**HTML/CSS/JS/TS**

**HTML (viết tắt của từ HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Nó có thể được trợ giúp bởi các công nghệ như CSS và các ngôn ngữ kịch bản giống như JavaScript.**

1. DOM trong HTML là gì ?

DOM (The Document Object Model) là một giao diện lập trình cho các document HTML. Nó đại diện cho trang để các chương trình có thể thay đổi cấu trúc, kiểu và nội dung của tài liệu. DOM đại diện cho tài liệu dưới dạng các nút và đối tượng.

1. Điểm khác nhau của thẻ <span> và <div> là gì?

span là phần tử nội tuyến (inline).

div là phần tử khối (block).

Bạn nên dùng các thẻ div để chứa (bọc) các phần của tài liệu, còn các thẻ span để chứa các phần nhỏ của văn bản, hình ảnh, v…v…

1. Các thẻ Meta là gì?

Thẻ meta là các đoạn văn bản mô tả nội dung của trang; các thẻ meta không xuất hiện trên chính trang mà chỉ xuất hiện trong mã nguồn của trang.

Các thẻ meta về cơ bản chính là các mô tả nội dung nhỏ giúp cho các công cụ tìm kiếm biết nội dung của một trang web.

1. Điểm khác nhau của ID selector và Class selector trong CSS?

ID là duy nhất. Mỗi phần tử chỉ có thể có một ID và trang HTML chỉ có thể có một phần tử có ID đó

Class thì ngược lại. Bạn có thể sử dụng cùng một class trên nhiều phần tử và một phần tử có thể có nhiều class.

1. Làm thế nào để áp dụng các quy tắc CSS cụ thể cho một media?

Quy tắc @media được sử dụng trong các truy vấn media để áp dụng các kiểu khác nhau cho các loại/ thiết bị media khác nhau.

1. Trong CSS Pseudo-class là gì?

Một Pseudo class trong CSS được dùng để xác định trạng thái đặc biệt của một phần tử. Nó có thể được kết hợp với một bộ chọn CSS (selector) để thêm hiệu ứng cho các phần tử hiện có dựa trên trạng thái của chúng.

1. Điểm khác biệt giữa relative, fixed, absolute và statically positioned element là gì?

Relative là phần tử được định vị tương đối so với vị trí bình thường của nó.

Fixed là phần tử được định vị liên quan đến cửa sổ trình duyệt.

Absolute là phần tử được đặt hoàn toàn vào vị trí gốc đầu tiên của nó.

Static đây là giá trị mặc định, tất cả các phần tử theo thứ tự khi chúng xuất hiện trong tài liệu.

1. PUT và POST khác nhau như thế nào?

PUT: Thay thế tài nguyên đích bằng payload yêu cầu. Có thể được sử dụng để cập nhật hoặc tạo một tài nguyên mới.

POST: Thực hiện xử lý tài nguyên cụ thể trên payload. Có thể được sử dụng cho các hành động khác nhau bao gồm tạo tài nguyên mới, tải lên tệp hoặc gửi biểu mẫu web.

1. Long-Polling, Websockets và Server-Sent Events khác nhau thế nào?

Long-Polling mở một yêu cầu HTTP và vẫn mở cho đến khi nhận được bản cập nhật. Khi nhận được bản cập nhật, một yêu cầu mới sẽ được mở ngay lập tức khi vẫn đang chờ bản cập nhật tiếp theo.

Websocket Giao thức WebSocket cho phép giao tiếp liên tục, hai chiều giữa máy chủ và máy khách. Đối với thử nghiệm này, Primus được sử dụng để tóm tắt nhiều hoạt động của giao thức.

Server-Sent Event Các sự kiện do máy chủ gửi dựa trên kết nối HTTP lâu dài, nơi các bản cập nhật liên tục được gửi đến máy khách.

