viettel
software services

CHƯƠNG TRÌNH ĐÀO TẠO JAVA NÂNG CAO

Java Networking

Đơn vị tổ chức: Phòng đào tạo và phát triển nguồn nhân lực

viettel
software services

NỘI DUNG HỌC PHẦN

BÀI 1

Java Networking

BÀI 2

Design Pattern

viettel
software services

BÀI 1

Java Networking

viettel
software services

MỤC LỤC

01 | Network

02 | Java Networking

viettel
software services

01

Network

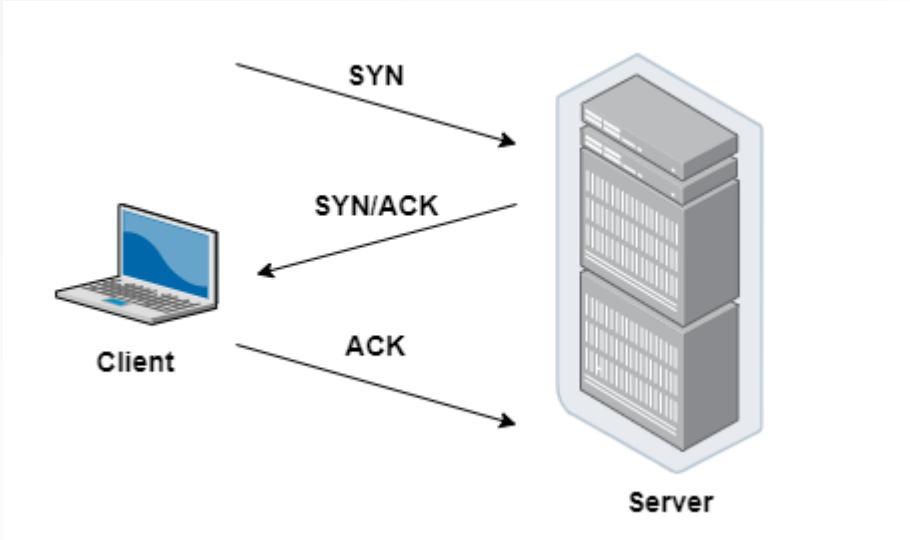
viettel
software services

TCP/IP và UDP

- **UDP và TCP** là hai giao thức truyền dẫn qua **internet** phổ biến và thông dụng nhất hiện nay.
- **Transmission Control Protocol (TCP)** là giao thức tiêu chuẩn trên Internet đảm bảo trao đổi thành công các gói dữ liệu giữa các thiết bị qua mạng. **TCP hoạt động với giao thức Internet (IP)** để chỉ định cách dữ liệu được trao đổi trực tuyến. **IP** chịu trách nhiệm gửi từng gói đến đích của nó, trong khi **TCP đảm bảo rằng các byte được truyền theo thứ tự** mà chúng được gửi mà không có lỗi hoặc thiếu sót nào. **Hai giao thức kết hợp với nhau được gọi là TCP/IP**.
- Giao thức UDP hoạt động tương tự như TCP, nhưng nó **bỏ qua quá trình kiểm tra lỗi**. Khi một ứng dụng sử dụng giao thức UDP, các gói tin được gửi cho bên nhận và **bên gửi không phải chờ để đảm bảo bên nhận đã nhận được gói tin**, do đó nó lại tiếp tục gửi gói tin tiếp theo. Nếu bên nhận bỏ lỡ một vài gói tin UDP, họ sẽ mất vì bên gửi không gửi lại chúng. Do đó thiết bị có thể giao tiếp nhanh hơn.

Cách hoạt động của TCP

- **Quá trình giao vận của TCP bao gồm 3 bước.**
- **Thiết lập kết nối:** Client gửi một gói tin có cờ **SYN** cho server để yêu cầu mở cổng dịch vụ. Sau khi nhận được gói tin, server gửi lại gói tin có cờ **SYN-ACK** để xác nhận. Sau khi kết nối đã được thiết lập, client gửi tới server gói tin **ACK** để đáp ứng nhu cầu của server.



viettel
software services

Cách hoạt động của TCP

- **Truyền dữ liệu:**
- Sau khi kết nối được thiết lập, **TCP** hoạt động bằng cách chia nhỏ dữ liệu đã truyền thành các **segment (phân đoạn)**, mỗi **segment được đóng gói** thành một gói dữ liệu và được gửi đến đích của nó. TCP dán nhãn các gói tin theo dạng đánh số.
- Với mỗi gói tin nhận được, thiết bị sẽ yêu cầu bên nhận gửi phản hồi đã nhận được cho bên gửi thông qua một gói xác nhận. Tin báo nhận chính là tín hiệu về tình trạng đường truyền giữa hai bên. Sau khi hết thời gian chờ, không nhận được xác nhận, nguồn gửi sẽ gửi lại gói tin bị mất hoặc bị hoãn. Nhờ vậy, các vấn đề về lặp gói tin, truyền lại các gói dữ liệu bị hỏng hoặc mất và sai thứ tự gói tin đều được giải quyết.

