|  |  |  |
| --- | --- | --- |
| img-logo.jpg | UNIVERSIDADE REGIONAL DE BLUMENAU - FURB  CENTRO DE CIÊNCIAS EXATAS E NATURAIS  DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO  DISCIPLINA: Sistemas Distribuídos | Prof. Aurélio Hoppe |

**Tutorial: Aprenda a escrever sua primeira aplicação RMI usando NetBeans**

Neste tutorial mostrarei passo-a-passo como escrever um aplicação RMI bem simples usando a IDE NetBeans (aqui usei a versão 8.0.1). Note, contudo, que isso não o(a) liberará da tarefa de executar alguns comandos na janela do Prompt de comando. Desta forma, se você ainda não sabe como fazê-lo e não se esqueça de veriﬁcar se as variáveis de ambiente PATH e CLASSPATH encontram-se devidamente conﬁguradas.

**Sobre a aplicação**

A aplicação RMI que escreveremos como exemplo é realmente muito simples. Trata-se de uma calculadora que recebe dois inteiros e devolve sua soma. A aplicação cliente se conectará à aplicação servidor, enviará dois valores inteiros e o servidor devolverá a soma dos valores informados. Lembre-se de que usaremos apenas RMI, nada de sockets envolvidos neste tutorial (somente nos bastidores, é claro).

**Escrevendo a aplicação RMI servidor**

O primeiro passo para escrever a aplicação cliente é abrir o NetBeans. Em seguida vá até a opção "Arquivo -> Novo Projeto". Na janela de novo projeto escolha "Java" de um lado e "Aplicativo Java" do outro. Clique o botão "Próximo". Dê o nome "CalculadoraServidor" ao projeto e salve-o no seu diretório de projetos, por exemplo "C:\temp\SD" de forma que a pasta do projeto seja "C:\temp\SD\CalculadoraServidor". Clique o botão "Finalizar".

Vamos começar criando a interface remota, que será compartilhada tanto pela aplicação servidor quanto pela aplicação cliente. Assim, vá na opção "Arquivo -> Novo Arquivo". Na janela Novo arquivo escolha Java de um lado e Interface Java do outro. Clique o botão Próximo. Dê o nome "CalculadoraServerInterface" à interface e selecione o pacote "calculadoraservidor". Clique o botão "Finalizar".

Altere o código do arquivo CalculadoraServerInterface.java para a versão mostrada abaixo:

package calculadoraservidor;

import java.rmi.\*;

public interface CalculadoraServerInterface extends Remote{

// método público que recebe dois valores inteiros e

// retorna sua soma

public int somar(int a, int b) throws RemoteException;

}

Aqui nós temos um método somar() que recebe dois inteiros e retorna sua soma. Vamos agora escrever uma classe que implementa esta interface. Vá na opção "Arquivo -> Novo Arquivo". Na janela Novo arquivo escolha "Java" de um lado e "Classe Java" do outro. Clique no botão "Próximo". Dê o nome "CalculadoraServerInterfaceImpl" à classe e selecione o pacote "calculadoraservidor". Clique no botão "Finalizar".

Altere o código do arquivo CalculadoraServerInterfaceImpl.java para a versão mostrada abaixo:

package calculadoraservidor;

import java.rmi.\*;

import java.rmi.server.\*;

public class CalculadoraServerInterfaceImpl extends

UnicastRemoteObject

implements CalculadoraServerInterface{

public CalculadoraServerInterfaceImpl() throws RemoteException{

// construtor padrão

}

// método público que recebe dois valores inteiros e

// retorna sua soma

public int somar(int a, int b) throws RemoteException{

System.out.println("Somar: A " + a + " B " + b);

return a + b;

}

}

Agora já podemos implementar o servidor e registrá-lo com o RMI server. Crie a classe Main no pacote calculadoraservidor e altere o código do arquivo Main.java para a seguinte versão:

package calculadoraservidor;

import java.rmi.registry.\*;

public class Main {

public static void main(String[] args) {

try {

CalculadoraServerInterface sdrmi = new

CalculadoraServerInterfaceImpl();

Registry registry = LocateRegistry.getRegistry();

registry.rebind("CalculadoraServerInterfaceImpl", sdrmi);

System.out.println("Servidor Calculadora " + sdrmi +

" registrado e pronto para aceitar solicitações.");

}

catch (Exception ex) {

System.out.println("Houve um erro: " + ex.getMessage());

}

}

}

Pronto! A aplicação servidor já está concluída. Pressione F6 para executá-la. Teremos o seguinte erro de tempo de execução:

Houve um erro: Connection refused to host: 127.0.0.1; nested exception is: java.net.ConnectException: Connection refused: connect