1. Sự khác biệt giữa cookies, session storage, and local storage?

Local storage như tên gọi (bộ nhớ cục bộ) là bộ nhớ của trình duyệt, có khả năng lưu trữ vô thời hạn lên tới 10MB, session storage (bộ nhớ phiên) cũng như vậy, nhưng như tên gọi của nó, bộ nhớ này dựa trên phiên và sẽ bị xóa sau khi đóng trình duyệt của bạn. Bộ nhớ phiên lưu trữ được ít hơn, lên đến 5MB.

Cookie là dữ liệu rất nhỏ lưu trữ trong trình duyệt, có thể lưu trữ đến 4KB và có thể được truy cập thông qua máy chủ hoặc trình duyệt.

1. CORS là gì?

CORS là viết tắt của Cross-Origin Resource Sharing.

CORS là một cơ chế trình duyệt cho phép truy cập có kiểm soát vào các tài nguyên nằm bên ngoài một miền nhất định. Nó mở rộng và thêm tính linh hoạt cho chính sách cùng nguồn gốc.

1. Promise là gì?

Promise được dùng để xử lý các hoạt động bất đồng bộ. Chúng có thể xử lý nhiều hoạt động bất đồng bộ một cách dễ dàng và cung cấp khả năng xử lý lỗi tốt hơn so với callback và event.

1. Các state khác nhau trong promise?

Một promise có 3 state:

fullfilled: hoàn thành

rejected: lỗi/ không thể hoàn thành

pending: đang chờ

1. Hoisting trong JavaScript là gì?

Hoisting là một thuật ngữ được dùng để mô tả sự di chuyển của các biến và hàm lên đầu phạm vi của chúng, ở đấy chúng ta có thể xác định được biến hoặc hàm đó.

1. Các giá trị falsy trong JavaScript là gì?

Là giá trị khi được chuyển đổi thành boolean sẽ trở thành sai.

''

0

null

undefined

NaN

false

-0

0n

1. SSR và CSR khác nhau như thế nào?

SSR – Server Side Rendering là việc thực hiện Render ở phía máy chủ. Bạn có thể hiểu một cách đơn giản là khi truy cập vào web bất kỳ thì nó sẽ gửi yêu cầu về máy chủ, sau đó máy chủ sẽ trả toàn bộ mã HTML và CSS về hiển thị trên trình duyệt cho người dùng.

Còn CSR – Client Side Rendering là thực hiện Render ở phía người dùng, khách hàng. Tức là khi bạn truy cập vào web bất kỳ, nó sẽ gửi yêu cầu lên máy chủ và máy chủ sẽ trả về một thẻ div rỗng và một số file Javascript. Sau đó nó mới thực thị và hiển thị trên giao diện của người dùng.

1. Bạn hiểu CSS là gì?

CSS – Cascading Style Sheet là một ngôn ngữ tạo kiểu phổ biến dành cho HTML khi thiết kế website. Nó có thể sử dụng với bất kỳ tài liệu XML nào, từ thuần tuý cho đến SVG và XUL.

1. CSS framework là như thế nào?

Một CSS framework về bản chất sẽ bao gồm 1 số bảng định kiểu CSS sẵn giúp nhà phát triển, thiết kế web có thể sử dụng. Nó được dùng cho các chức năng thiết kế website tiêu chuẩn như: Bố cục, phông chữ, màu sắc, thanh điều hướng,… Thông qua CSS framework người dùng chỉ cần viết code HTML với các lớp, cấu trúc, cùng ID chính xác cho một web. Bởi nó đã tích hợp sẵn các lớp trong các phần tử trang web thông dụng hiện nay

1. Bạn hãy chỉ một vài cách khác nhau để tích hợp CSS vào trang HTML?

Cách 1 – Tích hợp CSS bằng cách dùng style-tags trong phần head của trang HTML. Cách 2 – Tích hợp style bằng cách dùng inline-styling. Cách 3 – Viết CSS trong tập tin riêng, sau đó thêm nó vào HTML qua thẻ liên kết.

