



EXERCISE 3A

PART A Java RMI

Alberto Hernández Lado and Álvaro López Pereda

SERVER AND CLIENT FILES

CalcObject.java interface

```
CalcObject.java x
1  import ...
3
4  public interface CalcObject extends Remote {
5      double calculate(double a, double b) throws RemoteException;
6
7  }
```

CalcObject2.java interface

```
CalcObject2.java x
1  import ...
3
4  public interface CalcObject2 extends Remote {
5      ResultType calculate(InputType inputParam) throws RemoteException;
6  }
```

InputType.java class

```
InputType.java x
1  import java.io.Serializable;
2
3  public class InputType implements Serializable {
4      private static final long serialVersionUID = 101L;
5
6      String operation;
7      public double x1;
8      public double x2;
9
10     public double getX1() { return x1; }
13
14     public double getX2() { return x2; }
17 }
```

MyData.java class

```
MyData.java x
1  import ...
5
6  public class MyData {
7
8      public static void info() {
9          DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
10         System.out.println("yyyy/MM/dd HH:mm:ss-> "+dtf.format(LocalDateTime.now()));
11
12         System.out.println("Data:" + "Alvaro Lopez Pereda and Alberto Hernandez Lado");
13
14         System.out.println("User Name:" + System.getProperty("user.name"));
15
16         System.out.println("Operating System Name:" + System.getProperty("os.name"));
17
18         System.out.println("Version:" + System.getProperty("java.runtime.version"));
19
20         InetAddress ip;
21         try {
22             ip = InetAddress.getLocalHost();
23             System.out.println("Ip Address:" + ip);
24         } catch (UnknownHostException e) {
25             // TODO Auto-generated catch block
26             e.printStackTrace();
27         }
28     }
29 }
```

ResultType.java class

```
ResultType.java x
1  import java.io.Serializable;
2
3  public class ResultType implements Serializable {
4      private static final long serialVersionUID = 102L;
5
6      String result_description;
7      public double result;
8  }
```

srv.policy file

```
srv.policy x
1  grant{
2      permission java.security.AllPermission;
3  };
```

SERVER SIDE

CalcObjImpl.java class

```
1  import ...
3
4  public class CalcObjImpl extends UnicastRemoteObject implements CalcObject {
5      private static final long serialVersionUID = 101L;
6
7      public CalcObjImpl() throws RemoteException{
8          super();
9      }
10
11     @Override
12     public double calculate(double a, double b) throws RemoteException {
13         double result = a + b;
14         return result;
15     }
16 }
```

CalcObjImpl2.java class

```
1  import ...
3
4  public class CalcObjImpl2 extends UnicastRemoteObject implements CalcObject2 {
5      public CalcObjImpl2() throws RemoteException {
6          super();
7      }
8
9      @Override
10     public ResultType calculate(InputType inputParam) throws RemoteException {
11         double zm1, zm2;
12         ResultType res = new ResultType();
13
14         zm1 = inputParam.getX1();
15         zm2 = inputParam.getX2();
16
17         res.result_description = "Operation " + inputParam.operation;
18
19         switch (inputParam.operation) {
20             case "add":
21                 res.result = zm1 + zm2;
22                 break;
23             case "sub":
24                 res.result = zm1 - zm2;
25                 break;
26             default:
27                 res.result = 0;
28                 res.result_description = "An invalid operation was specified!";
29                 return res;
30         }
31
32         return res;
33     }
34 }
```