viettel
software services

Cách hoạt động của TCP

- **Kết thúc kết nối:** Khi muốn đóng liên kết, bên client sẽ gửi đi một gói tin FIN cho server. Sau khi nhận được gói FIN, server gửi lại gói ACK để hồi đáp, đồng thời vào trạng thái đóng liên kết và gửi tiếp gói FIN. Sau khi client nhận được gói FIN, client sẽ gửi gói ACK để xác nhận. Cuối cùng, server nhận được gói ACK xác nhận và đóng liên kết.

viettel
software services

Cấu trúc gói tin TCP

- **Source port** (16 bit): Số cổng của thiết bị gửi.
- **Destination port** (16 bit): Số cổng của thiết bị nhận.
- **Sequence number** (32 bit): Dùng để đánh số thứ tự gói tin (từ số sequense nó sẽ tính ra được số byte đã được truyền)
- **Acknowledgment number** (32 bit): Dùng để báo đã nhận được gói tin nào và mong nhận được byte mang số thứ tự nào tiếp theo.
- **DO** (4 bit): Cho biết toàn bộ header dài bao nhiêu tính theo đơn vị word (1 Word = 4 byte).
- **RSV** (4 bit): Đều được thiết lập bằng 0.

viettel
software services

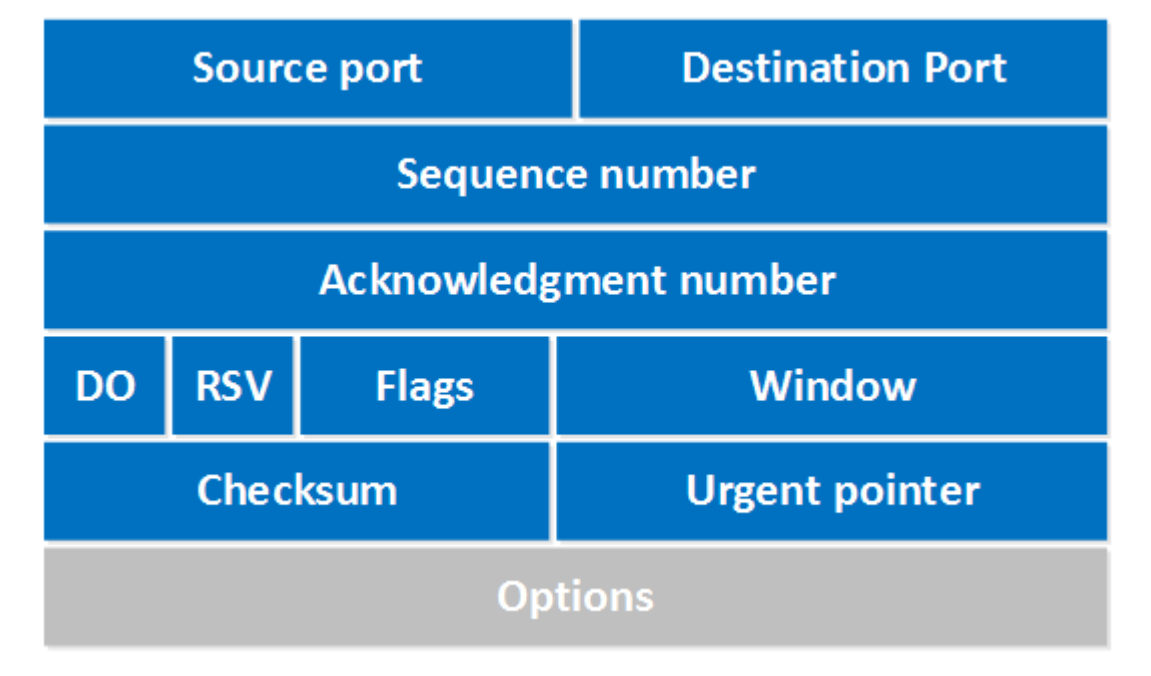
Cấu trúc gói tin TCP

- **Flags** (9 bit): Được sử dụng để thiết lập kết nối, gửi dữ liệu và chấm dứt kêt nối.
  - URG: Ưu tiên dữ liệu này hơn các dữ liệu khác.
  - ACK: Được sử dụng để xác nhận.
  - PSH: Segment yêu cầu chức năng push.
  - RST: Thiết lập lại kết nối.
  - SYN: Được sử dụng để đặt số thứ tự ban đầu.
  - FIN: Kết thúc kết nối TCP.
- **Windows** (16 bit): Số lượng byte được thiết bị sẵn sàng tiếp nhận.
- **Checksum** (16 bit): Kiểm tra lỗi của toàn bộ TCP segment.
- **Urgent pointer** (16 bit): Sử dụng trong trường hợp cần ưu tiên dữ liệu.
- **Options** (tối đa 32 bit): Cho phép thêm vào TCP các tính năng khác.

