

**BỘ ĐỀ ÔN THI LẬP TRÌNH MẠNG – PHẦN RMI**

**MỤC LỤC**

ĐỀ SỐ 1 [B21DCCN319] .....	2
BÀI 1. [Data] BỘ BA SỐ PYTAGO.....	2
BÀI 2. [Character] XỬ LÝ XÂU KÝ TỰ .....	2
BÀI 3. [Byte] PHẦN TỬ XUẤT HIỆN NHIỀU LẦN NHẤT.....	3
BÀI 4. [Object] SẢN PHẨM - 1 .....	4
ĐỀ SỐ 2 [B21DCCN564] .....	5
BÀI 1. [Data] TỔ HỢP TIẾP THEO.....	5
BÀI 2. [Character] ĐẾM SỐ LẦN XUẤT HIỆN - 1 .....	5
BÀI 3 [Byte]. PHẦN TỬ XUẤT HIỆN ÍT LẦN NHẤT.....	6
BÀI 4. [Object] SẢN PHẨM – 2 .....	6
ĐỀ SỐ 3 [B21DCCN008] .....	8
BÀI 1. [Data] PHẦN TỬ LỚN THỨ 3.....	8
BÀI 2 [Character] PHÉP XOR - 1 .....	8
BÀI 3. [Byte] CHUYỂN ĐỔI HỆ BÁT PHÂN.....	9
BÀI 4. [Object] BÀI TOÁN QUẢN LÝ THƯ VIỆN - 1.....	9
ĐỀ SỐ 4 [B21DCCN012] .....	11
BÀI 1. [Data] ĐỔI TIỀN.....	11
BÀI 2. [Character] ĐẾM SỐ LẦN XUẤT HIỆN - 2 .....	11
BÀI 3 [Byte]. PHÉP XOR – 2.....	12
BÀI 4. [Object] BÀI TOÁN QUẢN LÝ THƯ VIỆN - 2.....	13
ĐỀ SỐ 5 [B21DCCN015] .....	14
BÀI 1. [Data] LIỆT KÊ SỐ NGUYÊN TỐ .....	14
BÀI 2 [Character] CHUYỂN ĐỔI LA MÃ – THẬP PHÂN .....	14
BÀI 3. [Byte] GIẢI MÃ.....	15
BÀI 4. [Object] BÀI TOÁN QUẢN LÝ THƯ VIỆN - 2.....	15
ĐỀ SỐ 6 [B21DCCN016] .....	17
BÀI 1. [Data] PHƯƠNG SAI – ĐỘ LỆCH CHUẨN.....	17
ĐỀ SỐ 7 [B21DCCN021] .....	17
BÀI 1. [Data] LIỆT KÊ SỐ ĐỐI XỨNG.....	17
BÀI 2. [Character] MÃ HOÁ VIGENERE.....	18
BÀI 4. [Object] BÀI TOÁN QUẢN LÝ HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ.....	18
ĐỀ SỐ 8 [B21DCCN022] .....	19

BÀI 1. [Data] PHÂN TÍCH RA THỪA SỐ NGUYÊN TỐ.....	19
BÀI 2. [Character] MÃ HOÁ CAESAR - 1 .....	20
ĐỀ SỐ 9 [B21DCCN023] .....	20
BÀI 4. [Object] BÀI TOÁN QUẢN LÝ SINH VIÊN .....	20
ĐỀ SỐ 10 [B21DCCN026] .....	21
BÀI 1. [Data] LIỆT KÊ SỐ CHÍNH PHƯƠNG .....	21
BÀI 3. [Byte] MÃ HOÁ CAESAR - 2 .....	22
ĐỀ SỐ 11 [B21DCCN029] .....	23
BÀI 1. [Byte] SẮP XẾP CHẴN LẺ.....	23
BÀI 2. [Character] MÃ HOÁ Base64.....	23
BÀI 4. [Object] TỔ CHỨC SỰ KIỆN .....	24
ĐỀ SỐ 12 [B21DCCN032] .....	25
BÀI 2. [Character] CHUYỂN ĐỔI THẬP PHÂN – LA MÃ .....	25
BÀI 3. [Byte] MÃ HOÁ HEXA .....	26
ĐỀ SỐ 13 [B21DCCN036] .....	26
BÀI 3. [Byte] NÉN RLE .....	26
ĐỀ SỐ 14 [B21DCCN038] .....	27
BÀI 4. [Object] BÀI TOÁN QUẢN LÝ NHÂN VIÊN.....	27
ĐỀ SỐ 15[B21DCCN040] .....	28
BÀI 3. [Byte] SẮP XẾP CHẴN LẺ.....	28
ĐỀ SỐ 16 [B21DCCN048] .....	28
BÀI 4. [Object] QUẢN LÝ VÉ SỰ KIỆN.....	28
ĐỀ SỐ 17 [B21DCCN053] .....	29
BÀI 1. [Data] SINH TỔ HỢP .....	29
BÀI 2. [Character] MÃ HOÁ URL.....	30
ĐỀ SỐ 18 [B21DCCN063] .....	<b>Error! Bookmark not defined.</b>
BÀI 3. [Byte] PHƯƠNG SAI VÀ ĐỘ LỆCH CHUẨN .....	<b>Error! Bookmark not defined.</b>
ĐỀ SỐ 19 [B21DCCN066] .....	30
BÀI 3. [Byte] PHẦN TỬ LỚN THỨ K.....	30

**BÀI 1. [Data] BỘ BA SỐ PYTAGO**

[Mã câu hỏi (qCode): NMATI6Zw]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu.

**Giao diện từ xa:**

```
public interface DataService extends Remote {  
    public Object requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, Object data) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface DataService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa DataService được đăng ký với RegistryServer với tên là: RMIDataService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhận được từ RMI Server:

- Triệu gọi phương thức requestData để nhận một số nguyên dương N từ server, đại diện cho giới hạn trên của khoảng cần kiểm tra.
- Xác định tất cả các bộ ba số nguyên (a, b, c) sao cho  $a^2 + b^2 = c^2$  và  $a < b < c \leq N$ . Kết quả trả về là danh sách các bộ ba số Pythagore thỏa mãn yêu cầu.  
Ví dụ: Với  $N = 20$ , kết quả là [(3, 4, 5), (5, 12, 13), (8, 15, 17)].
- Triệu gọi phương thức submitData để gửi đối tượng List<List<Integer>> danh sách các bộ ba số Pytago đã tìm được trở lại server.
- Kết thúc chương trình client.

**BÀI 2. [Character] XỬ LÝ XÂU KÝ TỰ**

[Mã câu hỏi (qCode): NFIdNPp6]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi.

**Giao diện từ xa:**

```
public interface CharacterService extends Remote {  
    public String requestCharacter(String studentCode, String qCode) throws  
        RemoteException;  
    public void submitCharacter(String studentCode, String qCode, String strSubmit) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface CharacterService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa CharacterService được đăng ký với RegistryServer với tên là: RMISCharacterService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:

a. Triệu gọi phương thức `requestCharacter` để nhận chuỗi JSON ngẫu nhiên từ server với định dạng: "Chuỗi JSON đầu vào".

b. Phân tích cú pháp chuỗi JSON nhận được và trích xuất các cặp key: value dựa trên vị trí của chúng:

- Các cặp key: value ở vị trí chẵn sẽ được đưa vào chuỗi đầu tiên.
- Các cặp key: value ở vị trí lẻ sẽ được đưa vào chuỗi thứ hai.
- Hai chuỗi kết quả sẽ được nối với nhau và phân tách bởi dấu ;

Ví dụ: Chuỗi JSON ban đầu `{"name": "Alice", "age": 25, "city": "Wonderland", "country": "Fictionland"}` → Kết quả trích xuất: `"name: Alice, city: Wonderland; age: 25, country: Fictionland"`.

c. Triệu gọi phương thức `submitCharacter` để gửi chuỗi kết quả trích xuất đã được định dạng trở lại server.

d. Kết thúc chương trình client.

### BÀI 3. [Byte] PHẦN TỬ XUẤT HIỆN NHIỀU LẦN NHẤT

[Mã câu hỏi (qCode): 1mPMIkGJ]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

**Giao diện từ xa:**

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface `ByteService` được viết trong package `RMI`.

Đối tượng cài đặt giao diện từ xa `ByteService` được đăng ký với `RegistryServer` với tên là: `RMIByteService`.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân nhận được từ RMI Server:

a. Triệu gọi phương thức `requestData` để nhận một mảng dữ liệu nhị phân (`byte[]`) từ server.

b. Tìm phần tử xuất hiện nhiều nhất trong mảng `byte[]`. Nếu có nhiều phần tử có cùng số lần xuất hiện cao nhất, chỉ cần trả về phần tử đầu tiên xuất hiện trong các phần tử đó.

