A blue logo with a black background

Description automatically generated

BÁO CÁO THỰC HÀNH

**Bài thực hành số 05**

**Môn học:** Lập trình ứng dụng mạng

**Lớp:** NT109.O21.MMCL

**SINH VIÊN THỰC HIỆN:**

|  |  |
| --- | --- |
| **Họ và tên** | **MSSV** |
| Nguyễn Văn Huấn | 21522108 |

MỤC LỤC

[**A.** BÁO CÁO CHI TIẾT 2](#_Toc110261086)

[1. Nội dung 1 2](#_Toc110261087)

[a. Nội dung a 2](#_Toc110261088)

[b. Nội dung b 3](#_Toc110261089)

[2. Nội dung 2 3](#_Toc110261090)

[a. Nội dung a 3](#_Toc110261091)

[**B.** TÀI LIỆU THAM KHẢO 3](#_Toc110261092)

# BÁO CÁO CHI TIẾT

## Nội dung 1

### 1.1: Bài 1

PHẦN SERVER

A screenshot of a computer

Description automatically generated

(file IMath.java)

Giải thích: Khai báo package:

package rmiserver;

Dòng này xác định rằng giao diện IMath nằm trong gói rmiserver.

Nhập các thư viện cần thiết:

import java.rmi.Remote;

import java.rmi.RemoteException;

import logger.Logger;

Các import này đưa vào các lớp cần thiết từ thư viện Java:

java.rmi.Remote là một giao diện mà các giao diện từ xa phải mở rộng.

java.rmi.RemoteException là ngoại lệ có thể được ném ra trong quá trình gọi phương thức từ xa.

logger.Logger là một lớp tùy chỉnh dùng để ghi nhật ký.

Định nghĩa giao diện IMath:

public interface IMath extends Remote {

IMath là một giao diện công khai mở rộng Remote. Việc mở rộng Remote đánh dấu giao diện này là một giao diện từ xa.

Khai báo biến log:

static Logger log = new Logger("NguyenVanHuan\_21522108\_lab05");

Đây là một biến tĩnh log của kiểu Logger, được khởi tạo với tên "NguyenVanHuan\_21522108\_lab05". Biến này có thể được sử dụng để ghi lại các hoạt động trong giao diện này.

Định nghĩa các phương thức từ xa:

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

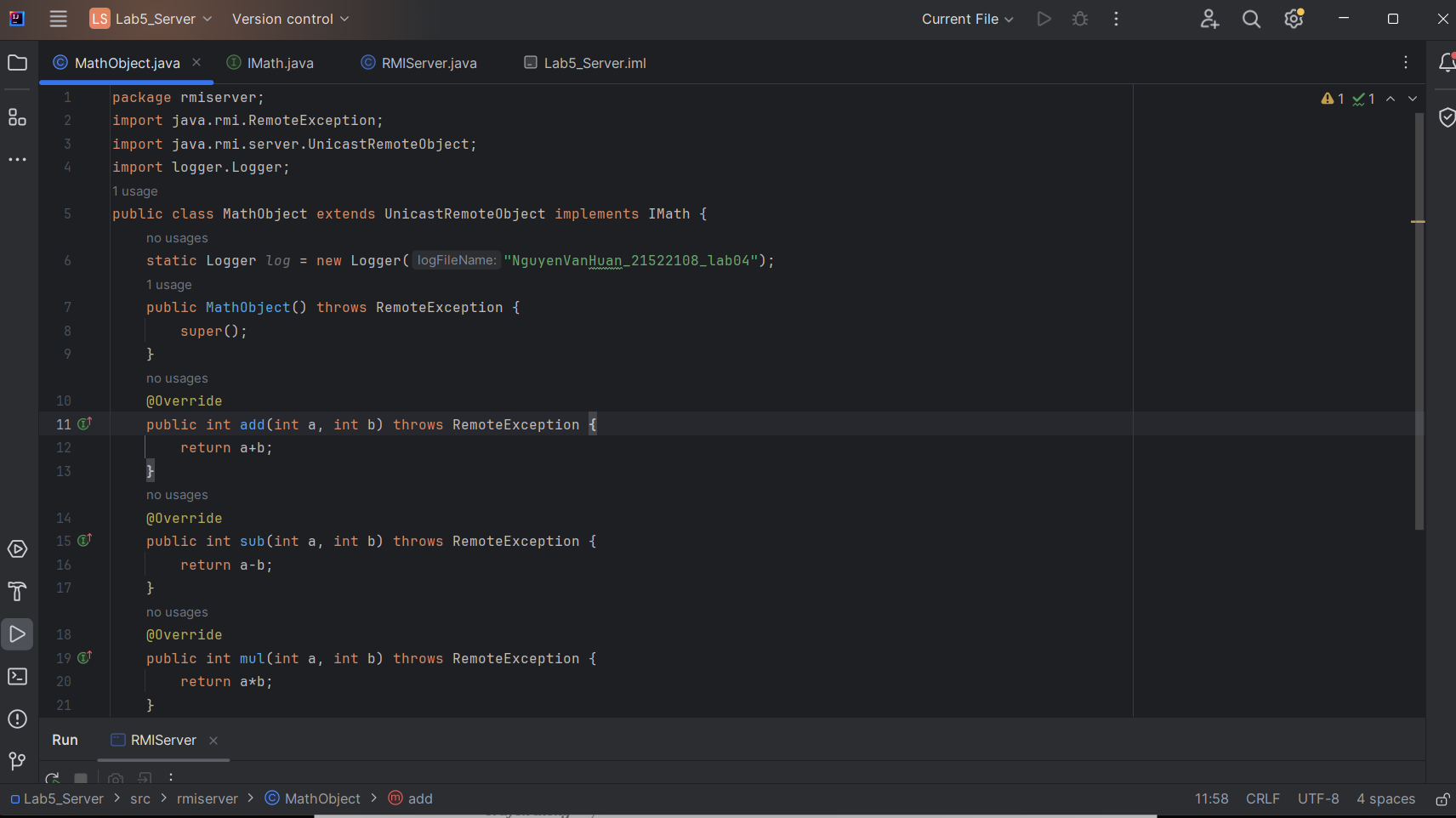
public int sub(int a, int b) throws RemoteException;