Để xem các trường này hoạt động, các bạn có thể sử dụng công cụ Wireshard.

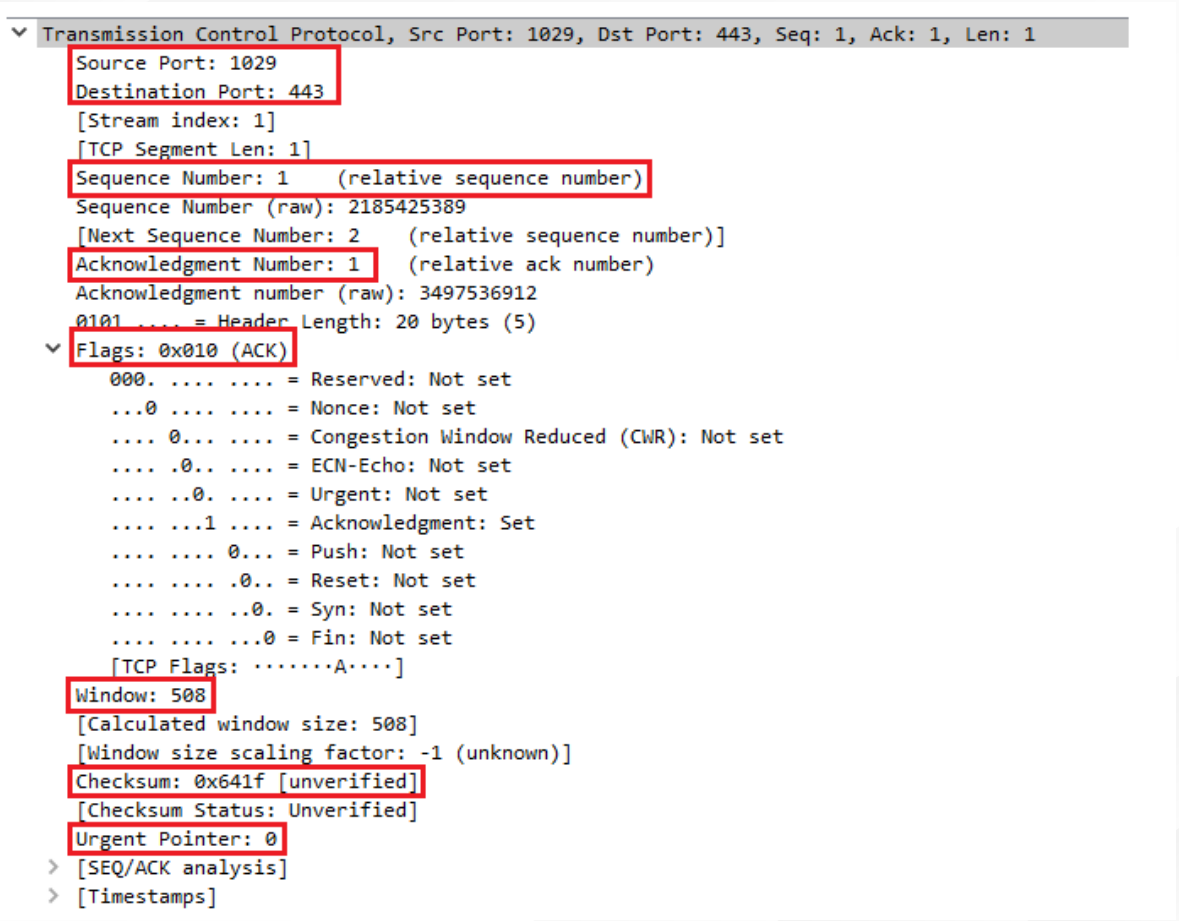
viettel
software services

Cấu trúc gói tin TCP



viettel
software services

Cấu trúc gói tin TCP sử dụng wireshard



viettel
software services

Ứng dụng của TCP

- Vì tính ứng dụng thực tế cao của nó, TCP/IP gần như là phủ sóng mức độ ảnh hưởng và tầm quan trọng của nó trong giao thức mạng. Sau đây là 3 loại giao thức TCP/IP phổ biến nhất:
- **HTTP**: **HTTP** được sử dụng để truyền dữ liệu không an toàn giữa một web client và một web server Theo quy trình, web client (trình duyệt Internet trên máy tính) sẽ gửi một yêu cầu đến một web server để xem một website. Sau đó, máy chủ web nhận được yêu cầu đó và gửi thông tin website về cho web client.
- **HTTPS**: HTTPS được sử dụng để truyền dữ liệu an toàn giữa một web client và một web server. Ngày nay giao thức HTTPS được sử dụng thay thế cho HTTP.
- **FTP**: **FTP** là phương thức trao đổi file được sử dụng giữa hai hoặc nhiều máy tính thông qua Internet. Nhờ **FTP**, các máy tính có thể gửi và nhận dữ liệu đến nhau một các trực tiếp.