Ví dụ: Nếu mảng dữ liệu nhận được là `[1, 2, 3, 2, 1, 2]`, phần tử xuất hiện nhiều nhất là 2, với tần suất xuất hiện 3 lần.

c. Triệu gọi phương thức `submitData` để gửi mảng `byte` chứa phần tử phổ biến nhất, cùng với tần suất xuất hiện của nó trở lại server.

d. Kết thúc chương trình client

## **BÀI 4. [Object] SẢN PHẨM - 1**

[Mã câu hỏi (qCode): dx3nt4Ij]. Một chương trình (tạm gọi là RMI server) cung cấp giao diện cho phép triệu gọi từ xa với thông tin như sau:

### **1. Giao diện từ xa**

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qAlias) throws  
RemoteException;  
    public void submitObject(String studentCode, String qAlias, Serializable object) throws  
RemoteException;  
}
```

- **Lớp Product** gồm các thông tin: id int, code String, importPrice float, exportPrice float.

Trường dữ liệu: private static final long serialVersionUID = 20151107L;

### **02 hàm khởi dựng**

```
public Product()  
public Product(int id, String code, float ImportPrice, float ExportPrice)
```

### **Trong đó:**

- interface ObjectService và lớp Product được viết trong package RMI
- Đối tượng cài đặt giao diện từ xa ObjectService được đăng ký với RegistryServer: RMIOBJECTSERVICE

Yêu cầu yêu cầu viết chương trình tại máy trạm (RMI client) thực hiện chuẩn hóa sản phẩm theo thứ tự:

- Triệu gọi phương thức requestObject để lấy về đối tượng sản phẩm cần chuẩn hóa.
- Thực hiện chuẩn hóa đối tượng nhận được theo nguyên tắc:
  - Chuyển mã sản phẩm thành in hoa.
  - Cập nhật giá xuất (exportPrice) bằng giá nhập (importPrice) + 20%
- Triệu gọi phương thức submitObject để gửi dữ liệu đã chuẩn hóa
- Kết thúc chương trình client

**HẾT**

## ĐỀ SỐ 2 [B21DCCN564]

### BÀI 1. [Data] TỔ HỢP TIẾP THEO

[Mã câu hỏi (qCode): 3Qe0Qb5w]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu.

**Giao diện từ xa:**

```
public interface DataService extends Remote {  
    public Object requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, Object data) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface DataService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa DataService được đăng ký với RegistryServer với tên là: RMIDataService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhận được từ RMI Server:

- a. Triệu gọi phương thức requestData để nhận một chuỗi các số nguyên.
- b. Sử dụng thuật toán sinh tổ hợp kế tiếp để tìm tổ hợp kế tiếp của chuỗi số này theo thứ tự từ điển. Nếu chuỗi đã là tổ hợp lớn nhất, trả về tổ hợp nhỏ nhất (sắp xếp lại từ đầu theo thứ tự từ điển). Ví dụ: Với chuỗi 1, 2, 3 tổ hợp kế tiếp là 1, 3, 2. Nếu chuỗi là 3, 2, 1 (tổ hợp lớn nhất), kết quả sẽ là 1, 2, 3 (tổ hợp nhỏ nhất).
- c. Triệu gọi phương thức submitData để gửi chuỗi chứa tổ hợp kế tiếp đã tìm được trở lại server.
- d. Kết thúc chương trình client.

### BÀI 2. [Character] ĐẾM SỐ LẦN XUẤT HIỆN - 1

[Mã câu hỏi (qCode): U5BEBBOW]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi.

**Giao diện từ xa:**

```
public interface CharacterService extends Remote {  
    public String requestCharacter(String studentCode, String qCode) throws  
        RemoteException;  
    public void submitCharacter(String studentCode, String qCode, String strSubmit) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface CharacterService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa CharacterService được đăng ký với RegistryServer với tên là: RMISCharacterService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:

## Luyện tập RMI

---

- Triệu gọi phương thức `requestCharacter` để nhận chuỗi ngẫu nhiên từ server với định dạng: "Chuỗi đầu vào".
- Thực hiện đếm tần số xuất hiện của mỗi ký tự trong chuỗi đầu vào và tạo ra chuỗi kết quả theo định dạng <Ký tự><Số lần xuất hiện>, sắp xếp theo thứ tự xuất hiện của các ký tự trong chuỗi.  
Ví dụ: Chuỗi đầu vào "AAABBC" → Kết quả: "A3B2C1".
- Triệu gọi phương thức `submitCharacter` để gửi chuỗi kết quả trở lại server.
- Kết thúc chương trình client.

### BÀI 3 [Byte]. PHẦN TỬ XUẤT HIỆN ÍT LẦN NHẤT

Mã câu hỏi (qCode): L7A2NPQU]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

#### Giao diện từ xa:

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws  
        RemoteException;  
}
```

#### Trong đó:

- Interface `ByteService` được viết trong package `RMI`.
- Đối tượng cài đặt giao diện từ xa `ByteService` được đăng ký với `RegistryServer` với tên là: `RMIByteService`.
- Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân nhận được từ RMI Server:
- Triệu gọi phương thức `requestData` để nhận một mảng dữ liệu nhị phân (`byte[]`) từ server.
  - Tìm phần tử có số lần xuất hiện ít nhất trong mảng, nếu có nhiều phần tử có cùng số lần xuất hiện ít nhất, chỉ cần trả về phần tử đầu tiên xuất hiện trong các phần tử đó  
Ví dụ: Nếu mảng dữ liệu nhận được là [1, 2, 3, 2, 1], chương trình sẽ tìm ra phần tử xuất hiện ít nhất là 3.
  - Triệu gọi phương thức `submitData` để gửi mảng `byte` kết quả chứa phần tử có số lần xuất hiện ít nhất và số lần xuất hiện trở lại server.
  - Kết thúc chương trình client.

### BÀI 4. [Object] SẢN PHẨM – 2

[Mã câu hỏi (qCode): PY43T66m]. Một chương trình (tạm gọi là RMI Server) cung cấp các mã khuyến mãi sản phẩm ngẫu nhiên cho sinh viên, được mô tả như sau:

#### • Giao diện từ xa

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qAlias) throws  
        RemoteException;  
    public void submitObject(String studentCode, String qAlias, Serializable object) throws  
        RemoteException;
```

}

\* **Lớp ProductX** gồm các thuộc tính id String, code String, discountCode String, discount int.

- Một hàm khởi dựng với đầy đủ các thuộc tính liệt kê ở trên

- Trường dữ liệu: private static final long serialVersionUID = 20171107;

\* Đối tượng triệu gọi từ xa được đăng ký RegistryServer với tên: RMIOBJECTSERVICE

\* Tất cả các lớp được viết trong package RMI

**Yêu cầu** là xây dựng một chương trình client thực hiện các tương tác với hệ thống phần mềm ở trên theo kịch bản dưới đây:

1. Triệu gọi phương thức từ xa requestObject từ RMI Server với tham số đầu vào là mã sinh viên, mã câu để nhận về đối tượng ProductX
2. Nhận về đối tượng ProductX từ RMI Server với giá trị ban đầu đã được thiết lập. Tính tổng các chữ số nằm trong chuỗi mã giảm giá (discountCode) để ra giá trị được khuyến mãi của sản phẩm và cập nhật giá trị của khuyến mãi (discount)
3. Triệu gọi phương thức từ xa submitObject với tham số đầu vào là đối tượng Product đã được cập nhật đầy đủ thông tin giá trị khuyến mãi
4. Kết thúc chương trình

**HẾT**



**BÀI 1. [Data] PHẦN TỬ LỚN THỨ 3**

[Mã câu hỏi (qCode): wzG6G9ji]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu.

Giao diện từ xa:

```
public interface DataService extends Remote {  
    public Object requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, Object data) throws  
        RemoteException;  
}
```

Trong đó:

- Interface DataService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa DataService được đăng ký với RegistryServer với tên là: RMIDataService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhận được từ RMI Server:

- Triệu gọi phương thức requestData để nhận một chuỗi chứa mảng số nguyên và một số nguyên K từ server với định dạng: “mảng; số nguyên K”.
- Sử dụng thuật toán phù hợp để tìm phần tử lớn thứ K trong mảng số nguyên đã cho.  
Ví dụ: Với Chuỗi “3, 1, 5, 12, 2, 11; 3” nghĩa là mảng [3, 1, 5, 12, 2, 11] và K = 3 → cần tìm phần tử lớn thứ 3 là 5.
- Triệu gọi phương thức submitData để gửi kết quả phần tử lớn thứ K đã tìm được trở lại server.
- Kết thúc chương trình client.

**BÀI 2 [Character] PHÉP XOR - 1**

[Mã câu hỏi (qCode): Xidb0Thk]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi.

**Giao diện từ xa:**

```
public interface CharacterService extends Remote {  
    public String requestCharacter(String studentCode, String qCode) throws RemoteException;  
    public void submitCharacter(String studentCode, String qCode, String strSubmit) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface CharacterService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa CharacterService được đăng ký với RegistryServer với tên là: RMISCharacterService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:

- Triệu gọi phương thức requestCharacter để nhận chuỗi ngẫu nhiên từ server với định dạng: "Khóa XOR;Chuỗi đầu vào".

b. Thực hiện thao tác mã hóa XOR cho chuỗi đầu vào với khóa XOR nhận được từ server. Mã hóa XOR thực hiện bằng cách áp dụng phép XOR trên từng ký tự trong chuỗi đầu vào và ký tự tương ứng trong khóa (khóa được lặp lại để khớp độ dài chuỗi đầu vào).

Ví dụ: Chuỗi ban đầu "A;HELLO" -> Khóa "A" → chuỗi mã hóa là: "IFMMN"

c. Triệu gọi phương thức submitCharacter để gửi chuỗi đã được mã hóa trở lại server.

d. Kết thúc chương trình client.

### **BÀI 3. [Byte] CHUYỂN ĐỔI HỆ BÁT PHÂN**

[Mã câu hỏi (qCode): HhPAxeDw]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

**Giao diện từ xa:**

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface ByteService được viết trong package RMI.

Đối tượng cài đặt giao diện từ xa ByteService được đăng ký với RegistryServer với tên là: RMIByteService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân nhận được từ RMI Server:

- a. Triệu gọi phương thức requestData để nhận một mảng dữ liệu nhị phân (byte[]) từ server.
- b. Chuyển đổi mỗi byte trong mảng dữ liệu nhận được thành chuỗi biểu diễn số bát phân (octal). Mỗi byte sẽ được chuyển thành một chuỗi số bát phân gồm ba chữ số (nếu cần, thêm các chữ số 0 ở đầu để đảm bảo đúng định dạng).

Ví dụ: Nếu mảng dữ liệu nhận được là [72, 101, 108], chương trình sẽ thực hiện chuyển đổi từng byte thành dạng bát phân:

- 72 → "110"
- 101 → "145"
- 108 → "154"

Kết quả là chuỗi octal "110145154". Chuyển chuỗi bát phân này trở lại thành mảng byte[] (mỗi ký tự trong chuỗi octal được mã hóa dưới dạng ASCII byte) để gửi về server.

- c. Triệu gọi phương thức submitData để gửi mảng byte[] chứa chuỗi bát phân trở lại server.
- d. Kết thúc chương trình client.

### **BÀI 4. [Object] BÀI TOÁN QUẢN LÝ THƯ VIỆN - 1**

[Mã câu hỏi (qCode): nCOdwcKJ]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để gán mã code cho sách trong một hệ thống quản lý thư viện. Chương trình sẽ ngẫu nhiên tạo ra đối tượng Book với các giá trị ban đầu và cung cấp cho RMI client như sau:

**Giao diện từ xa:**

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qCode) throws  
    RemoteException;  
    public void submitObject(String studentCode, String qCode, Serializable object) throws  
    RemoteException;  
}
```

**Lớp Book gồm các thuộc tính:** id String, title String, author String, yearPublished int, pageCount int, code String.

Trường dữ liệu: private static final long serialVersionUID = 20241123L;

**02 hàm khởi dựng:**

```
public Book()  
public Book(int id, String title, String author, int yearPublished, int pageCount)
```

**Trong đó:**

Interface ObjectService và lớp Book được viết trong package RMI.

Đối tượng cài đặt giao diện từ xa ObjectService được đăng ký với RegistryServer: RMIObjectService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với đối tượng sách được nhận từ RMI Server:

- a. Triệu gọi phương thức requestObject để nhận đối tượng Book ngẫu nhiên từ server.
- b. Tạo mã code cho sách dựa trên các thành phần sau:
  - Lấy chữ cái đầu của họ và tên đầu của tác giả. Ví dụ, với tác giả "John Doe", hai chữ cái đầu sẽ là "JD".
  - Lấy hai chữ số cuối cùng của năm xuất bản (yearPublished). Ví dụ, nếu yearPublished là 1985, sẽ lấy "85".
  - Lấy tổng số chữ cái trong title của sách.
  - Lấy số chữ số của pageCount (số trang) và nếu số chữ số < 3, thêm các số 0 ở đầu để đạt đủ ba chữ số.

Kết hợp tất cả các thành phần trên để tạo ra mã code theo định dạng: "[Chữ cái đầu tên tác giả][Hai chữ số cuối của năm xuất bản][Tổng số chữ cái trong title][Số chữ số của pageCount]". Ví dụ, nếu tác giả là "Jane Austen", yearPublished là 1813, title là "Pride and Prejudice" với 18 ký tự, và pageCount là 345, thì mã code sẽ là: "JA1318345".

- c. Cập nhật giá trị code trong đối tượng Book.
- d. Triệu gọi phương thức submitObject để gửi đối tượng Book đã được xử lý trở lại server.
- e. Kết thúc chương trình client.

**HẾT**

**BÀI 1. [Data] ĐỔI TIỀN**

[Mã câu hỏi (qCode): Iz06p8Zw]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu.

**Giao diện từ xa:**

```
public interface DataService extends Remote {  
    public Object requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, Object data) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface DataService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa DataService được đăng ký với RegistryServer với tên là: RMIDataService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhận được từ RMI Server:

- Triệu gọi phương thức requestData để nhận một số nguyên dương amount từ server, đại diện cho số tiền cần đạt được.
- Sử dụng thuật toán xếp đồng xu với các mệnh giá cố định [1, 2, 5, 10] để xác định số lượng đồng xu tối thiểu cần thiết để đạt được số tiền amount. Nếu không thể đạt được số tiền đó với các mệnh giá hiện có, trả về -1.  
Ví dụ: Với amount = 18 và mệnh giá đồng xu cố định [1, 2, 5, 10], kết quả là 4 ( $18 = 10 + 5 + 2 + 1$ ). Chuỗi cần gửi lên server là: 4; 10,5,2,1
- Triệu gọi phương thức submitData để gửi chuỗi chứa kết quả số lượng đồng xu tối thiểu và giá trị các đồng xu tương ứng trở lại server.
- Kết thúc chương trình client.

**BÀI 2. [Character] ĐẾM SỐ LẦN XUẤT HIỆN - 2**

[Mã câu hỏi (qCode): ctRflejL]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi.

**Giao diện từ xa:**

```
public interface CharacterService extends Remote {  
    public String requestCharacter(String studentCode, String qCode) throws  
        RemoteException;  
    public void submitCharacter(String studentCode, String qCode, String strSubmit) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface CharacterService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa CharacterService được đăng ký với RegistryServer với tên là: RMISCharacterService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:

- Triệu gọi phương thức requestCharacter để nhận chuỗi ngẫu nhiên từ server với định dạng: "Chuỗi văn bản đầu vào".
- Thực hiện thao tác đếm tần số xuất hiện của từng ký tự trong chuỗi đầu vào. Kết quả trả về là danh sách các ký tự kèm theo số lần xuất hiện của mỗi ký tự.  
Ví dụ: Chuỗi ban đầu "Hello world" -> Kết quả đếm tần số ký tự: {"H": 1, "e": 1, "l": 3, "o": 2, " ": 1, "w": 1, "r": 1, "d": 1}.
- Triệu gọi phương thức submitCharacter để gửi kết quả đếm tần số ký tự trở lại server dưới dạng chuỗi kết quả đã được định dạng.
- Kết thúc chương trình client.

### BÀI 3 [Byte]. PHÉP XOR – 2

[Mã câu hỏi (qCode): 4BraNTI5]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

**Giao diện từ xa:**

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws  
        RemoteException;  
}
```

**Trong đó:**

- Interface ByteService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa ByteService được đăng ký với RegistryServer với tên là: RMIByteService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân nhận được từ RMI Server:

- Triệu gọi phương thức requestData để nhận một mảng dữ liệu nhị phân (byte[]) từ server, đại diện cho một chuỗi văn bản ASCII.
- Thực hiện mã hóa XOR cho mảng dữ liệu nhị phân bằng cách sử dụng một khóa là chuỗi "PTIT". Thực hiện phép XOR trên từng byte trong mảng dữ liệu với byte tương ứng trong khóa (khóa sẽ được lặp lại để khớp với độ dài của mảng dữ liệu).  
Ví dụ: Nếu dữ liệu nhị phân nhận được là [72, 101, 108, 108, 111] (tương ứng với chuỗi "Hello"), và khóa là "PTIT", chương trình sẽ thực hiện mã hóa như sau:
  - Chuyển khóa "PTIT" thành mảng byte [80, 84, 73, 84].
  - Lặp lại khóa để khớp độ dài dữ liệu: [80, 84, 73, 84, 80].
  - Thực hiện phép XOR trên từng byte của dữ liệu với khóa:
  - Kết quả mã hóa là mảng [24, 49, 37, 56, 63].
- Triệu gọi phương thức submitData để gửi mảng dữ liệu đã được mã hóa bằng XOR trở lại server.
- Kết thúc chương trình client.

