



EXERCICE 1

PART A

Álvaro López Pereda and Alberto Hernandez Lado

```

1 package partA;
2
3 import org.apache.xmlrpc.WebServer;
4
5 public class ServerRPC {
6
7     public static void main(String[] args) {
8         try {
9             System.out.println("Starting XML_RPC server...");
10            int port = (10000 + 1);
11            WebServer server = new WebServer(port);
12            //Bellow server object with name MyServer is created and run:
13            server.addHandler("MyServer", new ServerRPC());
14            server.start();
15            System.out.println("Server started successfully.");
16            System.out.println("Listening on port: " + port);
17            System.out.println("Press Ctrl+C to stop the server.");
18
19            } catch (Exception exception) {
20                System.err.println("XML-RPC server: " + exception);
21            }
22        }
23
24
25        @SuppressWarnings("deprecation")
26        public Integer echo(int x, int y) {
27            return new Integer(x+y);
28        }
29
30
31
32
33
34
35
36
37
38
39
40
41
42

```

Server code.

```

Starting XML_RPC server...
Server started successfully.
Listening on port: 10001
Press Ctrl+C to stop the server.

```

Server initialized.

```

1 package partA;
2
3 import java.net.URL;
4
5
6
7 public class Ac implements AsyncCallback {
8
9     @Override
10    public void handleError(Exception arg0, URL arg1, String arg2) {
11        System.err.println("Exception: " + arg0);
12        System.out.println("URL: " + arg1);
13        System.out.println("Method: " + arg2);
14    }
15
16    @Override
17    public void handleResult(Object arg0, URL arg1, String arg2) {
18        System.out.println("Result: " + arg0);
19        System.out.println("URL: " + arg1);
20        System.out.println("Method: " + arg2);
21    }
22 }
23
24
25

```

AC's code.

```

1 package partA;
2
3 import java.util.Vector;
4
5
6 public class ClientRPC {
7
8     public static void main(String[] args) {
9         try {
10             XmlRpcClient srv = new XmlRpcClient("http://localhost:10001");
11             Vector<Integer> params = new Vector<Integer>();
12             params.addElement(13);
13             params.addElement(21);
14             Object result = srv.execute("MyServer.echo", params);
15             int value = ((Integer) result).intValue();
16             System.out.println(value);
17
18             Ac cb = new Ac();
19             Vector<Integer> params2 = new Vector<Integer>();
20             params2.addElement(1000);
21             srv.executeAsync("MyServer.execAsy", params2, cb);
22             System.out.println("Wywolano asynchronicznie");
23         } catch (Exception exception) {
24             System.err.println("XML-RPC client: " + exception);
25         }
26     }
27 }
28
29

```

Client's code.

```

34
Wywolano asynchronicznie
Result: 123
URL: http://localhost:10001
Method: MyServer.execAsy

```

Client initialized.