viettel
software services

Ứng dụng của TCP

- Vì tính ứng dụng thực tế cao của nó, TCP/IP gần như là phủ sóng mức độ ảnh hưởng và tầm quan trọng của nó trong giao thức mạng. Sau đây là 3 loại giao thức TCP/IP phổ biến nhất:
- **HTTP**: **HTTP** được sử dụng để truyền dữ liệu không an toàn giữa một web client và một web server Theo quy trình, web client (trình duyệt Internet trên máy tính) sẽ gửi một yêu cầu đến một web server để xem một website. Sau đó, máy chủ web nhận được yêu cầu đó và gửi thông tin website về cho web client.
- **HTTPS**: HTTPS được sử dụng để truyền dữ liệu an toàn giữa một web client và một web server. Ngày nay giao thức HTTPS được sử dụng thay thế cho HTTP.
- **FTP**: **FTP** là phương thức trao đổi file được sử dụng giữa hai hoặc nhiều máy tính thông qua Internet. Nhờ **FTP**, các máy tính có thể gửi và nhận dữ liệu đến nhau một các trực tiếp.

viettel
software services

UDP

- **Ngược lại với giao thức TCP** thì **UDP** là giao thức truyền **tải hướng không kết nối** (**connectionless**). Nó sẽ không thực hiện thao tác xây dựng kết nối trước khi truyền dữ liệu mà thực hiện truyền ngay lập tức khi có dữ liệu cần truyền (kiểu truyền best effort) => **truyền tải rất nhanh** cho dữ liệu của lớp ứng dụng.
- **Không đảm bảo tính tin cậy khi truyền dữ liệu** và **không có cơ chế phục hồi dữ liệu** (nó không quan tâm gói tin có đến đích hay không, không biết gói tin có bị mất mát trên đường đi hay không) => dễ bị lỗi.
- Không thực hiện các biện pháp đánh số thứ tự cho các đơn vị dữ liệu được truyền…
- Nhanh và hiệu quả hơn đối với các dữ liệu có kích thước nhỏ và yêu cầu khắt khe về thời gian.
- Bản chất không trạng thái nên UDP hữu dụng đối với việc trả lời các truy vấn nhỏ với số lượng lớn người yêu cầu.

viettel
software services

Cách hoạt động UDP

- **UDP** hoạt động tương tự như TCP nhưng nó không cung cấp kiểm tra lỗi khi truyền gói tin.
- Khi một ứng dụng sử dụng **UDP**, các gói tin chỉ được gửi đến người nhận. Người gửi không đợi để đảm bảo người nhận nhận được gói tin hay không, mà tiếp tục gửi các gói tiếp theo. Nếu người nhận miss mất một vài gói tin UDP thì gói tin đó coi như bị mất vì người gửi sẽ không gửi lại chúng. => Các thiết bị có thể giao tiếp nhanh hơn.

viettel
software services

Cấu trúc gói tin UDP

- **Cấu trúc gói tin UDP thì đơn giản hơn rất nhiều so với TCP**
  - **source port** và **destination port**(đều 16 bit): cho phép định danh một session của một ứng dụng nào đó chạy trên UDP. Có thể coi port chính là địa chỉ của tầng Transport
  - **UDP length**(16 bit): cho biết chiều dài của toàn bộ UDP datagram tổng cộng bao nhiêu byte. (16 bit thì sẽ có tổng cộng 2^16 byte = 65536 giá trị (từ 0 -> 65535 byte)).
  - **UDP checksum**(16 bit): sử dụng thuật toán mã vòng CRC để kiểm lỗi cho toàn bộ UDP datagram và chỉ kiểm tra một cách hạn chế
  - **Data**: dữ liệu tầng trên được đóng gói **vào UDP datagram** đang xét.

viettel
software services

Cấu trúc gói tin UDP

viettel
software services

Các ứng dụng của UDP

- Với ưu điểm là nhanh, nhẹ và hỗ trợ **broadcast**, UDP có thể được áp dụng cho những lĩnh vực như:
- **Media streaming:** Ví dụ như việc stream video, có lost vài khung hình cũng sẽ chấp nhận được, tuy nhiên đổi lại tốc độ load nhanh.
- **Gaming**: Tưởng tượng bạn chơi Liên Minh Huyền Thoại và cần upload liên tục những hành động như click chuột, bấm phím để di chuyển và tung chiêu, thì tần suất để gửi dữ liệu rất nhiều. Do đó **UDP** sẽ rất thích hợp.
- **DNS** Lookup…

viettel
software services

So sánh TCP và UDP

TCP	UDP
Đáng tin cậy	Không đáng tin cậy
Chậm hơn do nhiều RTT hơn	Nhanh hơn
Header nặng hơn (20-60 bytes)	Header 8 bytes nhẹ hơn
Không hỗ trợ Broadcast	Hỗ trợ Broadcast và Multicast
Quản lý được connection dựa vào các gói SYN/ACK/PSH...	Connectionless
Quản lý lỗi, chống tắc nghẽn	Không có
TCP là nền tảng của HTTP, HTTPs, FTP, SMTP and Telnet.	UDP là nền tảng của DNS, DHCP, TFTP, SNMP, RIP, and VoIP.