#### **BÀI 4. [Object] BÀI TOÁN QUẢN LÝ THƯ VIỆN - 2**

[Mã câu hỏi (qCode): CoWosBho]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý mã hóa dữ liệu tác phẩm trong hệ thống quản lý thư viện. Chương trình sẽ ngẫu nhiên tạo ra đối tượng BookX với các giá trị ban đầu và cung cấp cho RMI client như sau:

##### **Giao diện từ xa:**

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qCode) throws RemoteException;  
    public void submitObject(String studentCode, String qCode, Serializable object) throws  
    RemoteException;  
}
```

Lớp BookX gồm các thuộc tính: id String, title String, author String, yearPublished int, genre String, code String.

**Trường dữ liệu:** private static final long serialVersionUID = 20241124L;

##### **02 hàm khởi dựng:**

```
public BookX()  
public BookX(int id, String title, String author, int yearPublished, String genre)
```

##### **Trong đó:**

Interface ObjectService và lớp BookX được viết trong package RMI.

Đối tượng cài đặt giao diện từ xa ObjectService được đăng ký với RegistryServer: RMIOBJECTService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với đối tượng sách được nhận từ RMI Server:

a. Triệu gọi phương thức requestObject để nhận đối tượng BookX ngẫu nhiên từ server.

b. Tạo mã code cho sách dựa trên các quy tắc mã hóa sau:

- Lấy chữ cái đầu tiên và cuối cùng trong tên tác giả (author).
- Lấy hai chữ số cuối cùng của yearPublished.
- Số lượng chữ cái trong genre của sách.
- Độ dài của title chia lấy dư cho 10 (ví dụ: với tiêu đề dài 32 ký tự, giá trị này sẽ là 2).

Kết hợp tất cả các thành phần trên để tạo ra mã code theo định dạng: [Chữ cái đầu và cuối tên tác giả][Hai chữ số cuối của năm xuất bản][Số chữ cái của genre][Độ dài title modulo 10].

Ví dụ, nếu tác giả là "Mark Twain", năm xuất bản là 1884, thể loại là "Fiction" với 7 ký tự, và tiêu đề có 24 ký tự, mã code sẽ là: "MT8474".

c. Cập nhật giá trị code trong đối tượng BookX.

d. Triệu gọi phương thức submitObject để gửi đối tượng BookX đã được xử lý trở lại server.

e. Kết thúc chương trình client.

**HẾT**

## ĐỀ SỐ 5 [B21DCCN015]

### BÀI 1. [Data] LIỆT KÊ SỐ NGUYÊN TỐ

[Mã câu hỏi (qCode): nD8MdtME]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu.

#### Giao diện từ xa:

```
public interface DataService extends Remote {  
    public Object requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, Object data) throws  
        RemoteException;  
}
```

#### Trong đó:

- Interface DataService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa DataService được đăng ký với RegistryServer với tên là: RMIDataService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhận được từ RMI Server:

- a. Triệu gọi phương thức requestData để nhận một số nguyên dương N từ server, đại diện cho giới hạn trên của khoảng cần kiểm tra.
- b. Thực hiện tìm tất cả các số nguyên tố trong khoảng từ 1 đến N. Ví dụ: Với N = 10, kết quả là danh sách các số nguyên tố “2, 3, 5, 7”.
- c. Triệu gọi phương thức submitData để gửi List< Integer> danh sách các số nguyên tố đã tìm được trở lại server.
- d. Kết thúc chương trình client.

### BÀI 2 [Character] CHUYỂN ĐỔI LA MÃ – THẬP PHÂN

[Mã câu hỏi (qCode): 0JaasIw6]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi.

#### Giao diện từ xa:

```
public interface CharacterService extends Remote {  
    public String requestCharacter(String studentCode, String qCode) throws  
        RemoteException;  
    public void submitCharacter(String studentCode, String qCode, String strSubmit) throws  
        RemoteException;  
}
```

#### Trong đó:

- Interface CharacterService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa CharacterService được đăng ký với RegistryServer với tên là: RMISCharacterService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:



## Luyện tập RMI

---

- Triệu gọi phương thức `requestCharacter` để nhận chuỗi ngẫu nhiên từ server với định dạng: "Chuỗi số La Mã đầu vào".
- Thực hiện chuyển đổi chuỗi số La Mã nhận được thành số thập phân (Decimal).  
Quy tắc chuyển đổi: Các ký tự La Mã chính bao gồm: I=1, V=5, X=10, L=50, C=100, D=500, M=1000.  
Ví dụ: "MCMXCIV" → 1994.
- Triệu gọi phương thức `submitCharacter` để gửi số thập phân đã chuyển đổi trở lại server.
- Kết thúc chương trình client.

### BÀI 3. [Byte] GIẢI MÃ

[Mã câu hỏi (qCode): oeJljCIf]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

Giao diện từ xa:

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws RemoteException;  
}
```

Trong đó:

- Interface `ByteService` được viết trong package `RMI`.
- Đối tượng cài đặt giao diện từ xa `ByteService` được đăng ký với `RegistryServer` với tên là: `RMIByteService`.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân đã mã hóa dưới dạng Base64 nhận được từ RMI Server:

- Triệu gọi phương thức `requestData` để nhận một mảng dữ liệu nhị phân (`byte[]`) từ server, trong đó mảng này chứa một chuỗi Base64 được mã hóa.
- Thực hiện giải mã dữ liệu nhận được từ định dạng Base64 về dữ liệu gốc. Sử dụng giải mã Base64 để chuyển mảng `byte` Base64 thành mảng `byte` nguyên bản ban đầu.  
Ví dụ: Nếu dữ liệu Base64 nhận được là chuỗi mã hóa `SGVsbG8gV29ybGQ=` (tương ứng với chuỗi "Hello World" trong ASCII), chương trình sẽ thực hiện giải mã để nhận được mảng dữ liệu nhị phân `[72, 101, 108, 108, 111, 32, 87, 111, 114, 108, 100]`.
- Triệu gọi phương thức `submitData` để gửi mảng dữ liệu gốc đã giải mã trở lại server.
- Kết thúc chương trình client.

