L3 - RMI

1. Introduction

1.1 Objectives

The main objective of this lab is that you understand the concepts behind object oriented applications based on Java RMI and that you learn how to develop simple client-server applications using Java RMI.

2. Application

The application you shall develop is a variant of client-server application described in the first lab, this time using Java RMI.

Remote Interface

As a first step, you shall define the remote interface to be implemented by the server.

You need also to develop a class that implements this interface. This class can be the server class itself, see below, or a different class. This class should also print messages in the standard output that reveal the actions it performs.

Server

The server application must create a remote object that implements the remote interface, and register that object on the **`rmiregistry`**. The name to which the remote object shall be bound (**`<remote_object_name>`**) is passed to the server as a command line argument. Therefore, the server shall be invoked as follows:

```
java Server <remote_object_name>
```

To observe the operation of your solution, the server shall reveal what it is doing by printing messages with, e.g., the following format:

```
<oper> <opnd> * :: <out>
```

where:

`<oper>` should be "register" or "lookup", according to the operation invoked;

`<opnd>` **`*`** is the list of operands received in the request for the operation;

`<out>` is the value returned by the operation, if any.

Client

The client program for testing your server implementation shall be invoked as follows:

> **`java Client <host_name> <remote_object_name> <oper> <opnd>*`**
> where:
>
> > **`<host_name>`** is the name of the host where the server runs;
> > **`<remote_object_name>`** is the name the server bound the remote object to;
> > **`<oper>`** is "register" or "lookup", depending on the operation to invoke;
> > **`<opnd>*`** is the list of operands of the specified operation:
> >
> > > **`<DNS name> <IP Address>`**, for `register`;
> > > **`<DNS name>`**, for `lookup`.

The client shall invoke the remote operation on the remote object. To observe its operation, the client shall reveal what it is doing by printing messages with the format used by the server, i.e.:

> **`<oper> <opnd>*:: <out>`**
> where:
>
> > **`<oper> <opnd>*`** and have the same meaning and format of homonyms arguments from the command line and
> > **`<out>`** is the value returned by the operation invoked or "ERROR" if any error occurs.

The client shall terminate after receiving the response and printing the appropriate messages.

3 Documentation

- Getting Started Using Java RMI
  - A nice tutorial on Java RMI from Sun (now Oracle)