viettel
software services

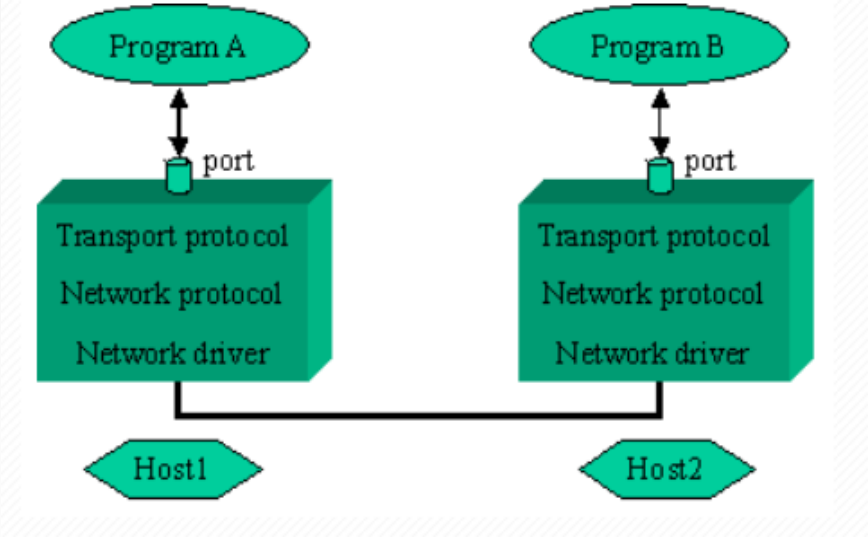
02

Java Networking

viettel
software services

Socket

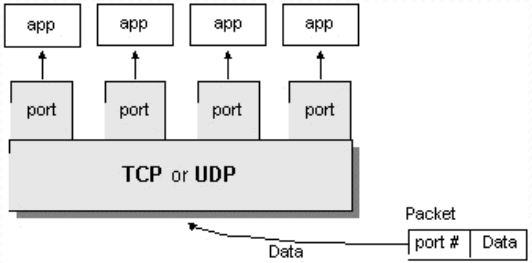
- **Socket** là một giao diện lập trình ứng dụng (API-Application Programming Interface).
- **Socket** cho phép thiết lập các kênh giao tiếp mà hai đầu kênh được đánh dấu bởi hai cổng (port). Thông qua các cổng này một quá trình có thể nhận và gởi dữ liệu với các quá trình khác.



viettel
software services

Port Number

- Để có thể thực hiện các cuộc giao tiếp, một trong hai quá trình phải công bố **số hiệu cổng** (**port number**) của **socket** mà mình sử dụng. Mỗi cổng giao tiếp thể hiện một địa chỉ xác định trong hệ thống. Khi quá trình được gán một số hiệu cổng, nó có thể nhận dữ liệu gởi đến cổng này từ các quá trình khác. Quá trình còn lại cũng được yêu cầu tạo ra một **socket**.
- **Số cổng (port number)** được sử dụng để xác định duy nhất các **ứng dụng khác nhau**. Nó hoạt động như một điểm kết cuối giao tiếp giữa các ứng dụng.
- **Số cổng được kết hợp với địa chỉ IP** để giao tiếp giữa hai ứng dụng



Port Number

- Số hiệu cổng gán cho Socket phải duy nhất trên phạm vi máy tính đó, có giá trị trong khoảng từ 0 đến 65535 (16 bits). Trong đó, giá trị cổng:
- Từ 0-1023: là cổng hệ thống (common hay well-known ports), được dành riêng cho các quá trình của hệ thống.
- Từ 1024-49151: là cổng phải đăng ký (registered port). Các ứng dụng muốn sử dụng cổng này phải đăng ký với IANA (Internet Assigned Numbers Authority).
- Từ 49152-65535: là cổng dùng riêng hay cổng động (dynamic hay private port). Người sử dụng có thể dùng cho các ứng dụng của mình, không cần phải đăng ký.

viettel
software services

Port Number

- Một số cổng thường được sử dụng:
- 21: dịch vụ FTP
- 23: dịch vụ Telnet
- 25: dịch vụ Email (SMTP)
- 80: dịch vụ Web (HTTP)
- 110: dịch vụ Email (POP)

viettel
software services

IP Adress

- Ngoài số hiệu cổng, hai bên giao tiếp còn phải biết địa chỉ IP của nhau. Địa chỉ IP giúp phân biệt máy tính này với máy tính kia trên mạng TCP/IP. Trong khi số hiệu cổng dùng để phân biệt các quá trình khác nhau trên cùng một máy tính.
- Địa chỉ IP (IP Address) là một số duy nhất được gán cho một nút của mạng, ví dụ: 192.168.0.1. Nó bao gồm các số thập phân trong khoảng từ 0 đến 255.
- IP Address là một địa chỉ logic (logical address) có thể được thay đổi.