### BÀI 4. [Object] BÀI TOÁN QUẢN LÝ THƯ VIỆN - 2

[Mã câu hỏi (qCode): y7HB10MN]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý mã hóa dữ liệu tác phẩm trong hệ thống quản lý thư viện. Chương trình sẽ ngẫu nhiên tạo ra đối tượng `BookX` với các giá trị ban đầu và cung cấp cho RMI client như sau:

Giao diện từ xa:

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qCode) throws RemoteException;
```



## Luyện tập RMI

---

```
public void submitObject(String studentCode, String qCode, Serializable object) throws  
RemoteException;  
}
```

Lớp BookX gồm các thuộc tính: id String, title String, author String, yearPublished int, genre String, code String.

Trường dữ liệu: private static final long serialVersionUID = 20241124L;

02 hàm khởi dựng:

```
public BookX()  
public BookX(int id, String title, String author, int yearPublished, String genre)
```

Trong đó:

Interface ObjectService và lớp BookX được viết trong package RMI.

Đối tượng cài đặt giao diện từ xa ObjectService được đăng ký với RegistryServer: RMIObjectService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với đối tượng sách được nhận từ RMI Server:

- a. Triệu gọi phương thức requestObject để nhận đối tượng BookX ngẫu nhiên từ server.
- b. Tạo mã code cho sách dựa trên các quy tắc mã hóa sau:
  - Lấy chữ cái đầu tiên và cuối cùng trong tên tác giả (author).
  - Lấy hai chữ số cuối cùng của yearPublished.
  - Số lượng chữ cái trong genre của sách.
  - Độ dài của title chia lấy dư cho 10 (ví dụ: với tiêu đề dài 32 ký tự, giá trị này sẽ là 2).

Kết hợp tất cả các thành phần trên để tạo ra mã code theo định dạng: [Chữ cái đầu và cuối tên tác giả][Hai chữ số cuối của năm xuất bản][Số chữ cái của genre][Độ dài title modulo 10].

Ví dụ, nếu tác giả là "Mark Twain", năm xuất bản là 1884, thể loại là "Fiction" với 7 ký tự, và tiêu đề có 24 ký tự, mã code sẽ là "MT8474".

- c. Cập nhật giá trị code trong đối tượng BookX.
- d. Triệu gọi phương thức submitObject để gửi đối tượng BookX đã được xử lý trở lại server.
- e. Kết thúc chương trình client.

**HẾT**

## ĐỀ SỐ 6 [B21DCCN016]

### BÀI 1. [Data] PHƯƠNG SAI – ĐỘ LỆCH CHUẨN

[Mã câu hỏi (qCode): uZMEY3Zg]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu.

#### Giao diện từ xa:

```
public interface DataService extends Remote {  
    public Object requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, Object data) throws RemoteException;  
}
```

#### Trong đó:

- Interface DataService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa DataService được đăng ký với RegistryServer với tên là: RMIDataService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhận được từ RMI Server:

- a. Triệu gọi phương thức requestData để nhận một chuỗi chứa tập hợp các số thực từ server.
- b. Tính toán phương sai (variance) và độ lệch chuẩn (standard deviation) của tập hợp dữ liệu này (làm tròn tới 02 chữ số thập phân)

Ví dụ: Với tập dữ liệu nhận được “2.0, 4.0, 4.0, 4.0, 5.0, 5.0, 7.0, 9.0”, phương sai là 4.00 và độ lệch chuẩn là 2.00

- c. Triệu gọi phương thức submitData để gửi chuỗi chứa kết quả phương sai và độ lệch chuẩn trở lại server dưới dạng một cặp giá trị “variance : standard deviation”
- d. Kết thúc chương trình client.

## ĐỀ SỐ 7 [B21DCCN021]

### BÀI 1. [Data] LIỆT KÊ SỐ ĐỐI XỨNG

[Mã câu hỏi (qCode): t3cBY2uk]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu.

#### Giao diện từ xa:

```
public interface DataService extends Remote {  
    public Object requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, Object data) throws RemoteException;  
}
```

#### Trong đó:

- Interface DataService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa DataService được đăng ký với RegistryServer với tên là: RMIDataService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhận được từ RMI Server:

- a. Triệu gọi phương thức requestData để nhận hai số nguyên dương N và K từ server, đại diện cho khoảng cần kiểm tra ( $N \leq \text{số} < K$ ).

b. Xác định tất cả các số nguyên đối xứng (Palindrome Number) trong khoảng từ N đến K. Kết quả trả về là danh sách các số đối xứng thỏa mãn yêu cầu.

Ví dụ: Với N = 50 và K = 150, kết quả là [55, 66, 77, 88, 99, 101, 111, 121, 131, 141].

c. Triệu gọi phương thức submitData để gửi đối tượng List<Integer> danh sách các số nguyên đối xứng đã tìm được trở lại server.

d. Kết thúc chương trình client.

## **BÀI 2. [Character] MÃ HOÁ VIGENERE**

[Mã câu hỏi (qCode): Y7YMXHs4]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi.

### **Giao diện từ xa:**

```
public interface CharacterService extends Remote {  
    public String requestCharacter(String studentCode, String qCode) throws RemoteException;  
    public void submitCharacter(String studentCode, String qCode, String strSubmit) throws  
    RemoteException;  
}
```

### **Trong đó:**

- Interface CharacterService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa CharacterService được đăng ký với RegistryServer với tên là: RMICCharacterService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:

a. Triệu gọi phương thức requestCharacter để nhận chuỗi ngẫu nhiên từ server với định dạng: "Tür khóa;Chuỗi đầu vào"

b. Thực hiện thao tác mã hóa Vigenère cho chuỗi nhận được. Biết rằng, mã hóa Vigenère thực hiện mã hóa mỗi ký tự trong chuỗi đầu vào được dịch đi một khoảng bằng với vị trí tương ứng của ký tự trong từ khóa. (Từ khóa được lặp lại để khớp với độ dài của chuỗi)

Ví dụ: chuỗi ban đầu "PTIT;HELLO" → từ khóa "PTIT" và chuỗi mã hóa là: "WXTED"

c. Triệu gọi phương thức submitCharacter để gửi chuỗi đã được mã hóa trở lại server.

d. Kết thúc chương trình client.

## **BÀI 4. [Object] BÀI TOÁN QUẢN LÝ HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ**

Mã câu hỏi (qCode): vLJWvWpf]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để quản lý thông tin đơn hàng trong hệ thống thương mại điện tử. Chương trình sẽ ngẫu nhiên tạo ra đối tượng Order với các giá trị ban đầu và cung cấp cho RMI client như sau:

### **Giao diện từ xa:**

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qCode) throws RemoteException;  
    public void submitObject(String studentCode, String qCode, Serializable object) throws  
    RemoteException;
```

}

**Lớp Order** gồm các thuộc tính: id String, customerCode String, orderDate String, shippingType String, orderCode String.

- Trường dữ liệu: private static final long serialVersionUID = 20241132L;
- 02 hàm khởi dựng:  
public Order()  
public Order(int id, String customerCode, String orderDate, String shippingType)

**Trong đó:**

- Interface ObjectService và lớp Order được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa ObjectService được đăng ký với RegistryServer: RMIObjectService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với đối tượng đơn hàng được nhận từ RMI Server:

a. Triệu gọi phương thức requestObject để nhận đối tượng Order ngẫu nhiên từ server.

b. Tạo mã orderCode cho đơn hàng dựa trên các quy tắc sau:

- Bắt đầu bằng hai ký tự đầu của shippingType, viết in hoa.
- Kế đến là ba ký tự cuối của customerCode.
- Cuối cùng là ngày và tháng từ orderDate (theo định dạng "ddMM").

Ví dụ: Nếu đơn hàng có mã khách hàng là "C123456", ngày đặt hàng là "2023-10-05", và loại giao hàng là "Express", thì mã orderCode sẽ là "EX4560510".

c. Cập nhật giá trị orderCode trong đối tượng Order.

d. Triệu gọi phương thức submitObject để gửi đối tượng Order đã được xử lý trở lại server.

e. Kết thúc chương trình client.

## ĐỀ SỐ 8 [B21DCCN022]

### BÀI 1. [Data] PHÂN TÍCH RA THỪA SỐ NGUYÊN TỐ

[Mã câu hỏi (qCode): zzmmquoc]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu.

Giao diện từ xa:

```
public interface DataService extends Remote {  
    public Object requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, Object data) throws RemoteException;  
}
```

**Trong đó:**

- Interface DataService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa DataService được đăng ký với RegistryServer với tên là: RMIDataService.

**Yêu cầu:** Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhận được từ RMI Server:

a. Triệu gọi phương thức requestData để nhận một số nguyên dương lớn từ server, gọi là N.

b. Thực hiện phân rã số N thành các thừa số nguyên tố. Kết quả trả về là danh sách các thừa số nguyên tố của N.

Ví dụ: Với N = 84, kết quả là danh sách “2, 2, 3, 7”.

c. Triệu gọi phương thức submitData để gửi danh sách các thừa số nguyên tố đã tìm được trở lại server.

d. Kết thúc chương trình client.

## **BÀI 2. [Character] MÃ HOÁ CAESAR - 1**

[Mã câu hỏi (qCode): j4rStb2a]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi. Cụ thể:

Giao diện từ xa:

```
public interface CharacterService extends Remote {  
    public String requestCharacter(String studentCode, String qCode) throws RemoteException;  
    public void submitCharacter(String studentCode, String qCode, String strSubmit) throws  
    RemoteException;  
}
```

Trong đó:

- Interface CharacterService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa CharacterService được đăng ký với RegistryServer với tên là: RMICCharacterService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:

- Triệu gọi phương thức requestCharacter để nhận chuỗi ngẫu nhiên từ server.
- Thực hiện thao tác giải mã Caesar (mã hóa chuỗi bằng cách dịch từng ký tự đi một số lượng vị trí nhất định). Biết rằng giá trị dịch đúng bằng kích thước của chuỗi chia lấy dư cho 7.
- Triệu gọi phương thức submitCharacter để gửi chuỗi đã được giải mã trở lại server.
- Kết thúc chương trình client.

