Atividade de Laboratório - Java RMI

Desenvolvendo um Servidor RMI

Para aceitar chamadas remotas de métodos via RMI, um servidor deve estender a interface java.rmi.Remote e declarar os métodos que serão acessados remotamente.

Na interface RMI abaixo é definido um método hello(), que retorna uma saudação quando é chamado.

```
/** HelloWorld.java **/import
java.rmi.Remote;
import java.rmi.RemoteException;

public interface HelloWorld extends Remote {
   public String hello() throws RemoteException;
}
```

A exceção java.rmi.RemoteException indica erros na chamada remota, e deve ser prevista pelos métodos de interfaces RMI.

Agora vamos implementar a interface. Para facilitar o nosso trabalho, vamos usar como base a classe UnicastRemoteObject, que já implementa alguns métodos necessários para o servidor. Temos que criar também um construtor para o nosso servidor (neste caso, ele apenas chama o construtor da classe base).

```
/** HelloServer.java **/

import java.rmi.*;
import java.rmi.server.*;
import java.rmi.registry.*;

public class HelloServer extends UnicastRemoteObject implements
HelloWorld {
   public HelloServer() throws RemoteException {
      super();
   }
   // main()
   // hello()
}
```

Na função main() do servidor iremos criar um objeto que implementa a interface HelloWorld e registrá-lo como um servidor no registro do RMI, com o nome "HelloWorld", para que ele possa ser localizado pelos clientes.

```
public static void main(String[] args) {
   try {
      HelloServer obj = new HelloServer();
      Naming.rebind("//localhost/HelloWorld", obj);
   } catch (Exception ex) {
      System.out.println("Exception: " + ex.getMessage());
   }
}
```

Temos que implementar agora os métodos definidos na interface do servidor. Nesta aplicação, há apenas o método hello() para ser implementado.

```
public String hello() throws RemoteException {
   System.out.println("Executando hello()");
   return "Hello!!!";
}
```

Criando um Cliente RMI

Crie um cliente que obtenha uma referência para o servidor no registro RMI e chame o método hello(). Abaixo temos um exemplo de aplicação com interface de texto.

```
/** HelloClient.java **/

import java.rmi.Naming;

public class HelloClient {
   public static void main(String[] args) {
      try {
         HelloWorld obj = (HelloWorld)Naming.lookup("//" + args[0] +
"/HelloWorld");
         System.out.println("Mensagem do Servidor: " + obj.hello());
      } catch (Exception ex) {
         System.out.println("Exception: " + ex.getMessage());
      }
   }
}
```

O servidor pode ser acessado também usando uma aplicação gráfica ou através de um applet.

Compilando a Aplicação RMI

Edite e compile os arquivos acima usando o compilador Java ou um IDE da sua escolha. Crie também *stubs* e *skeletons* (ou seja, o código para comunicação via RMI) para a nossa aplicação usando compilador RMI (rmic).

```
> javac *.java
> rmic HelloServer
```

Executando a Aplicação RMI

Primeiramente, inicie o registro RMI com o comando:

```
> start rmiregistry
```

Em seguida, inicie o servidor:

```
> start java HelloServer
```

Finalmente, inicie o cliente com o comando:

```
> java HelloClient localhost
```

Tente também, com a ajuda de um colega, fazer a comunicação entre máquinas diferentes. Para isto, substitua 'localhost' pelo nome da máquina do outro grupo ao chamar o cliente.