MAC Address

- **MAC** là viết tắt của từ **Media Access Control** là một thành phần (tầng) cung cấp các cơ chế đánh địa chỉ và điều khiển truy cập kênh giúp máy tính này có thể trao đổi hoặc truyền dữ liệu với máy tính khác
- Địa chỉ **MAC** là một dãy số 48-bit của phần cứng máy tính, được nhà sản xuất card mạng nhúng vào. Địa chỉ **MAC** được ví là **địa chỉ vật lý của thiết bị mạng** tương tự như việc muốn đi đến nhà nào cũng phải biết địa chỉ của nhà đó.

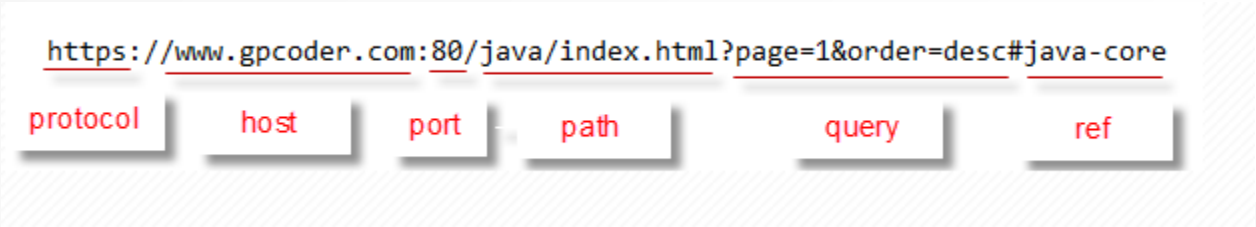
Cơ chế giao tiếp TCP và UDP

viettel
software services

Một số class quan trọng trong gói java.net

Class URL

- **URL** là viết tắt của **Uniform Resource Locator** và biểu diễn một tài nguyên trên **Web**, ví dụ như một trang **Web** hoặc thư mục **FTP**.
- Một URL có thể được phân chia thành các phần như sau:



viettel
software services

Class URL

- Class **java.net.Url** cung cấp các phương thức để truy cập các phần khác nhau của **URL** như sau:
- public String getPath() : trả về path của URL đó
- public String getQuery() : trả về phần query của URL đó
- public String getAuthority() : trả về authority của URL đó
- public int getPort() : trả về port của URL đó
- public int getDefaultPort() : trả về port mặc định cho protocol của URL đó
- public String getProtocol() : trả về protocol của URL đó
- public String getHost() : trả về host của URL đó
- public String getFile() : trả về filename của URL đó
- public String getRef() : trả về phần reference của URL đó
- public URLConnection openConnection() : mở một kết nối tới URL, cho phép một client giao tiếp với tài nguyên.

viettel
software services

Class URLConnection

- Phương thức **openConnection()** của lớp **java.net.Url** trả về một **java.net.URLConnection**. Đây là abstract class mà các lớp phụ của nó biểu diễn các kiểu kết nối URL khác nhau. Ví dụ:
- Nếu một **URL** mà protocol của nó là **HTTP**, thì phương thức openConnection() trả về một đối tượng **HttpURLConnection**.
- Nếu một **URL** mà biểu diễn một **JAR file**, thì phương thức **openConnection()** trả về một đối tượng **JarURLConnection**.

viettel
software services

Các phương thức quan trọng của URLConnection

- Object getContent() : lấy nội dung của URL connection này
- Object getContent(Class[] classes) : lấy nội dung của URL connection này
- String getContentEncoding() : trả về giá trị của trường content-encoding header
- int getContentLength() : trả về giá trị của trường content-length header
- String getContentType() : trả về giá trị của trường content-type header
- int getLastModified() : trả về giá trị của trường last-modified header
- long getExpiration() : trả về giá trị của trường expires header
- long getIfModifiedSince() : trả về giá trị của trường ifModifiedSince của đối tượng này