Isso aconteceu porque não iniciamos o RMI Registry. Para isso, abra um prompt de comandos e execute o seguinte comando:

start rmiregistry

Isso fará com que uma nova janela de comando seja aberta. Não a feche. O RMI Registry está sendo executado nela. Volte à aplicação e pressione F6 novamente. Agora o erro de tempo de execução é:

Houve um erro: RemoteException occurred in server thread;

nested exception is: java.rmi.UnmarshalException:

error unmarshalling arguments;

nested exception is:

java.lang.ClassNotFoundException: calculadoraservidor.CalculadoraServerInterface

Este erro ocorreu porque o RMI Registry não conseguiu encontrar nenhuma informação a respeito da interface CalculadoraServerInterface. Para contornarmos este problema, vamos apenas construir a aplicação (pressionando F11 ou acessando "Executar -> Construir projeto principal") e executá-la a partir do jar gerado (CalculadoraServidor.jar). Assim, abra uma janela de terminal e vá até o diretório de sua localização:

cd C:\SD\CalculadoraServidor\dist

e dispare o comando:

java -jar CalculadoraServidor.jar

Mesmo erro, não? Pare o RMI (Ctrl + C), feche todas as janelas de comando e adicione o valor "C:\temp\SD\CalculadoraServidor\dist\CalculadoraServidor.jar;" à sua variável de ambiente CLASSPATH. Inicie o RMI novamente, abra uma nova janela de terminal, vá

até o diretório "C:\temp\SD\CalculadoraServidor\dist" e execute o JAR novamente. Se tudo correu bem você terá o seguinte resultado:

Servidor Calculadora CalculadoraServerInterfaceImpl[UnicastServerRef [liveRef: [ endpoint:[127.0.0.1:4634](local),objID:[4124a53a:12e63b2ebea:-7ﬀf,

3839989767738511284]]]] registrado e pronto para aceitar solicitações.

Esta mensagem indica que o servidor RMI está 100% funcionando e pronto para atender às requisições do cliente. Hora de passarmos ao desenvolvimento do cliente.

Nota importante: Neste momento você já consegue executar a aplicação servidor a partir do NetBeans. Mas, é recomendável executá-la a partir do Prompt de Comando, uma vez que a aplicação servidor deverá estar rodando quando a aplicação cliente for executada.

**Escrevendo a aplicação RMI cliente**

Vamos então escrever a aplicação RMI cliente. Vá até a opção "Arquivo -> Novo Projeto". Na janela de novo projeto escolha "Java" de um lado e "Aplicativo Java" do outro. Clique o botão Próximo. Dê o nome "CalculadoraCliente" ao projeto e salve-o no diretório "C:\temp\SD" de forma que a pasta do projeto seja "C:\temp\SD\CalculadoraCliente". Clique o botão "Finalizar".

Na aplicação cliente nós precisaremos referenciar a interface CalculadoraServerInterface, que é parte da aplicação servidor.

Assim, clique com o botão direito em "Bibliotecas"e escolha a opção "Adicionar JAR/Pasta". Localize o diretório "C:\temp\SD\CalculadoraServidor\dist", selecione o JAR CalculadoraServidor.jar e clique o botão Open.

Agora modiﬁque a classe Main, no arquivo Main.java do cliente para a seguinte versão:

package calculadoracliente;

import java.rmi.registry.\*;

import calculadoraservidor.\*;

public class Main {

public static void main(String[] args) {

try {

Registry registry = LocateRegistry.getRegistry("localhost");

CalculadoraServerInterface c = (CalculadoraServerInterface)

registry.lookup("CalculadoraServerInterfaceImpl");

System.out.println("O objeto servidor " + c + " foi encontrado

com sucesso.\n");

// vamos efetuar uma soma?

System.out.println("A soma de 2 + 5 é: " + c.somar(2, 5));

}

catch(Exception ex){

System.out.println(ex);

}

}

}

A aplicação cliente pode ser executada a partir do NetBeans mesmo (não se esqueça de ativar o RMI Registry e executar a aplicação Servidor, é claro). Basta pressionar F6 e teremos o seguinte resultado:

O objeto servidor Proxy [CalculadoraServerInterface, RemoteObjectInvocationHandler[UnicastRef[liveRef: endpoint:[192.168.1.64:4649](remote),objID:[4d4c1a57:12e63bb67ba:-7ﬀf, 3392955188099866234]]]]] foi encontrado com sucesso.

A soma de 2 + 5 é: 7

Estabelecer as variáveis de ambiente:     
  
 JAVA\_HOME  =  C:\Arquivos de programas\Java\jdk1.8.0\_xx  
 CLASSPATH  =  .;C:\Arquivos de programas\Java\jdk1.8.0\_xx\lib

(lembre do ponto,  seguido de ponto e vírgula, no antes de C:\, no estabelecimento da variável de ambiente CLASSPATH).                         

PATH = C:\Arquivos de programas\Java\jdk1.8.0\_xx\bin  
  
Inicializar o computador