**Sô phức:   
  
ComplexNumber**

package Bai2;

import java.io.Serializable;

public class ComplexNumber implements Serializable {

private double real;

private double imaginary;

public ComplexNumber(double real, double imaginary) {

this.real = real;

this.imaginary = imaginary;

}

public double getReal() {

return real;

}

public void setReal(double real) {

this.real = real;

}

public double getImaginary() {

return imaginary;

}

public void setImaginary(double imaginary) {

this.imaginary = imaginary;

}

}

**ComplexNumberClient**

package Bai2;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

import java.util.Scanner;

public class ComplexNumberClient {

public static void main(String[] args) {

try {

Registry registry = LocateRegistry.getRegistry("localhost", 1091);

ComplexNumberInterface server = (ComplexNumberInterface) registry.lookup("ComplexNumberServer");

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the real part of complex number a: ");

double realA = scanner.nextDouble();

System.out.print("Enter the imaginary part of complex number a: ");

double imaginaryA = scanner.nextDouble();

ComplexNumber a = new ComplexNumber(realA, imaginaryA);

System.out.print("Enter the real part of complex number b: ");

double realB = scanner.nextDouble();

System.out.print("Enter the imaginary part of complex number b: ");

double imaginaryB = scanner.nextDouble();

ComplexNumber b = new ComplexNumber(realB, imaginaryB);

ComplexNumber sum = server.add(a, b);

ComplexNumber difference = server.subtract(a, b);

ComplexNumber product = server.multiply(a, b);

System.out.println("Sum: " + sum.getReal() + " + " + sum.getImaginary() + "i");

System.out.println("Difference: " + difference.getReal() + " + " + difference.getImaginary() + "i");

System.out.println("Product: " + product.getReal() + " + " + product.getImaginary() + "i");

// Close the Scanner

scanner.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**ComplexNumberInterface**

package Bai2;

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface ComplexNumberInterface extends Remote {

ComplexNumber add(ComplexNumber a, ComplexNumber b) throws RemoteException;

ComplexNumber subtract(ComplexNumber a, ComplexNumber b) throws RemoteException;

ComplexNumber multiply(ComplexNumber a, ComplexNumber b) throws RemoteException;

}

**ComplexNumberServer**

package Bai2;

import java.rmi.RemoteException;

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

import java.rmi.server.UnicastRemoteObject;

public class ComplexNumberServer implements ComplexNumberInterface {

public ComplexNumberServer() throws RemoteException {

super();

}

@Override

public ComplexNumber add(ComplexNumber a, ComplexNumber b) throws RemoteException {

return new ComplexNumber(a.getReal() + b.getReal(), a.getImaginary() + b.getImaginary());

}

@Override

public ComplexNumber subtract(ComplexNumber a, ComplexNumber b) throws RemoteException {

return new ComplexNumber(a.getReal() - b.getReal(), a.getImaginary() - b.getImaginary());

}

@Override

public ComplexNumber multiply(ComplexNumber a, ComplexNumber b) throws RemoteException {

double real = a.getReal() \* b.getReal() - a.getImaginary() \* b.getImaginary();

double imaginary = a.getReal() \* b.getImaginary() + a.getImaginary() \* b.getReal();

return new ComplexNumber(real, imaginary);

}

public static void main(String[] args) {

try {

ComplexNumberServer server = new ComplexNumberServer();

ComplexNumberInterface stub = (ComplexNumberInterface) UnicastRemoteObject.exportObject(server, 0);

Registry registry = LocateRegistry.createRegistry(1091);

registry.rebind("ComplexNumberServer", stub);

System.out.println("Server is ready.");

} catch (Exception e) {

e.printStackTrace();

}

}

}