

Lab# 4

Task : Develop the RMI Callback application of Payroll and PayrollMessage services for calculating the earnings.

1. payrollInterface.java

```
PayrollInterface.java PayrollMessageInterface.java Payroll.java PayrollServer.java PayrollMessage.java PayrollMessageServer.java
1 import java.rmi.*;
2 public interface PayrollInterface extends Remote{
3     public double earning(int id,double hours) throws RemoteException;
4 }
5
```

2. payrollMessageInterface.java

```
PayrollInterface.java PayrollMessageInterface.java Payroll.java PayrollServer.java PayrollMessage.java PayrollMessageServer.java
1 import java.rmi.*;
2 public interface PayrollMessageInterface extends Remote{
3     public void message(String message) throws RemoteException;
4 }
5
```

3. payroll.java

```
PayrollInterface.java PayrollMessageInterface.java Payroll.java PayrollServer.java PayrollMessage.java PayrollMessageServer.java
1 import java.rmi.*;
2 import java.rmi.server.*;
3 import java.net.*;
4 public class Payroll extends UnicastRemoteObject implements PayrollInterface{
5     private double pay;
6     public Payroll() throws RemoteException{}
7     public double earning(int id,double hours) throws RemoteException{
8         if(id==1)
9             pay=10*hours;
10        else if(id>1)
11            pay=100*hours;
12        String clientService="rmi://localhost/PayrollMessage";
13    try{
14        PayrollMessageInterface pm=(PayrollMessageInterface)Naming.lookup(clientService);
15        pm.message("I am ready to send payroll amount");
16    }catch(Exception e){e.printStackTrace();}
17    return pay;
18    }
19 }
```

4. payrollServer.java

```
PayrollInterface.java PayrollMessageInterface.java Payroll.java PayrollServer.java PayrollMessage.java PayrollMessageServer.java
1 import java.rmi.*;
2 class PayrollServer{
3 public static void main(String args[]){
4 try{
5 Naming.rebind("rmi://localhost/PayrollServer",new Payroll());
6 System.out.println("Payroll Server is ready.");
7 }catch(Exception e){
8 System.out.println("Payroll Server failed: "+e);
9 }}}
10
```

5. payrollMessage.java

```
PayrollInterface.java PayrollMessageInterface.java Payroll.java PayrollServer.java PayrollMessage.java PayrollMessageServer.java
1 import java.rmi.*;
2 import java.rmi.server.*;
3 import java.net.*;
4 public class PayrollMessage extends UnicastRemoteObject implements PayrollMessageInterface{
5 public PayrollMessage() throws RemoteException{ }
6 public void message(String message) throws RemoteException{
7 System.out.print("Message from server "+message);
8 }
9 }
10
```

6. payrollMessageServer.java

```
PayrollInterface.java PayrollMessageInterface.java Payroll.java PayrollServer.java PayrollMessage.java PayrollMessageServer.java
1 import java.rmi.*;
2 class PayrollMessageServer{
3 public static void main(String args[]){
4 String name="rmi://localhost/PayrollMessage";
5 try{
6 Naming.rebind(name,new PayrollMessage());
7 String serverName="rmi://localhost/PayrollServer";
8 PayrollInterface pr=(PayrollInterface)Naming.lookup(serverName);
9 System.out.print("\n"+pr.earning(1,12));
10 }catch(Exception e){e.printStackTrace();}
11 }}}
12
```

Output:

```
C:\Windows\System32\cmd.exe
C:\Users\nemon\Desktop\semester 8\dc labs\lab5>javac *.java
C:\Users\nemon\Desktop\semester 8\dc labs\lab5>start rmiregistry
C:\Users\nemon\Desktop\semester 8\dc labs\lab5>start java PayrollServer
C:\Users\nemon\Desktop\semester 8\dc labs\lab5>
```

```
C:\ProgramData\Oracle\Java\javapath\java.exe
Payroll Server is ready.
```

```
C:\Windows\System32\cmd.exe
C:\Users\memon\Desktop\semester 8\dc labs\lab5>javac *.java
C:\Users\memon\Desktop\semester 8\dc labs\lab5>start rmiregistry
C:\Users\memon\Desktop\semester 8\dc labs\lab5>start java PayrokkServer
C:\Users\memon\Desktop\semester 8\dc labs\lab5>start java PayrollServer
C:\Users\memon\Desktop\semester 8\dc labs\lab5>start java PayrollMessageServer
```

```
C:\ProgramData\Oracle\Java\javapath\java.exe
Message from server I am ready to send payroll amount
120.0
```

S