**I. Giới thiệu**

Java Remote Method Invocation (RMI) cho phép lập trình viên viết các chương trình Java phân tán. RMI giúp các chương trình Java có thể gọi các phương thức của các đối tượng (remote) có thể nằm trên một máy tính (remote) khác.

Kiến trúc giúp RMI hoạt động bao gồm ba thành phần: (1) một client, (2) một server, và (3) một dịch vụ định danh (rmiregistry). Client là quá trình mà nó gọi một phương thức trên một đối tượng remote. Ngược lại, server là quá trình sở hữu đối tượng remote. Đối tượng remote chỉ là một đối tượng thông thường khác trong không gian địa chỉ của server. Cuối cùng, **rmiregistry** là một máy chủ tên có một bảng ánh xạ các đối tượng tới các tham chiếu remote. Để rõ ràng, các máy chủ muốn cung cấp dịch vụ RMI cần đăng ký các đối tượng remote của họ tại **rmiregistry**. Ngoài ra, các client tìm kiếm một đối tượng remote cụ thể có thể liên hệ với **rmiregistry** để lấy tham chiếu remote cho đối tượng remote được yêu cầu.

**II. Bài tập**

Chương trình này sẽ là một hệ thống đặt phòng sử dụng RMI. Đối với bài tập này, bạn có thể giả sử rằng mọi người đang đặt phòng cho một khoảng thời gian cụ thể. Nói cách khác, hệ thống của chúng tôi không cho phép khách chọn ngày để đặt phòng. Có 5 loại phòng với các mức giá khác nhau:

* Chúng tôi có 10 phòng loại 0 là phòng đơn có giá 55 Euro mỗi đêm;
* Chúng tôi có 20 phòng loại 1 là phòng đôi có giá 75 Euro mỗi đêm;
* Chúng tôi có 5 phòng loại 2 là phòng đôi có hai giường có giá 80 Euro mỗi đêm;
* Chúng tôi có 3 phòng loại 3 là phòng ba có giá 150 Euro mỗi đêm;
* Chúng tôi có 2 phòng loại 4 là phòng bốn có giá 230 Euro mỗi đêm.

Để triển khai hệ thống này, bạn sẽ phải viết một server **HotelServer**, một client **HotelClient**, một giao diện remote **RoomManager**, và hiện thực giao diện remote **RoomManagerImpl**.

Client hotel **HotelClient** có thể chạy bằng cách nhập các lệnh sau:

* **java HotelClient host port [help]** - Nếu không có tùy chọn nào được cung cấp, **HotelClient** chỉ in và hiển thị cách chương trình này có thể được sử dụng.
* **java HotelClient host port -login <tên người dùng> <mật khẩu>** - Một quản lý hoặc nhân viên lễ tân đăng nhập vào hệ thống bằng tài khoản người dùng được cung cấp. Có hai vai trò trong hệ thống, tức là quản lý, lễ tân, và cả hai đều có thể đặt và đăng ký phòng cho khách. Hệ thống có khả năng theo dõi lịch sử đăng nhập của người dùng.
* java HotelClient host port -list: Liệt kê số lượng phòng có sẵn trong mỗi mức giá. Kết quả đầu ra phải trông như sau:
  + v phòng loại 0 có giá 55 Euro mỗi đêm
  + w phòng loại 1 có giá 75 Euro mỗi đêm
  + x phòng loại 2 có giá 80 Euro mỗi đêm
  + y phòng loại 3 có giá 150 Euro mỗi đêm
  + z phòng loại 4 có giá 230 Euro mỗi đêm
* java HotelClient host port -book <room\_type> <guest\_name> [guest\_ssn]: Đặt phòng của loại được chỉ định (nếu có sẵn) và đăng ký tên của khách cùng với thông tin của họ. Hệ thống sẽ tự động gán một số nhận dạng (guest\_ssn) cho khách nếu người quản lý/lễ tân không nhập.
* java HotelClient host port -guests: Liệt kê thông tin của tất cả các khách đã đăng ký trong hệ thống, cùng với phòng của họ. Chức năng này chỉ dành cho tài khoản quản lý.
* java HotelClient host port -logOut: Đăng xuất khỏi phiên người dùng hiện tại.

Đối với các tùy chọn logIn, list, book, và guests, HotelClient sẽ gọi từ xa các phương thức liên quan trong một đối tượng từ xa. HotelClient nhận kết quả của những phương thức từ xa này từ HotelServer và in chúng ra màn hình chuẩn. Ngoài ra, chương trình cần lưu trữ dữ liệu một cách lâu dài.

Trước khi triển khai chương trình này, bạn nên phân tích hệ thống, đặc biệt là các hàm và lớp.