BÁO CÁO THỰC HÀNH

**Môn học: Lập trình ứng dụng mạng**

**Buổi báo cáo: Lab 05**

**Tên chủ đề: Remote Method Invocation**

*GVHD: Trần Mạnh Hùng*

*Ngày thực hiện: 20/05/2024*

*Ngày nộp báo cáo: 27/05/2024*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT109.O21.MMCL.1

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Trần Lê Kiều Giang | 20521263 | 20521263@gm.uit.edu.vn |

1. **ĐÁNH GIÁ KHÁC:**

|  |  |
| --- | --- |
| **Nội dung** | **Kết quả** |
| Tổng thời gian thực hiện bài thực hành trung bình | 2 ngày |
| Link Video thực hiện  *(nếu có)* |  |
| Ý kiến *(nếu có)* | + Học được:   * Học được cấu trúc của một dự án RMI. * Biết cách tạo Server và Client, cách thêm RMIServer vào Client để thực thi. * Xây dựng được ứng dụng phân tán dùng RMI.   + Khó khăn:   * Cài đặt và chạy Corba. |

**Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

1. **Tạo ứng dụng phân tán dùng RMI**

* **Class RMI Server:**

A computer screen shot of a program

Description automatically generated

* Thiết lập một máy chủ RMI trong Java bằng cách tạo một đối tượng từ xa **LibraryImpl**, tạo một registry trên cổng 1263, và đăng ký đối tượng này với registry dưới tên "Library". Sau đó, ghi log thông báo rằng server đã sẵn sàng để các client có thể kết nối và gọi các phương thức từ xa. Đoạn mã cũng bao gồm khối **try-catch** để xử lý ngoại lệ và in ra thông tin lỗi nếu có sự cố xảy ra trong quá trình khởi tạo và đăng ký đối tượng từ xa.
* **Class Library:**

A screenshot of a computer program

Description automatically generated

* Định nghĩa một interface **Library** trong Java để sử dụng với RMI. Interface này extends từ **Remote** và khai báo ba phương thức từ xa: **searchBook**, **borrowBook**, và **returnBook**. Các phương thức này được sử dụng để tìm kiếm sách, mượn sách, và trả sách, tương ứng, và tất cả đều có thể ném ra ngoại lệ **RemoteException**. Điều này cho phép các đối tượng thực hiện từ xa của **Library** có thể được gọi qua mạng, hỗ trợ các thao tác liên quan đến quản lý sách trong một hệ thống thư viện phân tán.
* **Tạo class LibraryImpl:**

A computer screen shot of a program

Description automatically generated

* Định nghĩa lớp **LibraryImpl** kế thừa từ **UnicastRemoteObject** và cài đặt giao diện **Library** để sử dụng với Java RMI. Lớp này chứa một HashMap **books** để quản lý sách trong thư viện. Phương thức khởi tạo **LibraryImpl** khởi tạo bản đồ **books** với danh sách các sách mẫu và gán giá trị **null** cho người mượn. Điều này cho phép các phương thức từ xa được định nghĩa trong giao diện **Library** có thể được thực hiện trên đối tượng **LibraryImpl**, hỗ trợ các thao tác như tìm kiếm, mượn và trả sách trong hệ thống thư viện phân tán.
  + **Hàm BorrowBook:**

A computer code with text

Description automatically generated

* + Triển khai phương thức searchBook của lớp LibraryImpl, tìm kiếm sách theo tiêu đề. Phương thức này duyệt qua tập hợp các tiêu đề sách trong thư viện, chuyển tất cả sang chữ thường để so sánh không phân biệt hoa thường. Nếu tiêu đề sách chứa tiêu đề cần tìm (cũng chuyển sang chữ thường), tiêu đề đó sẽ được thêm vào danh sách kết quả. Cuối cùng, danh sách kết quả các tiêu đề sách phù hợp sẽ được trả về. Phương thức này có thể ném ra ngoại lệ RemoteException để xử lý các vấn đề liên quan đến RMI.
  + Hàm BorrowBook:

A computer code on a white background

Description automatically generated

* + Phương thức **borrowBook** trong lớp **LibraryImpl** kiểm tra xem sách có tồn tại và chưa được mượn (giá trị **null**) hay không. Nếu đúng, nó gán tên người dùng mượn sách vào bản đồ **books** và trả về **true**, ngược lại, trả về **false**. Phương thức này cũng có thể ném ra ngoại lệ **RemoteException** để xử lý các vấn đề liên quan đến RMI.
  + Hàm ReturnBook:

A close-up of a computer code

Description automatically generated

* + Phương thức **returnBook** trong lớp **LibraryImpl** kiểm tra xem sách có tồn tại trong thư viện và người dùng có phải là người đã mượn sách hay không. Nếu đúng, nó đặt giá trị của sách đó trong bản đồ **books** thành **null** (tức là sách đã được trả) và trả về **true**. Nếu không, nó trả về **false**. Phương thức này cũng có thể ném ra ngoại lệ **RemoteException** để xử lý các vấn đề liên quan đến RMI.
* **Class RMI Client:**

A screenshot of a computer

Description automatically generated

* Định nghĩa lớp **RMIClient** để tương tác với máy chủ RMI. Trong phương thức **main**, client kết nối tới registry RMI tại **localhost** trên cổng **1263**, sau đó tìm kiếm và lấy đối tượng **Library** từ registry. Một vòng lặp vô hạn được sử dụng để nhận lựa chọn từ người dùng, cho phép họ tìm kiếm sách, mượn sách, trả sách hoặc thoát khỏi chương trình. Người dùng nhập lựa chọn qua **Scanner**, và các hành động tương ứng sẽ được xử lý trong khối **switch** theo lựa chọn của họ. MyLogger được sử dụng để ghi log các hành động và lựa chọn của người dùng.
  + Case 1:

A computer code with text

Description automatically generated

* + Đoạn mã này là một phần của khối **switch** trong lớp **RMIClient**, xử lý trường hợp người dùng chọn tùy chọn tìm kiếm sách (case 1). Khi người dùng nhập tiêu đề sách, chương trình sẽ ghi log tiêu đề đó, gọi phương thức **searchBook** trên đối tượng **library** để tìm kiếm các sách khớp với tiêu đề, sau đó ghi log danh sách các sách tìm được. Cuối cùng, chương trình thoát khỏi case này với lệnh **break**.
  + Case 2:

A computer code with text

Description automatically generated

* + Đoạn mã này là một phần của khối switch trong lớp RMIClient, xử lý trường hợp người dùng chọn mượn sách (case 2). Khi người dùng nhập tiêu đề sách và tên của mình, chương trình ghi log cả hai thông tin này. Sau đó, chương trình gọi phương thức borrowBook trên đối tượng library để mượn sách và ghi log kết quả (true/false) cho biết việc mượn sách có thành công hay không. Cuối cùng, chương trình thoát khỏi case này với lệnh break.
  + Tạo hàm Case 3:

A computer code with text

Description automatically generated

* + Đoạn mã này là một phần của khối **switch** trong lớp **RMIClient**, xử lý trường hợp người dùng chọn trả sách (case 3). Khi người dùng nhập tiêu đề sách và tên của mình, chương trình ghi log cả hai thông tin này. Sau đó, chương trình gọi phương thức **returnBook** trên đối tượng **library** để trả sách và ghi log kết quả (true/false) cho biết việc trả sách có thành công hay không. Cuối cùng, chương trình thoát khỏi case này với lệnh **break**.
  + Tạo hàm Case 4:

A computer screen shot of a program

Description automatically generated

* + Đoạn mã này là phần cuối của khối **switch** trong lớp **RMIClient**, xử lý trường hợp người dùng chọn thoát (case 4) và trường hợp lựa chọn không hợp lệ (default). Nếu người dùng chọn thoát, chương trình ghi log thông báo, đóng đối tượng **scanner**, và thoát chương trình bằng **System.exit(0)**. Nếu người dùng nhập lựa chọn không hợp lệ, chương trình ghi log thông báo lỗi và yêu cầu người dùng thử lại. Khối **catch** bắt và in ra ngoại lệ nếu có lỗi xảy ra trong quá trình thực hiện.

1. **Kết quả**

**A screenshot of a computer

Description automatically generated**

Kết quả chạy Server.

A screenshot of a computer

Description automatically generated

Kết quả chạy Client và thực hiện tìm kiếm sách.

A white screen with black text

Description automatically generated

Kết quả khi thực hiện mượn sách.

A screenshot of a computer program

Description automatically generated

Kết quả khi thực hiện trả sách với sai tên user.

A screenshot of a computer

Description automatically generated

Kết quả khi thực hiện trả sách với tên user đúng.

A screenshot of a computer program

Description automatically generated

Kết quả khi thực hiện trả sách chưa mượn

A screenshot of a computer program

Description automatically generated

Kết quả khi chọn Exit. Chương trình sẽ dừng RMIClient.

**HẾT**