## **ĐỀ SỐ 9 [B21DCCN023]**

### **BÀI 4. [Object] BÀI TOÁN QUẢN LÝ SINH VIÊN**

[Mã câu hỏi (qCode): INV6xzmK]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý thông tin sinh viên trong hệ thống quản lý giáo dục. Chương trình sẽ ngẫu nhiên tạo ra đối tượng Student với các giá trị ban đầu và cung cấp cho RMI client như sau:

Giao diện từ xa:

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qCode) throws RemoteException;  
    public void submitObject(String studentCode, String qCode, Serializable object) throws  
    RemoteException;  
}
```

Lớp Student gồm các thuộc tính: id String, name String, enrollmentYear int, code String.

Trường dữ liệu: private static final long serialVersionUID = 20241130L;

02 hàm khởi dựng:

```
public Student()  
public Student(int id, String name, int enrollmentYear)
```

Trong đó:

Interface `ObjectService` và lớp `Student` được viết trong package `RMI`.

Đối tượng cài đặt giao diện từ xa `ObjectService` được đăng ký với `RegistryServer`: `RMIObjectService`.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với đối tượng sinh viên được nhận từ RMI Server:

- Triệu gọi phương thức `requestObject` để nhận đối tượng `Student` ngẫu nhiên từ server.
- Thực hiện

Tạo mã code cho sinh viên dựa trên các quy tắc sau:

- Bắt đầu bằng ký tự "B".
- Kế đến là hai chữ số cuối của `enrollmentYear`.
- Sau đó là TÊN của sinh viên, tất cả các ký tự của tên viết in hoa.
- Kết thúc với các chữ cái đầu tiên của Họ và Họ lót, đều viết in hoa.

Ví dụ: Nếu sinh viên có tên là "Nguyen Van Tuan", năm nhập học là 2022, mã code sẽ là "B22TUAN\_NV".

Chuẩn hóa tên (name) theo quy tắc:

- Chữ cái đầu tiên của Tên, Họ, và Tên lót phải viết hoa, các chữ cái khác viết thường.
- Ví dụ: Nếu name ban đầu là "nguYEn van tAi tUAN", sau khi chuẩn hóa sẽ trở thành "Nguyen Van Tai TUAN".

- Cập nhật giá trị mã (code) và tên (name) trong đối tượng `Student` và
- Triệu gọi phương thức `submitObject` để gửi đối tượng `Student` đã được xử lý trở lại server.
- Kết thúc chương trình client.

### ĐỀ SỐ 10 [B21DCCN026]

#### BÀI 1. [Data] LIỆT KÊ SỐ CHÍNH PHƯƠNG

[Mã câu hỏi (qCode): INV6xzmK]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý thông tin sinh viên trong hệ thống quản lý giáo dục. Chương trình sẽ ngẫu nhiên tạo ra đối tượng `Student` với các giá trị ban đầu và cung cấp cho RMI client như sau:

Giao diện từ xa:

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qCode) throws RemoteException;  
    public void submitObject(String studentCode, String qCode, Serializable object) throws  
    RemoteException;  
}
```

Lớp `Student` gồm các thuộc tính: `id String`, `name String`, `enrollmentYear int`, `code String`.

Trường dữ liệu: `private static final long serialVersionUID = 20241130L;`

02 hàm khởi dựng:

public Student()

public Student(int id, String name, int enrollmentYear)

Trong đó:

Interface ObjectService và lớp Student được viết trong package RMI.

Đối tượng cài đặt giao diện từ xa ObjectService được đăng ký với RegistryServer: RMIObjectService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với đối tượng sinh viên được nhận từ RMI Server:

a. Triệu gọi phương thức requestObject để nhận đối tượng Student ngẫu nhiên từ server.

b. Thực hiện

Tạo mã code cho sinh viên dựa trên các quy tắc sau:

- Bắt đầu bằng ký tự "B".
- Kế đến là hai chữ số cuối của enrollmentYear.
- Sau đó là TÊN của sinh viên, tất cả các ký tự của tên viết in hoa.
- Kết thúc với các chữ cái đầu tiên của Họ và Họ lót, đều viết in hoa.

Ví dụ: Nếu sinh viên có tên là "Nguyen Van Tuan", năm nhập học là 2022, mã code sẽ là "B22TUAN\_NV".

Chuẩn hóa tên (name) theo quy tắc:

- Chữ cái đầu tiên của Tên, Họ, và Tên lót phải viết hoa, các chữ cái khác viết thường.
- Ví dụ: Nếu name ban đầu là "nguYen van tAi tUAN", sau khi chuẩn hóa sẽ trở thành "Nguyen Van Tai TUAN".

c. Cập nhật giá trị mã (code) và tên (name) trong đối tượng Student và

d. Triệu gọi phương thức submitObject để gửi đối tượng Student đã được xử lý trở lại server.

e. Kết thúc chương trình client.

### BÀI 3. [Byte] MÃ HOÁ CAESAR - 2

[Mã câu hỏi (qCode): i0EVI2TB]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

Giao diện từ xa:

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws RemoteException;  
}
```

Trong đó:

- Interface ByteService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa ByteService được đăng ký với RegistryServer với tên là: RMIByteService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân nhận được từ RMI Server:

a. Triệu gọi phương thức requestData để nhận một mảng dữ liệu nhị phân (byte[]) từ server, đại diện cho một chuỗi văn bản ASCII.



b. Thực hiện mã hóa Caesar cho mảng dữ liệu nhị phân bằng cách dịch chuyển mỗi byte trong mảng đi một số bước cố định trong bảng mã ASCII. Số bước dịch chuyển là số ký tự ASCII trong mảng dữ liệu.

Ví dụ: Nếu dữ liệu nhị phân nhận được là [72, 101, 108, 108, 111] (tương ứng với chuỗi "Hello"), chương trình sẽ thực hiện mã hóa Caesar với độ dịch là 5. Kết quả mã hóa là mảng [77, 108, 113, 113, 116], tương ứng với chuỗi "Mlqqt".

c. Triệu gọi phương thức submitData để gửi mảng dữ liệu đã được mã hóa bằng Caesar trở lại server.

d. Kết thúc chương trình client.

## ĐỀ SỐ 11 [B21DCCN029]

### BÀI 1. [Byte] SẮP XẾP CHẴN LẺ

[Mã câu hỏi (qCode): A7hytb1V]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

Giao diện từ xa:

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws RemoteException;  
}
```

Trong đó:

- Interface ByteService được viết trong package RMI.

Đối tượng cài đặt giao diện từ xa ByteService được đăng ký với RegistryServer với tên là: RMIByteService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân nhận được từ RMI Server:

a. Triệu gọi phương thức requestData để nhận một mảng dữ liệu nhị phân (byte[]) từ server.

b. Thực hiện phân chia mảng byte[] nhận được thành hai phần: phần đầu chứa các byte có giá trị chẵn và phần sau chứa các byte có giá trị lẻ, duy trì thứ tự xuất hiện của các phần tử trong từng nhóm.

Ví dụ: Nếu mảng dữ liệu nhận được là [1, 2, 3, 4, 5], sau khi phân chia chẵn-lẻ, mảng kết quả sẽ là [2, 4, 1, 3, 5] (tất cả phần tử chẵn được đặt trước, theo sau là tất cả phần tử lẻ).

c. Triệu gọi phương thức submitData để gửi mảng byte[] đã được phân chia chẵn-lẻ trở lại server.

d. Kết thúc chương trình client.

### BÀI 2. [Character] MÃ HOÁ Base64

[Mã câu hỏi (qCode): psd4Jmnt]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi.