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

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

Các phương thức add, sub, mul, và div được khai báo trong giao diện IMath. Mỗi phương thức nhận vào hai tham số kiểu int và trả về một giá trị kiểu int.

Mỗi phương thức ném ra RemoteException để xử lý các lỗi có thể xảy ra trong quá trình thực thi từ xa.



(file MathObject.java)

Khai báo package:

package rmiserver;

Dòng này xác định rằng lớp MathObject nằm trong gói rmiserver.

Nhập các thư viện cần thiết:

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

import logger.Logger;

Các import này đưa vào các lớp cần thiết từ thư viện Java:

java.rmi.RemoteException là ngoại lệ có thể được ném ra trong quá trình gọi phương thức từ xa.

java.rmi.server.UnicastRemoteObject là lớp cơ sở cho các đối tượng từ xa đơn giản.

logger.Logger là một lớp tùy chỉnh dùng để ghi nhật ký.

Định nghĩa lớp MathObject:

public class MathObject extends UnicastRemoteObject implements IMath {

MathObject mở rộng UnicastRemoteObject và triển khai giao diện IMath.

UnicastRemoteObject cung cấp triển khai mặc định cho các đối tượng từ xa đơn giản và hỗ trợ cơ chế giao tiếp từ xa.

IMath là giao diện chứa các phương thức toán học từ xa.

Khai báo biến log:

static Logger log = new Logger("NguyenVanHuan\_21522108\_lab04");

Đây là một biến tĩnh log của kiểu Logger, được khởi tạo với tên "NguyenVanHuan\_21522108\_lab04". Biến này có thể được sử dụng để ghi lại các hoạt động trong lớp này.

Constructor MathObject:

public MathObject() throws RemoteException {

super();

}

Constructor này gọi constructor của lớp cha (UnicastRemoteObject) để thiết lập đối tượng từ xa.

Nó có thể ném RemoteException để xử lý các lỗi có thể xảy ra trong quá trình tạo đối tượng từ xa.

Triển khai các phương thức từ xa:

@Override

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

return a + b;

}

@Override

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

return a - b;

}

@Override

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

return a \* b;

}

@Override

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

return a / b;

}

Các phương thức add, sub, mul, và div được triển khai từ giao diện IMath.

Mỗi phương thức nhận vào hai tham số kiểu int và trả về kết quả của phép toán tương ứng.

Các phương thức này ném ra RemoteException để xử lý các lỗi có thể xảy ra trong quá trình thực thi từ xa.

A screenshot of a computer

Description automatically generated

(file RMIServer.java)

Khai báo package:

package rmiserver;

Dòng này xác định rằng lớp RMIServer nằm trong gói rmiserver.

Nhập các thư viện cần thiết:

import java.rmi.Naming;

import java.rmi.registry.LocateRegistry;

import logger.Logger;

Các import này đưa vào các lớp cần thiết từ thư viện Java và một lớp tùy chỉnh:

java.rmi.Naming cung cấp các phương thức để liên kết, giải liên kết, và tìm kiếm đối tượng từ xa trong RMI registry.

java.rmi.registry.LocateRegistry cung cấp các phương thức để tạo và lấy một tham chiếu tới RMI registry.

logger.Logger là một lớp tùy chỉnh dùng để ghi nhật ký.

