

Prática 9

1. Exemplo de passagem de objetos usando RMI.

- Interface CalculadoraSimples.java

```
import java.rmi.Remote;  
import java.rmi.RemoteException;  
  
public interface CalculadoraSimples extends Remote {  
  
    int soma(Operandos ops) throws RemoteException;  
  
}
```

- Classe Operandos.java

```
import java.io.Serializable;  
  
public class Operandos implements Serializable{  
  
    /**  
     *  
     */  
    private static final long serialVersionUID = 1L;  
  
    public int n1;  
  
    public int n2;  
  
    public Operandos(int v1, int v2) {  
        this.n1 = v1;  
        this.n2 = v2;  
    }  
  
}
```

- Classe ImplCalculadoraSimples.java

```
public class ImplCalculadoraSimples implements CalculadoraSimples{  
  
    @Override  
    public int soma(Operandos ops) {  
  
        System.out.println("Somando " + ops.n1 + " + " + ops.n2);  
        return ops.n1 + ops.n2;  
    }  
  
}
```

```
}  
  
}
```

- Classe Servidor.java

```
import java.rmi.registry.LocateRegistry;  
import java.rmi.registry.Registry;  
import java.rmi.server.UnicastRemoteObject;  
  
public class Servidor {  
  
    public static void main(String args[]) {  
  
        System.setProperty("java.rmi.server.hostname", "127.0.0.1");  
  
        try {  
  
            ImplCalculadoraSimples refObjetoRemoto = new ImplCalculadoraSimples();  
  
            CalculadoraSimples skeleton = (CalculadoraSimples)  
UnicastRemoteObject.exportObject(refObjetoRemoto, 0);  
  
            LocateRegistry.createRegistry(20002);  
  
            //Registry registry =  
LocateRegistry.getRegistry(InetAddress.getLocalHost().getHostAddress());  
            Registry registro = LocateRegistry.getRegistry(20002);  
  
            registro.bind("CalcSimples", skeleton);  
  
            System.err.println("Servidor pronto:");  
  
        } catch (Exception e) {  
            System.err.println("Server exception: " + e.toString());  
            e.printStackTrace();  
        }  
    }  
}
```

- Classe Cliente.java

```
import java.rmi.registry.LocateRegistry;  
import java.rmi.registry.Registry;  
import java.util.Scanner;  
  
public class Cliente {  
  
    private Cliente() {}  
  
    public static void main(String[] args) {
```

```

Scanner teclado = new Scanner(System.in);

System.out.println("Informe o nome/endereço do RMIRegistry:");
String host = teclado.nextLine();

try {

    Registry registro = LocateRegistry.getRegistry(host, 20002);

    CalculadoraSimples stub = (CalculadoraSimples) registro.lookup("CalcSimples");

    int resposta = stub.soma(new Operandos(10, 20));

    System.out.println("Retorno do método soma(10, 20): " + resposta);

    teclado.close();

} catch (Exception e) {
    System.err.println("Client exception: " + e.toString());
    e.printStackTrace();
}
}

```

Exercícios:

1. Crie um projeto no Eclipse, codifique, execute e observe o funcionamento dessa aplicação.
2. Desenvolva uma aplicação onde um servidor realiza o gerenciamento de uma agenda de contatos.
 - Cada contato tem um nome, telefone e endereço.
 - Operações do servidor: cadastrar, listar, alterar e remover contatos.