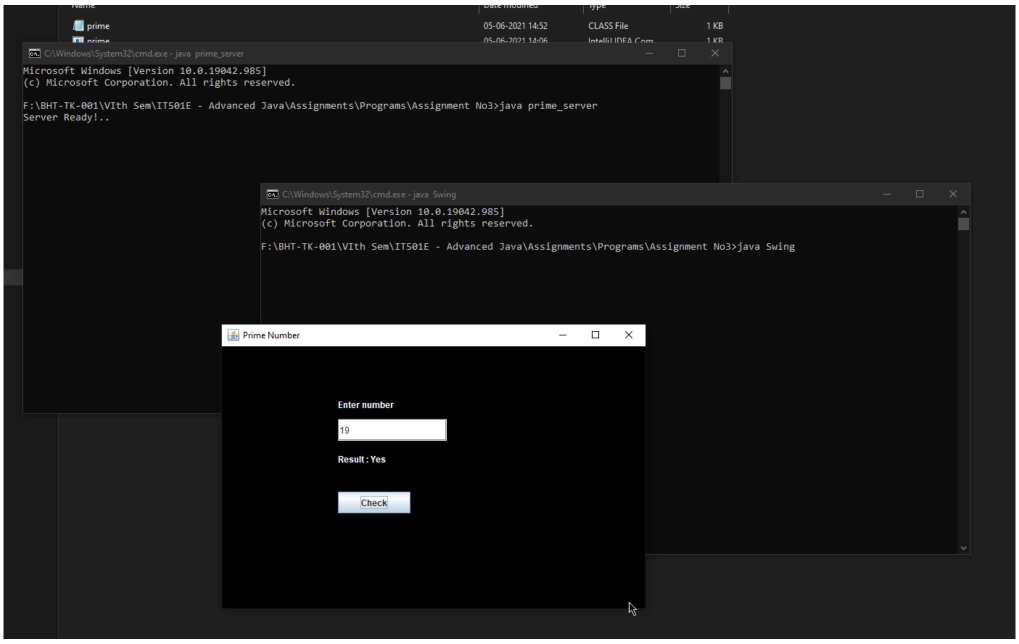
```
import java.rmi.RemoteException; import
java.rmi.registry.LocateRegistry; import
java.rmi.registry.Registry; import
java.rmi.server.UnicastRemoteObject;
```

```
public class prime_server extends UnicastRemoteObject implements prime{
public prime_server() throws  RemoteException
   {
      super();
   }

    //@Ovverride

    public String prime(int n) throws RemoteException {
                        int num = n;
String p = "Noo";
boolean flag = false;
                        for (int i = 2; i <= num / 2; ++i) {
                                    // condition for nonprime number
                        if (num % i == 0) {
                                                flag = true;
                                                break;
                                    }
                        }
                        if (!flag)
                                    p="Yes";
            else
p="No";
                        return p;
                        }

   public static void main(String args[]) throws RemoteException
   {
try
{
       Registry reg = LocateRegistry.createRegistry(3333);
reg.rebind("Hii Server", new prime_server());
       System.out.println("Server Ready!..");
    }
    catch(RemoteException e)
    {
        System.out.println("Exception" +e);
    }
   }
}
```

Register Program :

```
import java.rmi.*; public interface
prime extends Remote
{
    public String prime(int n) throws RemoteException ;
}
```

Outptuts:


C:\Windows\System32\cmd.exe - java prime_server

```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

F:\BHT-TK-001\VIth Sem\IT501E - Advanced Java\Assignments\Programs\Assignment No3>java prime_server
Server Ready!..
```

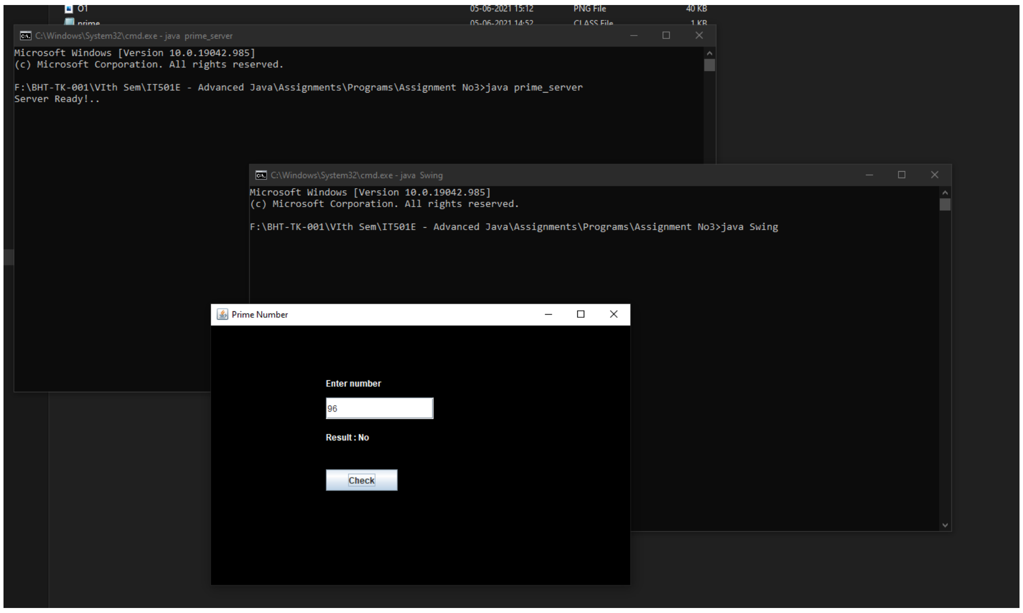
C:\Windows\System32\cmd.exe - java Swing

```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

F:\BHT-TK-001\VIth Sem\IT501E - Advanced Java\Assignments\Programs\Assignment No3>java Swing
```

Enter number

19

Result : Yes


C:\Windows\System32\cmd.exe - java prime_server

```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

F:\BHT-TK-001\VIth Sem\IT501E - Advanced Java\Assignments\Programs\Assignment No3>java prime_server
Server Ready!..
```

C:\Windows\System32\cmd.exe - java Swing

```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

F:\BHT-TK-001\VIth Sem\IT501E - Advanced Java\Assignments\Programs\Assignment No3>java Swing
```

Enter number

96

Result : No