Giao diện từ xa:

```
public interface CharacterService extends Remote {
```



## Luyện tập RMI

---

```
public String requestCharacter(String studentCode, String qCode) throws RemoteException;
public void submitCharacter(String studentCode, String qCode, String strSubmit) throws
RemoteException;
}
```

Trong đó:

- Interface CharacterService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa CharacterService được đăng ký với RegistryServer với tên là: RMICCharacterService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:

- a. Triệu gọi phương thức requestCharacter để nhận chuỗi ngẫu nhiên từ server với định dạng: "Chuỗi đầu vào".
- b. Thực hiện thao tác mã hóa Base64 cho chuỗi đầu vào nhận được từ server. Mã hóa Base64 chuyển đổi chuỗi nhị phân thành định dạng văn bản ASCII bằng cách mã hóa mỗi nhóm 6 bit của chuỗi thành một ký tự.  
Ví dụ: Chuỗi ban đầu "HELLO" -> Chuỗi mã hóa Base64 là: "SEVMTE8="
- c. Triệu gọi phương thức submitCharacter để gửi chuỗi đã được mã hóa trở lại server.
- d. Kết thúc chương trình client.

## BÀI 4. [Object] TỔ CHỨC SỰ KIỆN

[Mã câu hỏi (qCode): FKRGvwLM]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để quản lý thông tin các sự kiện trong hệ thống tổ chức sự kiện. Chương trình sẽ ngẫu nhiên tạo ra đối tượng Event với các giá trị ban đầu và cung cấp cho RMI client như sau:

Giao diện từ xa:

```
public interface ObjectService extends Remote {
    public Serializable requestObject(String studentCode, String qCode) throws RemoteException;
    public void submitObject(String studentCode, String qCode, Serializable object) throws
RemoteException;
}
```

Lớp Event gồm các thuộc tính: id String, eventName String, eventDate String, expectedAttendance int, eventCode String.

- Trường dữ liệu: private static final long serialVersionUID = 20241131L;
- 02 hàm khởi dựng:

```
public Event()
    public Event(int id, String eventName, String eventDate, int expectedAttendance)
```

Trong đó:

- Interface ObjectService và lớp Event được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa ObjectService được đăng ký với RegistryServer: RMIOBJECTService.

## Luyện tập RMI

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với đối tượng sự kiện được nhận từ RMI Server:

- Triệu gọi phương thức `requestObject` để nhận đối tượng Event ngẫu nhiên từ server.
- Tạo mã `eventCode` cho sự kiện dựa trên các quy tắc sau:
  - Nếu `expectedAttendance`  $\geq 1000$ , thêm "L" vào `eventCode`; Nếu `expectedAttendance` từ 500 đến 999, thêm "M" vào `eventCode`; Nếu `expectedAttendance` dưới 500, thêm "S" vào `eventCode`.
  - Thêm chữ cái đầu và cuối của `eventName`, tất cả viết hoa.
  - Thêm ngày và tháng từ `eventDate` (theo định dạng "ddMM").Ví dụ: Nếu sự kiện có tên là "Charity Run", ngày diễn ra là "2024-05-12" và số lượng người tham gia dự kiến là 1200, mã `eventCode` sẽ là "LCR1205".
- Cập nhật giá trị `eventCode` trong đối tượng Event.
- Triệu gọi phương thức `submitObject` để gửi đối tượng Event đã được xử lý trở lại server.
- Kết thúc chương trình client.

### ĐỀ SỐ 12 [B21DCCN032]

#### BÀI 2. [Character] CHUYÊN ĐỔI THẬP PHÂN – LA MÃ

[Mã câu hỏi (qCode): ADu6zRYE]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi.

Giao diện từ xa:

```
public interface CharacterService extends Remote {  
    public String requestCharacter(String studentCode, String qCode) throws RemoteException;  
    public void submitCharacter(String studentCode, String qCode, String strSubmit) throws  
        RemoteException;  
}
```

Trong đó:

- Interface `CharacterService` được viết trong package `RMI`.
- Đối tượng cài đặt giao diện từ xa `CharacterService` được đăng ký với `RegistryServer` với tên là: `RMICharacterService`.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:

- Triệu gọi phương thức `requestCharacter` để nhận chuỗi ngẫu nhiên từ server với định dạng: "Số thập phân đầu vào".
- Thực hiện chuyển đổi số thập phân nhận được thành chuỗi số La Mã (Roman).

Quy tắc chuyển đổi:

Các ký tự La Mã chính bao gồm: I=1, V=5, X=10, L=50, C=100, D=500, M=1000.

Ví dụ: 58  $\rightarrow$  "LVIII".

- Triệu gọi phương thức `submitCharacter` để gửi chuỗi số La Mã đã chuyển đổi trở lại server.
- Kết thúc chương trình client.

### **BÀI 3. [Byte] MÃ HOÁ HEXA**

[Mã câu hỏi (qCode): pK0IZNnt]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

Giao diện từ xa:

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws RemoteException;  
}
```

Trong đó:

- Interface ByteService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa ByteService được đăng ký với RegistryServer với tên là: RMIByteService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân nhận được từ RMI Server:

- a. Triệu gọi phương thức requestData để nhận một mảng dữ liệu nhị phân (byte[]) từ server.
- b. Chuyển đổi mảng dữ liệu nhị phân nhận được thành một chuỗi biểu diễn hex. Mỗi byte trong mảng sẽ được chuyển đổi thành hai ký tự hex tương ứng.

Ví dụ: Nếu dữ liệu nhị phân nhận được là [72, 101, 108, 108, 111], chương trình sẽ chuyển đổi mảng này thành chuỗi hex "48656c6c6f", tương ứng với chuỗi "Hello" trong ASCII.

- c. Triệu gọi phương thức submitData để gửi chuỗi biểu diễn hex đã chuyển đổi thành mảng byte trở lại server.
- d. Kết thúc chương trình client.

### **ĐỀ SỐ 13 [B21DCCN036]**

### **BÀI 3. [Byte] NÉN RLE**

[Mã câu hỏi (qCode): 2uG0lQGi]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

Giao diện từ xa:

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws RemoteException;  
}
```

Trong đó:

- Interface ByteService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa ByteService được đăng ký với RegistryServer với tên là: RMIByteService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu byte nhận được từ RMI Server:

- a. Triệu gọi phương thức requestData để nhận một mảng dữ liệu byte từ server, đại diện cho một chuỗi dữ liệu byte tổng quát.

b. Thực hiện nén dữ liệu sử dụng thuật toán Run-Length Encoding (RLE) bằng cách ghi nhận mỗi byte và số lần lặp liên tiếp của nó. Kết quả nén là một mảng mới biểu diễn các cặp (byte, số lần lặp).

Ví dụ: Nếu dữ liệu byte nhận được là [10, 10, 10, 20, 20, 30, 30, 30, 30], chương trình sẽ thực hiện nén RLE như sau:

Kết quả nén RLE là mảng [10, 3, 20, 2, 30, 4].

c. Triệu gọi phương thức submitData để gửi mảng dữ liệu đã được nén RLE trở lại server.

d. Kết thúc chương trình client.

## ĐỀ SỐ 14 [B21DCCN038]

### BÀI 4. [Object] BÀI TOÁN QUẢN LÝ NHÂN VIÊN

[Mã câu hỏi (qCode): 7fSWnlHB]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý thông tin nhân viên. Chương trình sẽ ngẫu nhiên tạo ra đối tượng Employee với các giá trị ban đầu và cung cấp cho RMI client như sau:

Giao diện từ xa:

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qCode) throws RemoteException;  
    public void submitObject(String studentCode, String qCode, Serializable object) throws  
    RemoteException;  
}
```

Lớp Employee gồm các thuộc tính: id String, name String, baseSalary double, experienceYears int, finalSalary double.

Trường dữ liệu: private static final long serialVersionUID = 20241119L;

02 hàm khởi dựng:

```
public Employee()
```

```
public Employee(int id, String name, double baseSalary, int experienceYears)
```

Trong đó:

- Interface ObjectService và lớp Employee được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa ObjectService được đăng ký với RegistryServer với tên là: RMIOBJECTService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với đối tượng nhân viên được nhận từ RMI Server:

a. Triệu gọi phương thức requestObject với tham số đầu vào là mã sinh viên và mã câu hỏi để nhận đối tượng Employee ngẫu nhiên từ server.

b. Tính toán finalSalary dựa trên baseSalary, experienceYears, và áp dụng thuật toán xử lý số phức tạp để tạo hệ số điều chỉnh, yêu cầu sinh viên phải thao tác và xử lý experienceYears như sau:

- Tách experienceYears thành tổng các chữ số (ví dụ, 14 sẽ thành  $1 + 4 = 5$ ).
- Tính số lượng ước số nguyên của experienceYears (ví dụ, nếu experienceYears là 12, các ước số là 1, 2, 3, 4, 6, 12, nên có 6 ước số).

## Luyện tập RMI

---

- Xác định hệ số điều chỉnh factor dựa trên công thức sau:  $\text{factor} = (\text{Số năm làm việc} + \text{tổng các chữ số} + \text{số lượng ước số}) / 100.0$
  - Tính finalSalary theo công thức:  $\text{finalSalary} = \text{baseSalary} * (1 + \text{factor})$
- c. Cập nhật giá trị finalSalary trong đối tượng Employee.  
d. Triệu gọi phương thức submitObject để gửi đối tượng Employee đã được xử lý trở lại server.  
e. Kết thúc chương trình client.

### ĐỀ SỐ 15[B21DCCN040]

#### BÀI 3. [Byte] SẮP XẾP CHẴN LẺ

[Mã câu hỏi (qCode): Sua79nkc]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

Giao diện từ xa:

```
public interface ByteService extends Remote {  
    public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
    public void submitData(String studentCode, String qCode, byte[] data) throws RemoteException;  
}
```

Trong đó:

- Interface ByteService được viết trong package RMI.

Đối tượng cài đặt giao diện từ xa ByteService được đăng ký với RegistryServer với tên là: RMIByteService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân nhận được từ RMI Server:

- Triệu gọi phương thức requestData để nhận một mảng dữ liệu nhị phân (byte[]) từ server.
- Thực hiện phân chia mảng byte[] nhận được thành hai phần: phần đầu chứa các byte có giá trị chẵn và phần sau chứa các byte có giá trị lẻ, duy trì thứ tự xuất hiện của các phần tử trong từng nhóm.

Ví dụ: Nếu mảng dữ liệu nhận được là [1, 2, 3, 4, 5], sau khi phân chia chẵn-lẻ, mảng kết quả sẽ là [2, 4, 1, 3, 5] (tất cả phần tử chẵn được đặt trước, theo sau là tất cả phần tử lẻ).

- Triệu gọi phương thức submitData để gửi mảng byte[] đã được phân chia chẵn-lẻ trở lại server.
- Kết thúc chương trình client.

### ĐỀ SỐ 16 [B21DCCN048]

#### BÀI 4. [Object] QUẢN LÝ VÉ SỰ KIỆN

[Mã câu hỏi (qCode): Uct8bABt]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để quản lý thông tin vé trong hệ thống bán vé sự kiện. Chương trình sẽ ngẫu nhiên tạo ra đối tượng Ticket với các giá trị ban đầu và cung cấp cho RMI client như sau:

Giao diện từ xa:

```
public interface ObjectService extends Remote {  
    public Serializable requestObject(String studentCode, String qCode) throws RemoteException;
```

## Luyện tập RMI

---

```
public void submitObject(String studentCode, String qCode, Serializable object) throws  
RemoteException;  
}
```

Lớp Ticket gồm các thuộc tính: id String, eventName String, saleDate String, ticketCode String.

- Trường dữ liệu: private static final long serialVersionUID = 20241133L;
- 02 hàm khởi dựng:  
public Ticket()  
public Ticket(int id, String eventName, String saleDate)

Trong đó:

- Interface ObjectService và lớp Ticket được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa ObjectService được đăng ký với RegistryServer: RMIOBJECTSERVICE.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với đối tượng vé được nhận từ RMI Server:

- Triệu gọi phương thức requestObject để nhận đối tượng Ticket ngẫu nhiên từ server.
- Tạo mã ticketCode cho vé dựa trên các quy tắc sau:
  - Bắt đầu với hai chữ cái là: Chữ cái đầu tiên và cuối cùng của eventName, viết hoa.
  - Thêm ngày và tháng từ saleDate (theo định dạng "MMdd")
  - Kết thúc bằng hai chữ số là: Chữ số lớn nhất và nhỏ nhất không xuất hiện trong saleDate.Ví dụ: tên sự kiện là "Charity Concert", và ngày bán là "15/06/2024", mã ticketCode sẽ là "CT061593".
- Cập nhật giá trị ticketCode trong đối tượng Ticket.
- Triệu gọi phương thức submitObject để gửi đối tượng Ticket đã được xử lý trở lại server.
- Kết thúc chương trình client.

## ĐỀ SỐ 17 [B21DCCN053]

### BÀI 1. [Data] SINH TỔ HỢP

[Mã câu hỏi (qCode): juC3u7C6]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu.

Giao diện từ xa:

```
public interface DataService extends Remote {  
public Object requestData(String studentCode, String qCode) throws RemoteException;  
public void submitData(String studentCode, String qCode, Object data) throws RemoteException;  
}
```

Trong đó:

- Interface DataService được viết trong package RMI.
- Đối tượng cài đặt giao diện từ xa DataService được đăng ký với RegistryServer với tên là: RMIDataService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhận được từ RMI Server:

## Luyện tập RMI

---

- Triệu gọi phương thức `requestData` để nhận một chuỗi gồm tập hợp số nguyên và một số nguyên  $K$  từ server với định dạng: “mảng; số nguyên  $K$ ”.
- Sử dụng thuật toán sinh tổ hợp để tạo ra tất cả các tổ hợp kích thước  $K$  của tập hợp đã cho. Kết quả trả về là danh sách các tổ hợp con có  $K$  phần tử.  
Ví dụ: Với tập hợp  $[1, 2, 3, 4]$  và  $K = 2$ , kết quả là danh sách các tổ hợp  $[[1, 2], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4]]$ .
- Triệu gọi phương thức `submitData` để gửi chuỗi chứa danh sách các tổ hợp đã sinh được trở lại server.
- Kết thúc chương trình client.

## BÀI 2. [Character] MÃ HOÁ URL

[Mã câu hỏi (qCode): KkihaRAB]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý chuỗi.

Giao diện từ xa:

```
public interface CharacterService extends Remote {  
    public String requestCharacter(String studentCode, String qCode) throws RemoteException;  
    public void submitCharacter(String studentCode, String qCode, String strSubmit) throws  
        RemoteException;  
}
```

Trong đó:

- Interface `CharacterService` được viết trong package `RMI`.
- Đối tượng cài đặt giao diện từ xa `CharacterService` được đăng ký với `RegistryServer` với tên là: `RMICharacterService`.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với chuỗi được nhận từ RMI Server:

- Triệu gọi phương thức `requestCharacter` để nhận chuỗi ngẫu nhiên từ server với định dạng: "Chuỗi đầu vào".
- Thực hiện thao tác mã hóa URL (URL Encoding) cho chuỗi đầu vào nhận được từ server. Mã hóa URL chuyển đổi các ký tự đặc biệt thành định dạng URL an toàn bằng cách thay thế các ký tự đó bằng ký hiệu phần trăm (%) và mã ASCII của chúng.  
Ví dụ: Chuỗi ban đầu "Hello World!" -> Chuỗi mã hóa URL là: "Hello%20World%21"
- Triệu gọi phương thức `submitCharacter` để gửi chuỗi đã được mã hóa trở lại server.
- Kết thúc chương trình client.

## ĐỀ SỐ 18[B21DCCN066]

## BÀI 3. [Byte] PHẦN TỬ LỚN THỨ K

[Mã câu hỏi (qCode): uIKHCTWG]. Một chương trình (tạm gọi là RMI Server) cung cấp giao diện cho phép triệu gọi từ xa để xử lý dữ liệu nhị phân.

Giao diện từ xa:

```
public interface ByteService extends Remote {
```



## Luyện tập RMI

---

```
public byte[] requestData(String studentCode, String qCode) throws RemoteException;  
public void submitData(String studentCode, String qCode, byte[] data) throws RemoteException;  
}
```

Trong đó:

- Interface ByteService được viết trong package RMI.

Đối tượng cài đặt giao diện từ xa ByteService được đăng ký với RegistryServer với tên là: RMIByteService.

Yêu cầu: Viết chương trình tại máy trạm (RMI client) để thực hiện các công việc sau với dữ liệu nhị phân nhận được từ RMI Server:

- Triệu gọi phương thức requestData để nhận một mảng dữ liệu nhị phân (byte[]) từ server, cùng với một số nguyên K, đại diện cho thứ tự phần tử cần tìm.
- Tìm phần tử lớn thứ K trong mảng byte[] với K là phần tử cuối cùng của mảng.  
Ví dụ: Nếu mảng dữ liệu nhận được là [3, 1, 5, 12, 2, 11, 3], giá trị K = 3, chương trình sẽ tìm phần tử lớn thứ ba -> Kết quả là 5.
- Triệu gọi phương thức submitData để gửi mảng byte gồm phần tử lớn thứ K và vị trí của K trở lại server.
- Kết thúc chương trình client.