1. BEM là gì? Tại sao cần dùng đến nó?

BEM viết tắt từ Block, Element và Modifier là quy ước đặt tên CSS cho các lớp để chúng dễ đọc và rõ ràng hơn. Mục đích dùng đến BEM là viết các khối CSS độc lập và có thể dùng lại chúng cho các dự án khác nhau.

1. Khi dùng External Style Sheets có ưu nhược điểm là gì?

Ưu điểm gồm: Sử dụng External Style Sheets giúp kiểm soát Style của một số documents từ trang web; Có thể tạo nhiều lớp với nhiều phần tử HTML và nhiều Documents; Đối với các trường hợp kiểu phức tạp, nó sẽ được dùng để chọn chiến lược và nhóm sử dụng. Nhược điểm gồm: Tải xuống bổ sung dự kiện có thông tin kiểu style khi nhập vào documents; Để hiển thị documents, các mẫu bên ngoài sẽ xếp chồng lên nhau; Không thực tế đối với small style definitions

1. Bạn hãy nêu một số thao tác DOM cơ bản mà bạn biết?

getElementById(); getElementsByClassName(); querySelector(); addEventListener(); removeEventListener(); appendChild().

1. Sự khác biệt của var, const và let là gì?

Khi bạn muốn biến có sự thay đổi thì khuyến khích sử dụng let. Còn nếu không muốn thay đổi thì nên dùng const

1. Truthy, Falsy là gì?

Truthy là những giá trị bị ép kiểu về boolean sẽ khiến nó bằng true. Còn Falsy là những gì trị khi ép kiểu về boolean cho nó sẽ bằng false. Ví dụ như: Falsy: 0, “” , null, undefined, NaN , false. Truthy: là các kiểu còn lại.

1. Tham trị, tham chiếu chứa những giá trị và địa chỉ nào?

Tham trị chứa các giá trị như: Number, String, Boolean. Còn Tham chiếu chứa các địa chỉ là: Array, Object.

1. Bạn hãy cho biết thẻ là gì?

“ <!DOCTYPE html> sẽ được khai báo ở đầu một file HTML, nó được dùng cho trình duyệt web khi lập trình sử dụng phiên bản ngôn ngữ đánh dấu.”

1. Thế nào là HTML semantic?

HTML semantic là việc dùng thẻ thích hợp cho ý nghĩa của nó. Hiểu một cách đơn giản là dùng các phần tử có ý nghĩa như: form, article và table thay cho việc dùng Span và div.

1. HTML5 là gì?

HTML5 là phiên bản thử 5 của ngôn ngữ lập trình HTML. Hiện nay phiên bản này được dùng rất phổ biến. Sử dụng nó có thể một số thẻ mới như: header, footer, article, nav, section, audio, video….

1. Iframe là gì và nó hoạt động như thế nào?

Iframe là một văn bản HTML (HTML document) được nhúng vào một trang html khác.

1. Mục đích của thuộc tính alt trong thẻ img?

Thuộc tính alt cung cấp thông tin về hình ảnh trong trường hợp người dùng không xem được ảnh (sai đường dẫn hoặc ảnh không tìm thấy).

Thuộc tính alt thường được dùng để mô tả thông tin hình ảnh , một số trường hợp thuộc tính alt được dùng trong mục đích trang trí, một số trường hợp thuộc tính alt có thể bị bỏ trống.

1. Liệt kê 3 cách để định nghĩa 1 color trong html?

Hex (color: #FFFFFF)

Name (color: red)

RGB(0,0,225)

1. Giải thích ý nghĩa các CSS selectors dưới đây?

“div, p”: chọn tất cả các thẻ div và tất cả các thẻ p.

“div p”: chọn tất cả các thẻ p nằm trong thẻ div.

“div > p”: chọn tất cả thẻ p có thẻ cha trực tiếp là thẻ div.

“div + p”: chọn tất cả thẻ p được đặt trực tiếp sau thẻ div.

“div ~ p”: chọn tất cả thẻ p đứng trước nó là thẻ div.

1. Giải thích các đơn vị độ dài trong css dưới đây?

Cm: centimeters.