MyServer.java class

```
MyServer.java x
1  import java.net.MalformedURLException;
2      import java.rmi.Naming;
3      import java.rmi.RemoteException;
4      import java.rmi.registry.LocateRegistry;
5      import java.rmi.registry.Registry;
6
7  public class MyServer {
8  @ public static void main(String[] args){
9      if (args.length == 0){
10         System.out.println("You have to enter RMI object address in the form: //host_address/service_name");
11         return;
12     }
13     try {
14         Registry reg = LocateRegistry.createRegistry( port 1099);
15     } catch (RemoteException e) {
16         e.printStackTrace();
17     }
18     System.setProperty("java.rmi.server.hostname", "192.168.56.1");
19     System.setProperty("java.security.policy", "D:\\2022_DCS\\zf\\cw3a_tel\\serverRMI\\src\\srv.policy");
20     if(System.getSecurityManager() == null)
21         System.setSecurityManager(new SecurityManager());
22     try{
23         MyData.info();
24         CalcObjImpl mySrvObj = new CalcObjImpl();
25         System.out.println(args[0]);
26         java.rmi.Naming.rebind(args[0], mySrvObj);
27         System.out.println("Server is registered now :-)");
28         //System.out.println("Press Ctrl+C to stop...");
29     } catch (Exception e) {
```

```
31         System.out.println("SERVER CAN'T BE REGISTERED!");
32         e.printStackTrace();
33         return;
34     }
35     try {
36         CalcObjImpl2 mySrvObj2 = new CalcObjImpl2();
37         System.out.println(args[1]);
38         Naming.rebind(args[1], mySrvObj2);
39         java.rmi.Naming.rebind(args[1], mySrvObj2);
40         System.out.println("Server2 is registered now!");
41         //System.out.println("Press Ctrl + C to stop...");
42     } catch (Exception e) {
43         System.out.println("Server can't be registered");
44         e.printStackTrace();
45         return;
46     }
47 }
48 }
```

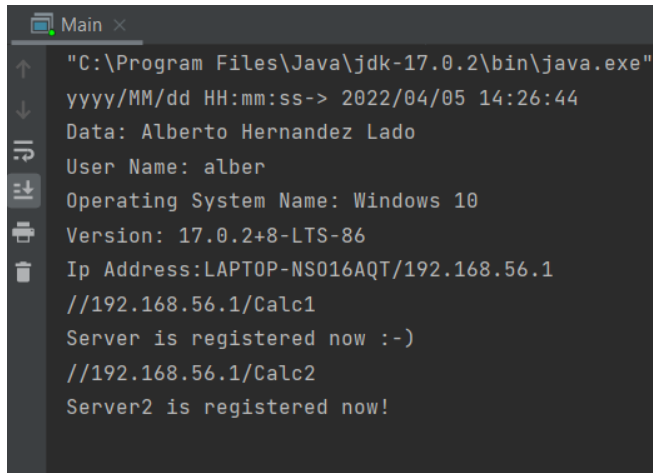
CLIENT SIDE

MyClient.java class

```
1  import java.rmi.Naming;
2
3  public class MyClient {
4  public static void main(String[] args ){
5      try {
6          MyData.info();
7      } catch (Exception e){
8          System.out.println("User data cannot be displayed!");
9          e.printStackTrace();
10         return;
11     }
12     double result;
13     CalcObject rObject;
14     CalcObject2 rObject2;
15     ResultType result2;
16     InputType inObj;
17     System.setProperty("java.security.policy", "D:\\2022_DCS\\zf\\cw3a_te\\serverRMI\\src\\srv.policy");
18     if (args.length == 0){
19         System.out.println("You have to enter RMI object address in the form: // host_address/service_name");
20         return;
21     }
22     String addr = args[0];
23     String addr2 = args[1];
24
25     try{
26         rObject = (CalcObject) java.rmi.Naming.lookup(addr);
27     } catch (Exception e){
28         System.out.println("Reference to " + addr + " can't be retrieved!");
29         e.printStackTrace();
30         return;
31     }
32
33     try {
34         rObject2 = (CalcObject2) Naming.lookup(addr2);
35     }
36     catch (Exception e) {
37         System.out.println("Reference to " + addr2 + " can't be retrieved!");
38         e.printStackTrace();
39         return;
40     }
41     System.out.println("Reference to "+addr + " is retrieved.");
42     System.out.println("Reference to "+addr2 + " is retrieved.");
43
44     inObj = new InputType();
45     inObj.x1 = 1001.1;
46     inObj.x2 = 2002.2;
47     inObj.operation = "sub";
48
49     try{
50         result = rObject.calculate( a: 10.1, b: 20.2);
51     } catch (Exception e){
52         System.out.println("Remote call error!");
53         e.printStackTrace();
54         return;
55     }
56
57     try {
58         result2 = rObject2.calculate(inObj);
59     } catch (Exception e) {
60         System.out.println("Remote call error!");
61         e.printStackTrace();
62         return;
63     }
64
65     System.out.println("Result = "+result);
66     System.out.println("Result2 = " + result2.result + ", " + result2.result_description);
67     return;
68 }
```

