COMP713 Distributed and Mobile Systems

Laboratory 2: Remote Method Invocation

Introduction

The main aim of this laboratory is to learn the basic procedures to invoke remote methods through remote method invocation (RMI).

Students should refer to the course manual Section 1.4 for details on this topic.

Part 1: First Program Using RMI

The process of using RMI (described on page 28 of the manual) is demonstrated here through a very simple program.

The source files required for this part of the lab are `RMIGreeting.java`, `RMIServer.java`, and `RMIClient.java` (found in `rmi.zip`). `RMIGreeting.java` contains the interface that extend `java.rmi.Remote`. This interface defines two methods. The server program that implements this interface is in `RMIServer.java`. Finally, `RMIClient.java` contains the client program. The source codes can also be found on pages 31-32 of the course manual.

Open a command prompt and change (`cd`) to the folder with the Java source files. Create a new folder named "RMIexample". Compile the Java files using the command:

```
javac -d <folder path of source files> RMIServer.java
RMI.Greeting.java RMIClient.java
```

The full folder path of the source files should be used here. Three class files will be generated.

Question 1: Where are the class files located? Why do we need a folder named "RMIexample"?

An executable file `rmiregistry.exe` is located in the `bin` folder of your Java installation folder. Run this program from the `rmiclasses` folder by issuing the command: `start rmiregistry`. Then run the server program:

```
java -cp <folder path of source files>
-Djava.rmi.server.codebase=file:<folder path of source files>
RMIexample.RMIServer
```

If successful, the remote object should be bound with the RMI registry.

Open another Command Prompt. Change to the folder with the source files. Then start the client program:

```
java -cp <folder path of source files> RMIexample.RMIClient
```

Exercise 1: What is the greeting message when the client program is started? You can now set a new message. Run the client again to check that the new greeting message has been set.

Question 2: How does this program work?

Part 2: Guess Number Game

Exercise 2: Implement the guess number game (see Lab 1) using RMI. You will need to:

- Design the `GuessNumberGame` interface that extends `Remote`.
- Create a Server class that implements the GuessNumberGame interface.
- Create a client class that remotely access Server and play the game.
- Test your implementation.

****