Em: elements, ví dụ 2 em = 2 lần font-size của thẻ đó.

In: inches. - Mm: millimeters.

Pc: picas ( 1pc = 12pt = 1/6 in)

Pt: points ( 1pt = 1/72 in) - Px: pixels ( 1/96 in)

1. CSS có phân biệt hoa thường không?

Không phân biệt.

1. Thuộc tính float trong CSS thường dùng để làm gì?

Dùng để đặt các thẻ HTML theo hướng nằm ngang, float thường dùng 2 giá trị left và right.

1. Margin và padding khác nhau thế nào?

Hiểu đơn giản thì margin căn lề ngoài và padding căn lề trong của thẻ đó.

1. Display:inline và display:block khác nhau thể nào?

Block sẽ làm thẻ đó có space trước và sau nó, và space đó sẽ chiếm hết các width có sẵn. Inline thì chỉ xài đủ width yêu cầu.

1. Làm sao để add 1 comments trong CSS?

Dùng cặp dấu /\* comments \*/.

1. Thuộc tính “opacity” được dùng làm gì?

Nó thể hiện như thuộc tính visibility và nó mờ dần từ 1-> 0 ( 1 -> 0.9 -> 0.8 …)

1. Giải thích CSS box model và các thành phần bố cục trong nó?

CSS box model là một mô hình bố cục hình chữ nhật cho các phần tử HTML bao gồm những điều sau:

Content - Nội dung của box, nơi text và images xuất hiện.

Padding - Vùng trong suốt bao quanh content (tức là khoảng cách giữa border và content).

Border - Đường viền bao quanh padding (nếu có) và content.

Margin - Một vùng trong suốt bao quanh border (tức là khoảng cách giữa border và bất kỳ phần tử lân cận nào).

1. Java Script là gì?

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra, javascript được sử dụng rộng rãi trong các ứng dụng website. Nó là một ngôn ngữ thông dịch

1. Các kiểu dữ liệu trong javascript

Có 6 kiểu: String, Number, Object, Underfined, Boolean, Array, Null

1. Hàm delete có chức năng gì?

Loại bỏ một thuộc tính ra khỏi object

1. this trong javascript là gì?

Từ khoá this dùng để chỉ đối tượng từ nơi nó đuọcw gọi

1. Sự khác nhau giữa undefined và null trong javascript?

Khi tạo ra một biến mà khong gán giá trị thì nó undefined. còn null là một object

1. Thay đổi style/class của element?

Sử dụng methods document trong javascript.

1. Tại sao Math.max() nhỏ hơn Math.min()?

Vì không có tham số truyền vào: Math.max() trả về Infinity còn Math.min() -Infinity

1. Phân biệt giữa Function Declaration và Function Expression

Function Decclaraton sử dụng từ khoá function rồi đến tên hàm, Còn Function Expression sẽ bắt đầu từ từ khoá var, const, let

1. Hàm Array.splice() và hàm Array.slice() khác nhau như thế nào?

Hàm Array.splice() sẽ thay thế một hoặc một số phần tử mảng bằng mọt số phần từ khác.

Hàm Array.slice() sẽ trích xuất một số phần từ của mảng, vị trí bắt đầu và kết thúc

1. Spread Operator trong javascript

Spread opertor là một biểu thức mở rộng giúp gộp các phần từ vào mtooj cách viếc ngắn gọn hơn. các viêts này thử hiện bằng giấy … áp dụng cho mảng và object

1. Tại sao 0.1 + 0.2 không bằng 0.3

Liên quan tơi việc javascript lưu trữ float dươi dạng nhị phân. Cách khắc phhuc dùng toFixed()

1. Toán tử typeof là gì?

JavaScript cung cấp một toán tử typeof có thể kiểm tra một giá trị

1. Cho biết output của 2 dòng lệnh dưới?

Console.log(1<2<3) -> true

Console.log(3>2>1) -> false

1. Khác nhau giữa null và undefined?

Undefined có nghĩa là không xác định. Trong javascript, khi bạn khai báo một biến nhưng chưa gán giá trị cho nó, giá trị của biến đó sẽ là undefined.

