Passos para desenvolver uma aplicação Hello World em Java RMI

O pacote java.rmi.registry tem uma classe chamada LocateRegistry (http://docs.oracle.com/javase/1.4.2/docs/api/java/rmi/registry/LocateRegistry.html) e uma interface Registry (http://docs.oracle.com/javase/1.4.2/docs/api/java/rmi/registry/Registry.html). Ambas servem para acessar o serviço de nomes do Java RMI.

Para iniciar o serviço de nomes, pode-se (1) chamar o executável rmiregistry via linha de comando (2) chamar rmiregistry dentro do código através do método exec() da classe Runtime ou (3) usar a classe LocateRegistry. Quero que vocês usem a terceira opção. Se optarem por usar as opções (1) e (2) é necessário chamar Naming.metodo() como nos slides vistos em sala.

1. Para iniciar um serviço de nomes no servidor:
Registry referenciaServicoNomes = LocateRegistry.createRegistry(int port);

De posse da referência do Serviço de Nomes (SN), pode-se chamar seus métodos bind(), rebind(), unbind(), lookup() e list() --> ver classe Naming (http://docs.oracle.com/javase/1.4.2/docs/api/java/rmi/Naming.html) ou interface Registry.

2. Para o cliente obter uma referência do SN, utiliza-se o método getRegistry() da classe LocateRegistry:
Registry referenciaServicoNomes = LocateRegistry.getRegistry(String maquinaServidor, int portaSN);

Passos de Implementação:

1. Criar 2 projetos distintos: um para o cliente chamado Cliente_HelloWorld e outro para o servidor chamado Servidor_HelloWorld.

2. Criar 2 pacotes distintos, um em cada projeto, ambos com o mesmo nome: um para o cliente chamado HelloWorld e outro para o servidor chamado HelloWorld.

3. Criar a interface Servidor - InterfaceServ - estendendo a interface Remote e inserir apenas um método chamar() que recebe dois parâmetros de entrada: uma String com o nome do cliente e a referência do cliente (tipo InterfaceCli);

4. Criar a interface Cliente - InterfaceCli - estendendo a interface Remote e inserir apenas um método echo() que recebe um parâmetro de entrada: uma String qualquer;

5. Criar a classe servente do servidor - ServImpl - que implementa a interface InterfaceServ. Quando um cliente invocar o método chamar(), o servidor receberá o nome e a referência do cliente e invocará o método echo();

6. Criar a classe servente do cliente - CliImpl - que implementa a interface InterfaceCli. Quando um servidor invocar o método echo(), o cliente apenas mostrará a string recebida na tela. Nessa classe o cliente deve ter a referência do servidor e com esta ele poderá chamar o método chamar() do servidor e passará seu nome e sua referência (estando em CliImpl, basta passar this);

7. Criar classe Servidor com método main que vai: iniciar o serviço de nomes (usar preferencialmente a classe LocateRegistry e método createRegistry), criar uma instância da classe ServImpl e registrar a referência da sua aplicação (tipo InterfaceServ) no serviço de nomes. Obs: a referência é o resultado da criação da instância de ServImpl;

8. Criar classe Cliente com método main que vai: obter referência do serviço de nomes que está executando no servidor (usar preferencialmente a classe LocateRegistry e método getRegistry), criar uma instância da classe CliImpl passando como argumento a referência do SN.

ATENÇÃO: as interfaces do cliente e do servidor devem estar as duas nos dois pacotes - desafio de abertura - divulgação de interfaces.

Existem várias outras formas de implementar, mas vamos fazer assim para facilitar o entendimento.

Boa sorte!!!

Cristina.