

```
// Declaracao da interface para o objeto remoto

import java.rmi.*;

// interface para objetos que implementam uma calculadora

public interface Calculadora extends Remote
{
    double soma( double x, double y ) throws RemoteException;

    double subtrai( double x, double y ) throws RemoteException;

    double multiplica( double x, double y ) throws RemoteException;

    double divide( double x, double y ) throws RemoteException;
}
```

CLIENTE:

```
// Definição do cliente que chama um método do objeto remoto

import java.util.*; // para manipular datas
import java.rmi.*;

public class SuperCalc
{
    // construtor da classe
    public SuperCalc ( ) { }

    public static void main ( String args [ ] )
    {
        double a = 6.0;
        double b = 2.0;

        String IP_Servidor = "127.0.0.1";
        String nome_objeto_remoto = "CalculadoraBasica";

        try
        {
            Calculadora calculadoraRemota = (Calculadora)
                Naming.lookup ( "rmi://" + IP_Servidor + "/" + nome_objeto_remoto );

            double s = calculadoraRemota.soma( a, b );
            System.out.println( "soma de " + a + " e " + b + " = " + s );

            double d = calculadoraRemota.subtrai( a, b );
            System.out.println( "diferença entre " + a + " e " + b + " = " + d );

            double p = calculadoraRemota.multiplica( a, b );
            System.out.println( "produto de " + a + " por " + b + " = " + p );

            double q = calculadoraRemota.divide( a, b );
            System.out.println( "divisão de " + a + " por " + b + " = " + q );
        }
        catch ( Exception e )
        {
            e.printStackTrace();
        }
    }
}
```

```

// Implementacao da classe do objeto remoto (calendario)
// Implementa a interface iCalendar

// Definicao do servidor que instancia a classe, isto e,
// cria o objeto remoto e o registra no servico de nomes

import java.util.*; // para manipular datas
import java.rmi.*;
import java.rmi.registry.*;
import java.rmi.server.*;

public class CalculadoraHP
    extends UnicastRemoteObject
    implements Calculadora
{
    // construtor:
    public CalculadoraHP ( ) throws RemoteException { }

    public double soma( double x, double y ) throws RemoteException
    {
        return x + y;
    }

    public double subtrai( double x, double y ) throws RemoteException
    {
        return x - y;
    }

    public double multiplica( double x, double y ) throws RemoteException
    {
        return x * y;
    }

    public double divide( double x, double y ) throws RemoteException
    {
        return x / y;
    }

    public static void main ( String args [ ] )
    {
        String nome_objeto_remoto = "CalculadoraBasica";

        try
        {
            // localiza o Registry
            LocateRegistry.getRegistry ( );
            // cria o objeto remoto
            Calculadora calc = new CalculadoraHP ( );
            // registra o objeto com o Registry local
            Naming.rebind ( "rmi://" + nome_objeto_remoto, calc );

            System.out.println ( "Calculadora HP pronta" );
            System.out.println ( );
        }
        catch ( Exception e )
        {
            System.out.println ( e );
        }
    }
}

```

```

// Implementacao da classe do objeto remoto (calendario)
// Implementa a interface iCalendar

// Definicao do servidor que instancia a classe, isto e,
// cria o objeto remoto e o registra no servico de nomes

import java.util.*; // para manipular datas
import java.rmi.*;
import java.rmi.registry.*;
import java.rmi.server.*;

public class CalculadoraTexas
    extends UnicastRemoteObject
    implements Calculadora
{
    // construtor:
    public CalculadoraTexas ( ) throws RemoteException { }

    public double soma( double x, double y ) throws RemoteException
    {
        return x + y - 1.0;
    }

    public double subtrai( double x, double y ) throws RemoteException
    {
        return (x - y) * 2;
    }

    public double multiplica( double x, double y ) throws RemoteException
    {
        return x * y / 10.0;
    }

    public double divide( double x, double y ) throws RemoteException
    {
        return x / y + 10;
    }

    public static void main ( String args [ ] )
    {
        String nome_objeto_remoto = "CalculadoraBasica";

        try
        {
            // localiza o Registry
            LocateRegistry.getRegistry ( );
            // cria o objeto remoto
            Calculadora calc = new CalculadoraTexas ( );
            // registra o objeto com o Registry
            Naming.rebind ( "rmi://" + nome_objeto_remoto, calc );

            System.out.println ( "Calculadora Texas pronta" );
            System.out.println ( );
        }
        catch ( Exception e )
        {
            System.out.println ( e );
        }
    }
}

```