Null là một loại object, tương tự như các loại object khác như 'string', 'number',...

1. Khác nhau giữa 2 hàm document.ready và body.onload?

Body.onload chỉ được gọi khi DOM và các resources kèm theo như images được load xong. Còn document.ready được khi DOM được load xong.

Có thể có nhiều document.ready nhưng chỉ có 1 body.onload.

1. Cho biết output của lệnh: Console.log(typeof undefined == typeof NULL)

Kết quả: true

1. Cho biết output của lệnh dưới đây: console.log(typeof typeof 1)

Kết quả: string

1. "This" trong javascript được dùng để làm gì?

Từ khóa "this" trong javascript tham chiếu đến đối tượng mà nó thuộc về. "This" có các giá trị khác nhau phụ thuộc vào nơi nó được sử dụng.

1. Hàm anonymous là gì và khi nào nên sử dụng nó?

Anonymous functions hay còn gọi là hàm ẩn danh, là một hàm được sinh ra đúng vào thời điểm chạy của chương trình.

1. Kể các cách khai báo 1 biến trong Javascript?

Có 3 cách : var, let, const

1. Kiểu object là gì?

Kiểu object đề cập đến một giá trị ghép nơi bạn có thể đặt các thuộc tính (các vị trí được đặt tên) mà mỗi thuộc tính là một giá trị riêng có thể là bất kỳ kiểu nào.

1. Làm sao để clone một mảng

Sử dùng hàm slice, JSON.stringify() và JSON.parse(), toán tử [...] (spread operator) trong ES6,

Object.assign:

1. Giải thích về Arrays trong JavaScript?

Một array là một đối tượng chứa các giá trị (thuộc bất kỳ kiểu nào) ở các vị trí được lập index

1. Kể tên 3 loại dữ liệu cơ bản của javascript

“number”

“string”

“boolean”

1. Điểm khác nhau giữa GET và POST?

GET:

Có thể bị cache.

Nằm trong history của browser.

Có thể bị bookmarked

Dữ liệu truyền đi sẽ hiện trên url và chúng bị giới hạn (độ dài tối đa của URL là 2048).

Hạn chế về loại dữ liệu: chỉ có ký tự ASCII được cho phép

POST:

Dùng để submit data tới servers.

Không bị cache

Không nằm trong history của browser.

Không bị bookmarked

Dữ liệu truyền đi nằm trong body của request.

Không bị giới hạn dữ liệu truyền đi.

Không giới hạn loại dữ liệu truyền đi.

1. Giải thích về Equality trong JavaScript?

JavaScript có 2 loại so sánh:

Strict comparison (ví dụ: ===) kiểm tra giá trị bằng nhau mà không tự động ép kiểu.

Abstract comparison (ví dụ: ==) kiểm tra giá trị bằng nhau có tự động ép kiểu.

1. Scope của 1 biến trong javascript là gì? Có bao nhiêu loại scope?

Scope của 1 biến là phạm vi trong chương trình của bạn mà biến đó được định nghĩa.

Trong javascript có 3 loại scope:

**Global scope**: có phạm vị hoạt động ở bất kỳ trong mã javascript của bạn.

**Function scope**: có phạm vi hoạt động trong function mà bạn khai báo biến đó.

**Block scope**: có phạm vị trong cặp dấu {} mà bạn khai báo biến đó. (ES6)

1. Lập trình bất đồng bộ trong javascript là gì?

Lập trình bất đồng bộ (async) là 1 phần quan trọng trong javascript. Cách tiếp cận phổ biến là sử dụng các callback.

1. Closure trong javascript là gì? Cho ví dụ

Trong javascript, closure là một hàm tham chiếu các biến có phạm vi ngoài phạm vi của hàm đó.

Nó cho phép bạn truy cập các biến hoặc tham số có phạm vị ngoài phạm vị hàm đó.

