**ÔN TẬP MÔN LẬP TRÌNH MẠNG VỚI JAVA**

Bài 1. [Mã câu hỏi (qCode): m81LDpdo]. Một chương trình server cho phép kết nối qua giao thức TCP tại cổng 2208 (hỗ trợ thời gian giao tiếp tối đa cho mỗi yêu cầu là 5s). Yêu cầu là xây dựng một chương trình client tương tác với server sử dụng các luồng byte (BufferedWriter/BufferedReader) theo kịch bản sau:

a. Gửi một chuỗi gồm mã sinh viên và mã câu hỏi với định dạng "studentCode;qCode". Ví dụ: "B15DCCN999;EC4F899B"

b. Nhận một chuỗi ngẫu nhiên là danh sách các một số tên miền từ server

Ví dụ: giHgWHwkLf0Rd0.io, I7jpjuRw13D.io, wXf6GP3KP.vn, MdpIzhxDVtTFTF.edu, TUHuMfn25chmw.vn, HHjE9.com, 4hJld2m2yiweto.vn, y2L4SQwH.vn, s2aUrZGdzS.com, 4hXfJe9giAA.edu

c. Tìm kiếm các tên miền .edu và gửi lên server

Ví dụ: MdpIzhxDVtTFTF.edu, 4hXfJe9giAA.edu

d. Đóng kết nối và kết thúc chương trình.

\*/

**BÀI 2. SẮP XẾP CHẴN LẺ [In]**

Một chương trình server tại địa chỉ 172.188.19.218 hỗ trợ kết nối qua giao thức TCP tại cổng 807 (hỗ trợ thời gian giao tiếp tối đa cho mỗi yêu cầu là 5s).

Yêu cầu xây dựng chương trình client thực hiện kết nối tới server trên sử dụng luồng byte dữ liệu (InputStream/OutputStream) để trao đổi thông tin theo thứ tự:

1. Gửi mã sinh viên và mã câu hỏi theo định dạng "studentCode;qCode".

Ví dụ: "B20DCCN999;ABCDEF" với ABCDEF là mã bài tập đã đề cập ở trên.

1. Nhận dữ liệu từ server là một xâu ký tự, chứa các số nguyên chưa sắp xếp, các số được ngăn cách với nhau bởi dấu phẩy. Ví dụ: “2,15,4,3,6,8,10,7,1”
2. Sắp xếp tăng dần các giá trị chẵn, rồi đến các giá trị lẻ và gửi lên server theo định dạng như sau:

“[2, 4, 6, 8, 10];[1, 3, 7, 15]”

1. Đóng kết nối và kết thúc chương trình

**BÀI 3. ĐẾM SỐ LẦN XUẤT HIỆN - 1** *(Mã câu hỏi: 721)* **[Buffer]**

Một chương trình server tại địa chỉ 172.188.19.218 cho phép kết nối qua giao thức TCP tại cổng 808 (hỗ trợ thời gian giao tiếp tối đa cho mỗi yêu cầu là 5s). Yêu cầu là xây dựng một chương trình client tương tác với server sử dụng các luồng byte (BufferedWriter/BufferedReader) theo kịch bản sau:

1. Gửi mã sinh viên và mã câu hỏi theo định dạng "studentCode;qCode".

Ví dụ: "B20DCCN999;ABCDEF" với ABCDEF là mã bài tập đã đề cập ở trên.

1. Nhận dữ liệu từ server là một xâu ký tự. Ví dụ: dgUOo ch2k22ldsOo
2. Liệt kê các ký tự xuất hiện **nhiều hơn một lần** theo định dạng. Ví dụ: “d:2,O:2,o:2,2:3”
3. Đóng kết nối và kết thúc chương trình

BÀI 4. auzMOkdP - TCP - Object Stream

[Mã câu hỏi (qCode): auzMOkdP]. Thông tin khách hàng cần thay đổi định dạng lại cho phù hợp với khu vực, cụ thể:

a. Tên khách hàng cần được chuẩn hóa theo định dạng mới. Ví dụ: nguyen van hai duong -> DUONG, Nguyen Van Hai

b. Ngày sinh của khách hàng hiện đang ở dạng mm-dd-yyyy, cần được chuyển thành định dạng dd/mm/yyyy. Ví dụ: 10-11-2012 -> 11/10/2012

c. Tài khoản khách hàng là các chữ cái in thường được sinh tự động từ họ tên khách hàng. Ví dụ: nguyen van hai duong -> nvhduong