viettel
software services

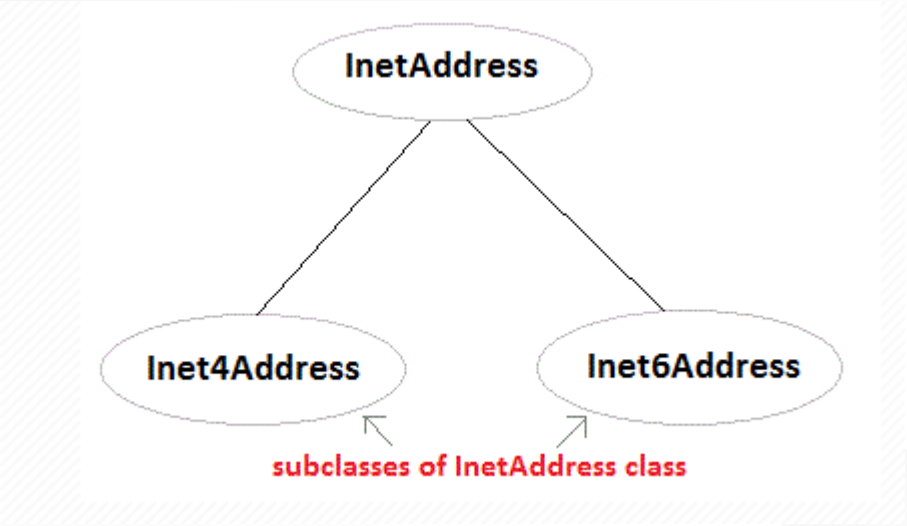
Các phương thức quan trọng của URLConnection

- public void setDoInput(boolean input) : Truyền là true để biểu thị rằng connection sẽ được sử dụng cho input. Giá trị mặc định là true bởi vì Client đọc từ một URLConnection.
- public void setDoOutput(boolean output) : Truyền là true để biểu thị rằng connection sẽ được sử dụng cho output. Giá trị mặc định là false bởi vì nhiều kiểu URL không hỗ trợ để được ghi trên đó.
- public InputStream getInputStream() throws IOException : trả về InputStream của URL connection để đọc từ nguồn.
- public OutputStream getOutputStream() throws IOException : trả về OutputStream của URL connection để ghi từ nguồn.
- public URL getURL() : trả về URL mà đối tượng URLConnection này được kết nối tới

viettel
software services

Class InetAddress

- Class Java **InetAddress** đại diện cho một địa chỉ **IP**. Class **java.net.InetAddress** cung cấp các phương thức để lấy **IP** của một **host** bất kỳ
- **InetAddress** có thể xử lý cả địa chỉ **IPv4 và IPv6**. **InetAddress không có constructor**, để tạo một đối tượng **InetAddress** , bạn phải sử dụng các phương thức **Factory**.



Các Factory của class InetAddress

- **public static InetAddress getLocalHost ()** : nó trả về thể hiện của InetAdddress chứa tên và địa chỉ localhost.
- **public static InetAddress getByName (String hostname)** : nó trả về thể hiện của InetAddress của hostname gồm IP và tên.
- **public static InetAddress[ ] getAllByName (String hostname)** : nó trả về danh sách các InetAddress của hostname gồm IP và tên.

viettel
software services

Một số phương thức của class InetAddress

- **public String getHostName()**: nó trả về tên máy chủ (host) của địa chỉ IP.
- **public String getHostAddress()**: nó trả về địa chỉ IP ở định dạng chuỗi.

Lập trình Socket

- Socket cung cấp cơ chế truyền thông giữa hai máy tính sử dụng TCP. Một máy khách tạo ra socket để kết nối đến với máy chủ.
- Lớp java.net.Socket đại diện cho một socket, và lớp java.net.ServerSocket cung cấp một cơ chế cho chương trình máy chủ để lắng nghe khách hàng và thiết lập kết nối với chúng
- **Các bước thiết lập kết nối TCP giữa hai máy tính sử dụng socket:**
- Máy chủ khởi tạo một đối tượng **ServerSocket** với một cổng giao tiếp (port).
- Máy chủ gọi phương thức accept() của lớp **ServerSocket**. Phương pháp này đợi cho đến khi một máy khách kết nối đến máy chủ trên cổng đã cho.

Lập trình Socket

- Trong khi **server** đang chờ đợi, một **client** khởi tạo đối tượng Socket, xác định server (IP hoặc domain) và số cổng để kết nối.
- Đối tượng socket của **client** cố gắng kết nối **client** tới máy chủ đã chỉ định và số cổng. Nếu truyền thông được thiết lập, máy khách bây giờ có một đối tượng socket có khả năng giao tiếp với server.
- Ở phía **server**, phương thức **accept()** trả về một tham chiếu đến một socket mới trên máy chủ được kết nối với socket của **client**.
- Sau khi kết nối được thiết lập, **server** và **client** có thể truyền và nhận thông tin thông qua **OutputStream** và **InputStream**.