Để sử dụng closure, đơn giản là bạn khai báo một function bên trong một function khác và return nó ra.

**Typescript**

1. Làm thế nào để gọi hàm constructor của lớp cơ sở từ lớp con trong TypeScript?

Chúng ta có thể gọi hàm constructor của lớp cơ sở bằng cách sử dụng super().

1. TypeScript là gì và tại sao sử dụng nó thay cho JavaScript?

TypeScript là một siêu tập hợp JavaScript chủ yếu cung cấp kiểu gõ tĩnh (static-typing), các lớp và interface tùy chọn. Một trong những lợi ích lớn là cho phép IDE cung cấp môi trường phong phú hơn để phát hiện các lỗi phổ biến khi bạn gõ mã.

1. TypeScript là gì?

- TypeScript là một ngôn ngữ được định hướng mạnh mẽ, hướng đối tượng và biên dịch.

- Ngôn ngữ này được phát triển và duy trì bởi Microsoft. Nó được thiết kế bởi "Anders Hejlsberg" tại Microsoft. Nó là một phần mở rộng của JavaScript.

- TypeScript là JavaScript và cũng có một số tính năng bổ sung như static typing và lập trình hướng đối tượng dựa trên lớp, tự động gán các tham số constructor và các giá trị null được gán và vân vân.

1. Mô tả cách tạo 1 class trong Typescript?

1 class là một định nghĩa mẫu của các phương thức và các biến trong một kiểu đối tượng cụ thể. Đó là chương trình, mã và mẫu có thể mở rộng để tạo các đối tượng.

Một TypeScript là JavaScript hướng đối tượng và nó cũng hỗ trợ các tính năng lập trình hướng đối tượng như các class, interface, v.v.

1 lớp có các Public, Private, Protected and Read-only modifiers theo mặc định.

1 lớp có thể chứa:

* Trường (Fields),
* Constructors.
* Hàm (Functions).

1. Named function trong typescript là gì?

Named function giống với hàm JavaScript và chỉ có một sự khác biệt - chúng ta phải khai báo kiểu trên các tham số đã truyền.

1. Biến trong Typescript là gì và cách khai báo biến như thế nào?

Biến chỉ đơn giản là tên của vị trí lưu trữ và tất cả các biến phải được xác định với các tên duy nhất và các tên duy nhất này được gọi là mã định danh.

1. Public, Private, Protected và Readonly modifiers trong Typescript là gì?

Public Modifier – Public là mặc đinh. Nó được truy xuất ở mọi nơi.

Private Modifier - Khi sử dụng Private modifier, chúng ta không thể truy xuất chúng từ ngoài class của nó.

Protected Modifier - Protected modifier giống như Private nhưng khác 1 điểm là nó có thể được truy xuất bởi instance của lớp kế thừa.

Readonly Modifier - Các thuộc tính Read-only phải đợi khởi tạo tại lúc khai báo chúng hoặc trong constructor.

1. Làm thế nào để thực hiện thêm chuỗi vào chuỗi trong TypeScript?

Bạn có thể sử dụng các vòng lặp mẫu,

Các ký tự mẫu được phân tách bằng dấu gạch ngược (`).

1. Tại sao nên sử dụng Typescript

Hỗ trợ lập trình hướng đối tượng.

Typescript thêm tính năng Static Typing vào JavaScript. Có tính năng Static Typing giúp dễ dàng phát triển và duy trì các ứng dụng phức tạp hơn.

Angular2 sử dụng TypeScript rất nhiều để đơn giản hóa mối quan hệ giữa các thành phần khác nhau và cách khung được xây dựng sẵn.

Cung cấp một hệ thống kiểu tùy chọn cho JavaScript.

Cung cấp các tính năng được lên kế hoạch từ các phiên bản JavaScript trong tương lai cho các công cụ JavaScript hiện tại.

Hỗ trợ định nghĩa kiểu.

1. Rest parameters function là gì?

Generics có thể tạo một component hoặc function để hoạt động trên nhiều kiểu thay vì một kiểu duy nhất.