Một chương trình server cho phép kết nối qua giao thức TCP tại cổng 2209 (hỗ trợ thời gian giao tiếp tối đa cho mỗi yêu cầu là 5s). Yêu cầu là xây dựng một chương trình client tương tác với server sử dụng các luồng đối tượng (ObjectInputStream / ObjectOutputStream) thực hiện gửi/nhận đối tượng khách hàng và chuẩn hóa. Cụ thể:

a. Đối tượng trao đổi là thể hiện của lớp Customer918 được mô tả như sau

• Tên đầy đủ của lớp: TCP.Customer918

• Các thuộc tính: id int, code String, name String, dayOfBirth String, userName String

• Hàm khởi tạo đầy đủ các thuộc tính được liệt kê ở trên

• Trường dữ liệu: private static final long serialVersionUID = 918;

b. Tương tác với server theo kịch bản dưới đây:

1) Gửi đối tượng là một chuỗi gồm mã sinh viên và mã câu hỏi ở định dạng "studentCode;qCode". Ví dụ: "B15DCCN999;F2DA54F3"

2) Nhận một đối tượng là thể hiện của lớp Customer918 từ server với các thông tin đã được thiết lập

3) Thay đổi định dạng theo các yêu cầu ở trên và gán vào các thuộc tính tương ứng. Gửi đối tượng đã được sửa đổi lên server

4) Đóng socket và kết thúc chương trình.

BÀI 5: piSQf5ct - UDP - String

[Mã câu hỏi (qCode): piSQf5ct]. [Loại bỏ ký tự đặc biệt, số, trùng và giữ nguyên thứ tự xuất hiện]

Một chương trình server cho phép kết nối qua giao thức UDP tại cổng 2208 . Yêu cầu là xây dựng một chương trình client trao đổi thông tin với server theo kịch bản dưới đây:

a. Gửi thông điệp là một chuỗi chứa mã sinh viên và mã câu hỏi theo định dạng ";studentCode;qCode". Ví dụ: ";B15DCCN001;06D6800D"

b. Nhận thông điệp là một chuỗi từ server theo định dạng "requestId;strInput"

• requestId là chuỗi ngẫu nhiên duy nhất

• strInput là chuỗi thông điệp cần xử lý

c. Thực hiện loại bỏ ký tự đặc biệt, số, ký tự trùng và giữ nguyên thứ tự xuất hiện của chúng. Gửi thông điệp lên server theo định dạng "requestId;strOutput", trong đó strOutput là chuỗi đã được xử lý ở trên

d. Đóng socket và kết thúc chương trình.

BÀI 6: O1UMX4aS - TCP - Byte Stream

[Mã câu hỏi (qCode): O1UMX4aS]. Một chương trình server cho phép kết nối qua giao thức TCP tại cổng 2206 (thời gian giao tiếp tối đa cho mỗi yêu cầu là 5s). Yêu cầu là xây dựng một chương trình client tương tác tới server ở trên sử dụng các luồng byte (InputStream/OutputStream) để trao đổi thông tin theo thứ tự:

a. Gửi mã sinh viên và mã câu hỏi theo định dạng "studentCode;qCode". Ví dụ: "B16DCCN999;2B3A6510"

b. Nhận dữ liệu từ server là một chuỗi các giá trị số nguyên được phân tách nhau bởi ký tự ",". Ví dụ: 1,3,9,19,33,20

c. Tìm và gửi lên server giá trị lớn thứ hai cùng vị trí xuất hiện của nó trong chuỗi.Ví dụ: 20,5

d. Đóng kết nối và kết thúc chương trình.

BÀI 7: awS5eLtD - UDP - Object

[Mã câu hỏi (qCode): awS5eLtD]. Thông tin khách hàng được yêu cầu thay đổi định dạng lại cho phù hợp với khu vực, cụ thể:

a. Tên khách hàng cần được chuẩn hóa theo định dạng mới. Ví dụ: nguyen van hai duong -> DUONG, Nguyen Van Hai

b. Ngày sinh của khách hàng đang ở dạng mm-dd-yyyy, cần được chuyển thành định dạng dd/mm/yyyy. Ví dụ: 10-11-2012 -> 11/10/2012

c. Tài khoản khách hàng được tạo từ các chữ cái in thường được sinh tự động từ họ tên khách hàng. Ví dụ: nguyen van hai duong -> nvhduong

Một chương trình server cho phép giao tiếp qua giao thức UDP tại cổng 2209. Yêu cầu là xây dựng một chương trình client giao tiếp với server theo mô tả sau:

a. Đối tượng trao đổi là thể hiện của lớp UDP.Customer được mô tả như sau

• Tên đầy đủ của lớp: UDP.Customer

• Các thuộc tính: id String, code String, name String, , dayOfBirth String, userName String

• Một Hàm khởi tạo với đầy đủ các thuộc tính được liệt kê ở trên

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

b. Client giao tiếp với server theo các bước

• Gửi thông điệp là một chuỗi chứa mã sinh viên và mã câu hỏi theo định dạng “;studentCode;qCode”. Ví dụ: “;B15DCCN001;EE29C059”

• Nhận thông điệp chứa: 08 byte đầu chứa chuỗi requestId, các byte còn lại chứa một đối tượng là thể hiện của lớp Customer từ server. Trong đó, các thuộc tính id, code, name,dayOfBirth đã được thiết lập sẵn.

