

20. Java RMI

In this lab, we experiment with the Java Remote Method Interface (RMI).

1. Addition

Here are the files that make up the RMI client side of the RMI adder from the lecture:

- [AdditionClient.java](#)
- [AdderInterface.java](#)
- [security.policy](#)

To run it correctly, use the following VM argument (in Eclipse, go to run configurations > Arguments > VM Arguments):

```
-Djava.security.policy=file:${workspace_loc}/PROJECTNAME/  
security.policy (replace PROJECTNAME with your project name)
```

Task 1: Connect to the addition server on the lab teacher's machine (see the whiteboard for the IP address, the resource name is: ourAdder) and perform an addition.

Next, set up your own server. Here are the files for the server side:

- [AdditionServer.java](#)
- [Adder.java](#)
- [AdditionInterface.java](#)
- [security.policy](#)

Task 2: Perform an addition on your own server. Again, set up the security policy file in the VM arguments as shown above.

2. Print Server

Write a client and a server so that the client can send a string to the server and the server prints the string to standard output. Use the following interface:

- [PrintInterface.java](#)

An instance providing this service is running under the name "printer" on the machine shown on the whiteboard. You can use it for testing.

3. Echo Server

Write a client and a server so that the client can send a string to the server, the server modifies the string (e.g., attaches a prefix to it) and returns it to the client, and the client prints the string to standard output.

4. Broadcast Server

Write a server that accepts registrations from clients.

When a client sends a string to the server,

the server calls all registered clients and lets them print the string it received.

Hint: For this to work, you need to use remote object references.