viettel
software services

Lập trình DatagramSocket

- DatagramSocket cung cấp cơ chế truyền thông giữa hai máy tính sử dụng UDP.
- Với **UDP socket**, chúng ta không cần thiết lập liên kết “handshaking” giữa **client** và **server** trước khi truyền thông điệp, thông điệp được gửi đi một cách độc lập. Bên gửi sẽ chỉ rõ địa chỉ IP và số hiệu cổng (port number) của bên nhận. Sau khi thông điệp được truyền đến bên nhận, bên nhận có thể dựa vào địa chỉ IP và số hiệu cổng tương ứng của bên gửi được gắn trên gói tin để gửi lại response.

viettel
software services

Lập trình DatagramSocket

- Với UDP ta **không cần thiết lập liên kết 2 chiều giữa client và server**.
- **Về phía client:**
- Cần biết được địa chỉ IP cũng như số hiệu cổng của phía Server
- Mỗi gói tin gửi đi cần có địa chỉ IP, số hiệu cổng của nơi gửi đến, nơi gửi đi
- Gửi gói tin cho server
- **Về phía server:** nhận gói tin, trích xuất địa chỉ IP, số hiệu cổng của client, xử lý và respond gói tin lại cho client.

viettel
software services

BÀI 2

Design Pattern

viettel
software services

Design Pattern là gì

- **Design Pattern** là một kỹ thuật trong lập trình hướng đối tượng, nó khá quan trọng và mọi lập trình viên muốn giỏi đều phải biết. Được sử dụng thường xuyên trong các ngôn ngữ OOP. Nó sẽ cung cấp cho các **"mẫu thiết kế"**, giải pháp để giải quyết các vấn đề chung, thường gặp trong lập trình. Các vấn đề mà bạn gặp phải có thể bạn sẽ tự nghĩ ra cách giải quyết nhưng có thể nó chưa phải là tối ưu**. Design Pattern** giúp bạn giải quyết vấn đề một cách tối ưu nhất, cung cấp cho bạn các giải pháp trong lập trình OOP.
- **Design Patterns không phải là ngôn ngữ** cụ thể nào cả. Nó có thể thực hiện được ở phần lớn các ngôn ngữ lập trình, chẳng hạn như Java, C#, thậm chí là Javascript hay bất kỳ ngôn ngữ lập trình nào khác.

viettel
software services

Tại sao phải sử dụng Design Pattern

- **Design Pattern** giúp bạn tái sử dụng mã lệnh và dễ dàng mở rộng.
- Nó là tập hơn những giải pháp đã được tối ưu hóa, đã được kiểm chứng để giải quyết các vấn đề trong software engineering. Vậy khi bạn gặp bất kỳ khó khăn gì, **design patterns là kim chỉ nam giúp bạn giải quyết vấn đề** thay vì tự tìm kiếm giải pháp cho một vấn đề đã được chứng minh.
- **Design pattern cung cấp giải pháp ở dạng tổng quát**, giúp tăng tốc độ phát triển phần mềm bằng cách đưa ra các mô hình test, mô hình phát triển đã qua kiểm nghiệm.
- **Dùng lại các design pattern giúp tránh được các vấn đề tiềm ẩn** có thể gây ra những lỗi lớn, dễ dàng nâng cấp, bảo trì về sau.
- **Giúp cho các lập trình viên có thể hiểu code của người khác** 1 cách nhanh chóng (có thể hiểu là tính communicate). Mọi thành viên trong team có thể dễ dàng trao đổi với nhau để cùng xây dựng dự án mà không mất quá nhiều thời gian.

viettel
software services

Phân loại Design Pattern

- **Hệ thống các mẫu Design pattern hiện có 23 mẫu được định nghĩa trong cuốn "Design patterns Elements of Reusable Object Oriented Software" và được chia thành 3 nhóm:**
- **Creational Pattern (nhóm khởi tạo – 5 mẫu) gồm**: **Factory Method, Abstract Factory, Builder, Prototype, Singleton**. Những Design pattern loại này cung cấp một giải pháp để tạo ra các object và che giấu được logic của việc tạo ra nó, thay vì tạo ra object một cách trực tiếp bằng cách sử dụng method new. Điều này giúp cho chương trình trở nên mềm dẻo hơn trong việc quyết định object nào cần được tạo ra trong những tình huống được đưa ra.
- **Structural Pattern (nhóm cấu trúc – 7 mẫu) gồm**: **Adapter, Bridge, Composite, Decorator, Facade, Flyweight và Proxy**. Những Design pattern loại này liên quan tới class và các thành phần của object. Nó dùng để thiết lập, định nghĩa quan hệ giữa các đối tượng.
- **Behavioral Pattern (nhóm tương tác/ hành vi – 11 mẫu) gồm**: **Interpreter, Template Method, Chain of Responsibility, Command, Iterator, Mediator, Memento, Observer, State, Strategy và Visitor**. Nhóm này dùng trong thực hiện các hành vi của đối tượng, sự giao tiếp giữa các object với nhau.

viettel
software services

XIN CẢM ƠN