• Yêu cầu thay đổi thông tin các thuộc tính như yêu cầu ở trên và gửi lại đối tượng khách hàng đã được sửa đổi lên server với cấu trúc:

08 byte đầu chứa chuỗi requestId và các byte còn lại chứa đối tượng Customer đã được sửa đổi.

• Đóng socket và kết thúc chương trình.

BÀI 8: XYKJXgmx - UDP - Data Type

[Mã câu hỏi (qCode): XYKJXgmx]. Một chương trình server cho phép giao tiếp qua giao thức UDP tại cổng 2207. Yêu cầu là xây dựng một chương trình client trao đổi thông tin với server theo kịch bản:

a. Gửi thông điệp là một chuỗi chứa mã sinh viên và mã câu hỏi theo định dạng ";studentCode;qCode". Ví dụ: ";B15DCCN009;F3E8B2D4".

b. Nhận thông điệp là một chuỗi từ server theo định dạng "requestId;n, n", với:

--- requestId là chuỗi ngẫu nhiên duy nhất.

--- n là một số nguyên ngẫu nhiên ≤ 100.

c. Tính và gửi về server danh sách n số nguyên tố đầu tiên theo định dạng "requestId;p1,p2,...,pk", trong đó p1,p2,...,pk là các số nguyên tố.

d. Đóng socket và kết thúc chương trình.

BÀI 9: CSv4zu4f - RMI - Character

[Mã câu hỏi (qCode): CSv4zu4f]. 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:

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

b. 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.

c. Triệu gọi phương thức submitCharacter để gửi số thập phân đã chuyển đổi trở lại server.

d. Kết thúc chương trình client.

BÀI 10: ys1kmL1l - RMI - Object

[Mã câu hỏi (qCode): ys1kmL1l]. 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.

BÀI 11: baNyILKE - TCP - Data Stream

[Mã câu hỏi (qCode): baNyILKE]. Một chương trình server cho phép kết nối qua TCP tại cổng 2207 (hỗ trợ thời gian liên lạc tối đa cho mỗi yêu cầu là 5 giây). Yêu cầu là xây dựng chương trình client tương tác với server bằng các byte stream (DataInputStream/DataOutputStream) để trao đổi thông tin theo trình tự sau:

a. Gửi một chuỗi chứa mã sinh viên và mã câu hỏi ở định dạng "studentCode;qCode". Ví dụ: "B10DCCN000;A1B2C3D4".

b. Nhận từ server một mảng chứa n số nguyên, với n được gửi từ máy chủ. Ví dụ: Server gửi mảng [5, 9, 3, 6, 8].

c. Tính tổng, trung bình cộng, và phương sai của mảng. Gửi kết quả lần lượt lên server dưới dạng số nguyên và float. Ví dụ, gửi lên lần lượt: 31, 6.2, 4.5599995.

d. Đóng kết nối và kết thúc chương trình.

BÀI 12: FvoPU4bP - TCP - Character Stream

[Mã câu hỏi (qCode): FvoPU4bP]. Một chương trình server cho phép kết nối qua giao thức TCP tại cổng 2208 (hỗ trợ thời gian giao tiếp tối đa cho mỗi yêu cầu là 5 giây). Yêu cầu là xây dựng một chương trình client thực hiện kết nối tới server và sử dụng luồng ký tự (BufferedWriter/BufferedReader) để trao đổi thông tin theo kịch bản sau:

a. Gửi một chuỗi gồm mã sinh viên và mã câu hỏi với định dạng "studentCode;qCode". Ví dụ: "B15DCCN999;A1B2C3D4".

b. Nhận từ server một chuỗi ngẫu nhiên.

c. Xử lý chuỗi đã nhận theo các bước sau:

Bước 1: Tìm từ dài nhất trong trong chuỗi ngẫu nhiên (từ là chuỗi con phân tách bởi khoảng trắng).

Bước 2: Xác định vị trí bắt đầu của từ dài nhất đó trong chuỗi ban đầu.

d. Gửi lần lượt hai giá trị lên server:

- Từ dài nhất xuất hiện trong chuỗi.