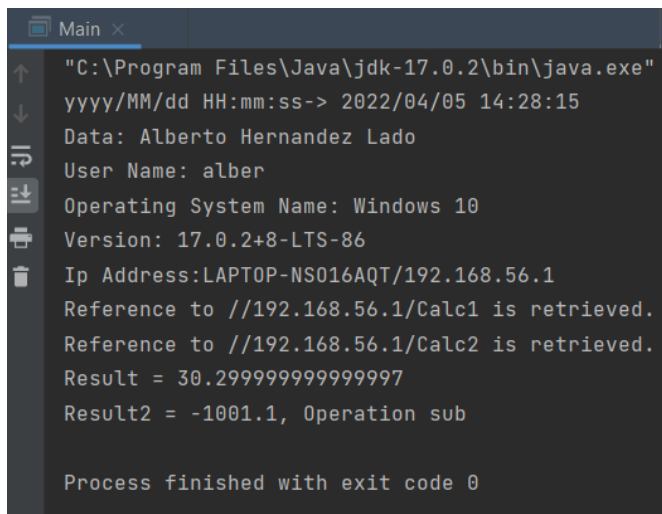
RUNNING AND TESTING THE APPLICATION

Running MyServer.java



```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe"
yyyy/MM/dd HH:mm:ss-> 2022/04/05 14:26:44
Data: Alberto Hernandez Lado
User Name: alber
Operating System Name: Windows 10
Version: 17.0.2+8-LTS-86
Ip Address:LAPTOP-NS016AQT/192.168.56.1
//192.168.56.1/Calc1
Server is registered now :- )
//192.168.56.1/Calc2
Server2 is registered now!
```

Running MyClient.java



```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe"
yyyy/MM/dd HH:mm:ss-> 2022/04/05 14:28:15
Data: Alberto Hernandez Lado
User Name: alber
Operating System Name: Windows 10
Version: 17.0.2+8-LTS-86
Ip Address:LAPTOP-NS016AQT/192.168.56.1
Reference to //192.168.56.1/Calc1 is retrieved.
Reference to //192.168.56.1/Calc2 is retrieved.
Result = 30.299999999999997
Result2 = -1001.1, Operation sub

Process finished with exit code 0
```

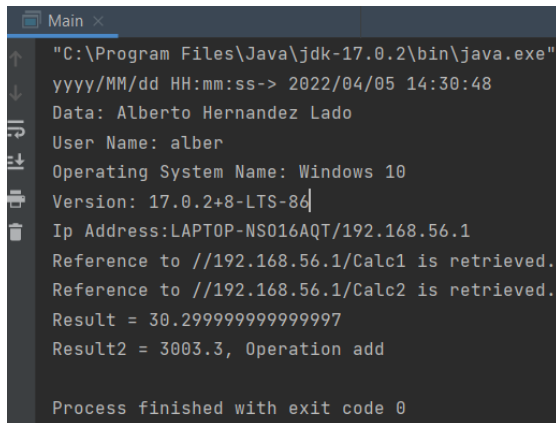
Testing operations

Addition

We change the code in MyClient.java. We have to edit the operation variable of the InputType instance to "add" for addition operation:

```
inObj = new InputType();
inObj.x1 = 1001.1;
inObj.x2 = 2002.2;
inObj.operation = "add";
```

When we execute MyClient.java we get the result of the operation in Result2. In this case, $1001.1 + 2002.2 = 3003.3$.



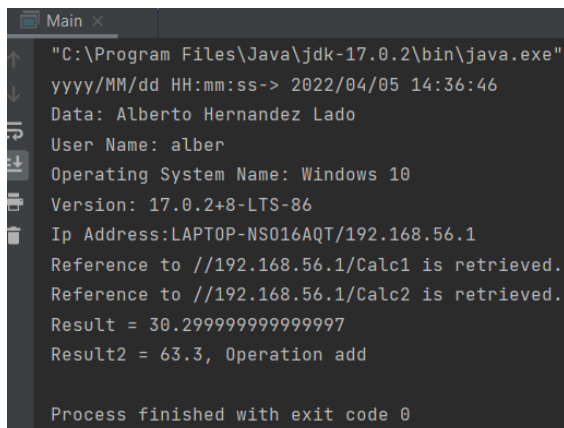
```

Main x
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe"
yyyy/MM/dd HH:mm:ss-> 2022/04/05 14:30:48
Data: Alberto Hernandez Lado
User Name: alber
Operating System Name: Windows 10
Version: 17.0.2+8-LTS-86
Ip Address:LAPTOP-NS016AQT/192.168.56.1
Reference to //192.168.56.1/Calc1 is retrieved.
Reference to //192.168.56.1/Calc2 is retrieved.
Result = 30.299999999999997
Result2 = 3003.3, Operation add
Process finished with exit code 0
```