Định nghĩa lớp RMIServer:

public class RMIServer {

Khai báo biến log:

static Logger log = new Logger("NguyenVanHuan\_21522108\_lab05");

Đây là một biến tĩnh log của kiểu Logger, được khởi tạo với tên "NguyenVanHuan\_21522108\_lab05". Biến này có thể được sử dụng để ghi lại các hoạt động trong lớp này.

Phương thức main:

public static void main(String[] args) {

// TODO code application logic here

try {

LocateRegistry.createRegistry(5000);

Naming.bind("rmi://localhost:5000/MyServices", new MathObject());

System.out.println("Server Started!!!");

} catch (Exception e) {

System.err.println("Error: " + e.getMessage());

}

}

Khối try:

LocateRegistry.createRegistry(5000);: Lệnh này tạo một RMI registry trên cổng 5000. RMI registry là một tên dịch vụ cho phép các client tìm kiếm các đối tượng từ xa dựa trên tên.

Naming.bind("rmi://localhost:5000/MyServices", new MathObject());: Lệnh này liên kết một đối tượng MathObject với tên "rmi://localhost:5000/MyServices" trong RMI registry. Client có thể sử dụng tên này để tìm và gọi các phương thức từ xa của MathObject.

System.out.println("Server Started!!!");: Thông báo ra console rằng máy chủ đã khởi động.

Khối catch:

catch (Exception e): Bắt bất kỳ ngoại lệ nào có thể xảy ra trong khối try.

System.err.println("Error: " + e.getMessage());: In thông báo lỗi ra console nếu có lỗi xảy ra.

A screenshot of a computer

Description automatically generated

(Chạy server)

PHẦN CLIENT

A screenshot of a computer

Description automatically generated

(Thêm module server vào trong các dependencies của phần client)

A screenshot of a computer

Description automatically generated

(file RMIClient.java)

Khai báo package:

package rmiclient;

Dòng này xác định rằng lớp RMIClient nằm trong gói rmiclient.

Nhập các thư viện cần thiết:

import logger.Logger;

import rmiserver.IMath;

import java.rmi.Naming;

Các import này đưa vào các lớp cần thiết:

logger.Logger là một lớp tùy chỉnh dùng để ghi nhật ký.

rmiserver.IMath là giao diện từ xa mà client sẽ sử dụng.

java.rmi.Naming cung cấp các phương thức để tìm kiếm và gọi các đối tượng từ xa trong RMI registry.

Định nghĩa lớp RMIClient:

public class RMIClient {

Khai báo biến log:

static Logger log = new Logger("NguyenVanHuan\_21522108\_lab05");

Đây là một biến tĩnh log của kiểu Logger, được khởi tạo với tên "NguyenVanHuan\_21522108\_lab05". Biến này có thể được sử dụng để ghi lại các hoạt động trong lớp này.

Phương thức main:

public static void main(String[] args) {

// TODO code application logic here

try {

IMath imath = (IMath) Naming.lookup("rmi://localhost:5000/MyServices");

int add = imath.add(2, 4);

int sub = imath.sub(10, 2);

int multi = imath.mul(5, 3);

int div = imath.div(100, 5);

System.out.println("Result: " + add + " " + sub + " " + multi + " " + div);

log.debug("Result: " + add + " " + sub + " " + multi + " " + div);

} catch (Exception e) {

System.err.println("Error: " + e.getMessage());

}

}

Khối try:

IMath imath = (IMath) Naming.lookup("rmi://localhost:5000/MyServices");: Lệnh này tìm kiếm đối tượng từ xa MathObject được đăng ký với tên "rmi://localhost:5000/MyServices" trong RMI registry và trả về một tham chiếu tới giao diện IMath.

int add = imath.add(2, 4);: Gọi phương thức add trên đối tượng từ xa imath với các tham số 2 và 4.

int sub = imath.sub(10, 2);: Gọi phương thức sub trên đối tượng từ xa imath với các tham số 10 và 2.

int multi = imath.mul(5, 3);: Gọi phương thức mul trên đối tượng từ xa imath với các tham số 5 và 3.

int div = imath.div(100, 5);: Gọi phương thức div trên đối tượng từ xa imath với các tham số 100 và 5.

System.out.println("Result: " + add + " " + sub + " " + multi + " " + div);: Hiển thị kết quả các phép toán trên console.

log.debug("Result: " + add + " " + sub + " " + multi + " " + div);: Ghi nhật ký kết quả các phép toán.

Khối catch:

catch (Exception e): Bắt bất kỳ ngoại lệ nào có thể xảy ra trong khối try.

System.err.println("Error: " + e.getMessage());: In thông báo lỗi ra console nếu có lỗi xảy ra.

A screen shot of a computer

Description automatically generated

(Kết quả sau khi chạy RMIClient)

A screenshot of a computer

Description automatically generated

(file log của phần client)

# TÀI LIỆU THAM KHẢO