- Vị trí bắt đầu của từ trong chuỗi ban đầu.

e. Đóng kết nối và kết thúc chương trình.

BÀI 13: LypGzI3m - RMI - Byte

[Mã câu hỏi (qCode): LypGzI3m]. 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.

BÀI 14: 0DJptnqK - RMI - Data

[Mã câu hỏi (qCode): 0DJptnqK]. 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 15: M7H6fSPk - WS - ObjectService

[Mã câu hỏi (qCode): M7H6fSPk]. Một dịch vụ web được định nghĩa và mô tả trong tệp ObjectService.wsdl, được triển khai trên server tại URL http://<Exam\_IP>:8080/JNPWS/ObjectService?wsdl để xử lý các bài toán với đối tượng.

Yêu cầu: Viết chương trình tại máy trạm (WS client) để giao tiếp với ObjectService thực hiện các công việc sau:

a. Triệu gọi phương thức requestObject với tham số đầu vào là mã sinh viên (studentCode) và mã câu hỏi (qCode) để nhận về một danh sách đối tượng WS.Student từ server. Mỗi đối tượng Student có các thuộc tính:

• name: kiểu String, đại diện cho tên của sinh viên.

• score: kiểu float, đại diện cho điểm trung bình của sinh viên.

b. Thực hiện tạo danh sách phân nhóm các sinh viên dựa trên điểm trung bình theo các mức sau: A: điểm từ 8.0 trở lên; B: điểm từ 6.5 đến dưới 8.0; C: điểm từ 5.0 đến dưới 6.5; D: điểm dưới 5.0

c. Triệu gọi phương thức submitObject(String studentCode, String qCode, List<String[]> data) để gửi danh sách nhóm sinh viên đã phân loại trở lại server.

d. Kết thúc chương trình client.

BÀI 16: Z5N771p9 - WS - DataService

[Mã câu hỏi (qCode): Z5N771p9]. Một dịch vụ web được định nghĩa và mô tả trong tệp DataService?wsdl, được triển khai trên server tại URL http://<Exam\_IP>:8080/JNPWS/DataService?wsdl để xử lý các bài toán với dữ liệu nguyên thủy.

Yêu cầu: Viết chương trình tại máy trạm (WS client) để giao tiếp với DataService thực hiện các công việc sau:

a. Triệu gọi phương thức getData với tham số đầu vào là mã sinh viên (studentCode) và mã câu hỏi (qCode) để nhận về một mảng số nguyên (int[]) từ server.

b. Với mảng số nguyên nhận được, thực hiện đếm số lần xuất hiện của từng phần tử trong mảng và lưu kết quả tần suất vào một mảng mới. Mỗi phần tử trong mảng mới này sẽ chứa hai giá trị: giá trị của phần tử và số lần xuất hiện tương ứng của nó.

c. Triệu gọi phương thức submitDataIntMatrix(String studentCode, String qCode, int[][] data) để gửi mảng kết quả chứa cặp giá trị [phần tử, tần suất] của từng phần tử đã tính được trở lại server.

Ví dụ: Nếu mảng số nguyên nhận được từ phương thức getData là [1, 2, 2, 3, 3, 3], kết quả tần suất sẽ là [[1, 1], [2, 2], [3, 3]], và mảng kết quả này sẽ được gửi lại server qua phương thức submitData.

d. Kết thúc chương trình client.

BÀI 17: ujSMrHrI - WS - CharacterService

[Mã câu hỏi (qCode): ujSMrHrI]. Một dịch vụ web được định nghĩa và mô tả trong tệp CharacterService.wsdl, được triển khai trên server tại URL http://<Exam\_IP>:8080/JNPWS/CharacterService?wsdl để xử lý các bài toán về chuỗi và ký tự.

Yêu cầu: Viết chương trình tại máy trạm (WS client) để giao tiếp với CharacterService thực hiện các công việc sau:

a. Triệu gọi phương thức requestCharacter với tham số đầu vào là mã sinh viên (studentCode) và mã câu hỏi (qCode) để nhận về một mảng ký tự (char[]) từ server.

b. Sắp xếp mảng ký tự nhận được theo thứ tự từ điển (alphabetical order).

c. Triệu gọi phương thức submitCharacterCharArray(String studentCode, String qCode, char[] data) để gửi mảng ký tự đã sắp xếp trở lại server.

Ví dụ: Nếu mảng ký tự nhận được từ phương thức requestCharacter là ['d', 'a', 'c', 'b'], sau khi sắp xếp theo thứ tự từ điển, mảng kết quả sẽ là ['a', 'b', 'c', 'd']. Mảng này sẽ được gửi lại server qua phương thức submitCharacter.

d. Kết thúc chương trình client.