Another example for addition:

```
inObj = new InputType();
inObj.x1 = 35.4;
inObj.x2 = 27.9;
inObj.operation = "add";
```

And the result:



```

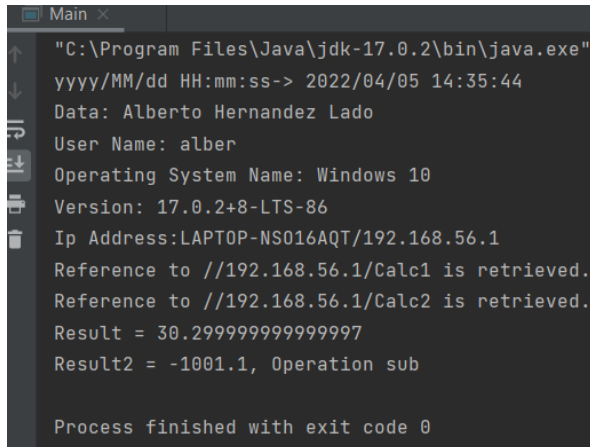
Main x
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe"
yyyy/MM/dd HH:mm:ss-> 2022/04/05 14:36:46
Data: Alberto Hernandez Lado
User Name: alber
Operating System Name: Windows 10
Version: 17.0.2+8-LTS-86
Ip Address:LAPTOP-NS016AQT/192.168.56.1
Reference to //192.168.56.1/Calc1 is retrieved.
Reference to //192.168.56.1/Calc2 is retrieved.
Result = 30.299999999999997
Result2 = 63.3, Operation add
Process finished with exit code 0
```


Subtraction

As we have done before, we have to change the operation variable for subtract:

```
inObj = new InputType();
inObj.x1 = 1001.1;
inObj.x2 = 2002.2;
inObj.operation = "sub";
```

Now we execute MyClient.java and we get $1001.1 - 2002.2 = -1001.1$:

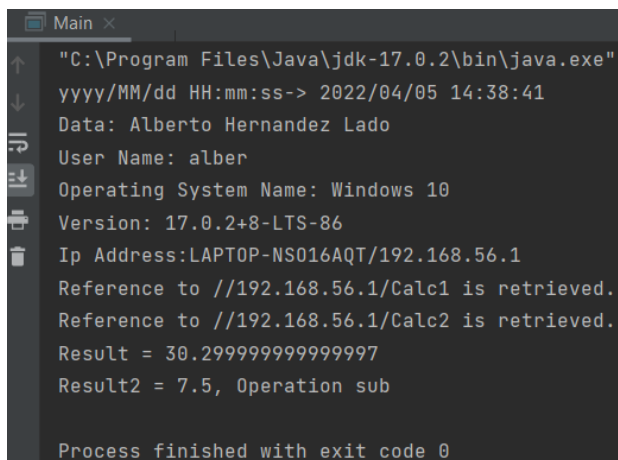


```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe"
yyyy/MM/dd HH:mm:ss-> 2022/04/05 14:35:44
Data: Alberto Hernandez Lado
User Name: alber
Operating System Name: Windows 10
Version: 17.0.2+8-LTS-86
Ip Address:LAPTOP-NS016AQT/192.168.56.1
Reference to //192.168.56.1/Calc1 is retrieved.
Reference to //192.168.56.1/Calc2 is retrieved.
Result = 30.299999999999997
Result2 = -1001.1, Operation sub
Process finished with exit code 0
```

Another example for addition:

```
inObj = new InputType();
inObj.x1 = 35.4;
inObj.x2 = 27.9;
inObj.operation = "sub";
```

And the result:



```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe"
yyyy/MM/dd HH:mm:ss-> 2022/04/05 14:38:41
Data: Alberto Hernandez Lado
User Name: alber
Operating System Name: Windows 10
Version: 17.0.2+8-LTS-86
Ip Address:LAPTOP-NS016AQT/192.168.56.1
Reference to //192.168.56.1/Calc1 is retrieved.
Reference to //192.168.56.1/Calc2 is retrieved.
Result = 30.299999999999997
Result2 = 7.5, Operation sub
Process finished with exit code 0
```