1. Chúng ta có cần biên dịch các tệp TypeScript không và tại sao?

Có. Typecript chỉ là một ngôn ngữ các trình duyệt không thể diễn giải nó. Chuyển đổi từ TypeScript sang JavaScript được gọi là biên dịch. Biên dịch không có nghĩa là mã nhị phân được tạo trong trường hợp này. Đối với loại bản dịch này, thuật ngữ chuyển đổi cũng được sử dụng thay vì biên dịch.

1. Default export trong Typescript là gì? Tại sao sử dụng chúng?

Mỗi module có thể tùy chọn xuất default export và default export hoạt động với từ khóa mặc định và chúng ta chỉ có thể sử dụng một default export cho mỗi module.

1. Static Typing là gì?

Mỗi biến (variable) có một kiểu cố định liên kết với nó. Kiểu của biến được kiểm tra lúc compiled-time và trình biên dịch yêu cầu ta phải khai báo rõ kiểu của biến trước khi sử dụng.

1. Interface trong Typescript là gì?

Interface trong typescript cũng tương tự như trong các ngôn ngữ lập trình hướng đối tượng khác.

1. Anonymous function (hàm ẩn danh) là gì?

Một hàm ẩn danh là một hàm được khai báo mà không có bất kỳ định danh được đặt tên nào để tham chiếu đến nó.

1. Method Overriding là gì?

Trong Method Overriding, tái định nghĩa các method của lớp cha trong lớp kế thừa hoặc lớp con.

**Angular**

1. Angular là gì

Angular là một framework phát triển ứng dụng web phía client được xây dựng bằng TypeScript, một ngôn ngữ lập trình được biên dịch thành JavaScript. Angular được phát triển bởi nhóm Angular của Google và được sử dụng rộng rãi để xây dựng các ứng dụng web động và tương tác.

1. Dùng hàm nào trong lifecycle hook của Angular để bắt sự kiện page đã render xong?

ngAfterViewChecked

1. let khác gì var?

scope: var là function scope, let là block scope

hoisting: biến var sẽ được hoisting lên đầu hàm trong khi let thì không

global object property: ở top level, var tạo ra propery của global object (window) còn let thì không.

redeclaration: let không cho phép khai báo lại biến như var

1. Truyền dữ liệu giữa các component trong Angular như nào?

Từ parent component xuống child component

input binding

dùng @ViewChild()

Từ child component lên parent component

dùng event binding a.k.a parent lắng nghe child event

dùng local variable a.k.a template reference

Từ 2 sibling component

trao đổi thông qua parent component

Giữa 2 component không có mối liên hệ trực tiếp

dùng service (đặt service này trong module có chứa cả 2 component)

1. HTTP Interceptor?

Dùng để thay đổi request phản hồi trước khi chuyển nó sang interceptor kế tiếp, hoặc thay đổi luồng dữ liệu phản hồi với next.handle().

Đối với HTTP request, tham số truyền vào bao gồm (req: HttpRequest<any>, next: HttpHandler) và trả về Observable<HttpEvent<any>>, trong đó next là interceptor kế tiếp trong chuỗi interceptor.

Một số trường hợp sử dụng phổ biến: authentication/authorization,

1. Arrow function khác gì so với function bình thường?

Arrow function là một cách viết ngắn gọn và có thể thay thế function bình thường trong một số trường hợp.

Một số hạn chế của Arrow function:

Không có binding tới this hay super, và do vậy không nên dùng cho phương thức của object.

Không hỗ trợ các từ khóa arguments hay new.target

Không thể dùng với các hàm call, apply và bind (các hàm dùng để thiết lập scope)

Không thể dùng làm hàm constructor

Không dùng yield trong thân hàm được

1. ES6 khác gì so với ES5?

S6 có thêm các tính năng mới

Arrow function

Object destruction, Spread operator

let, const

lặp qua iterable object với for…of

symbol data type

class

import

1. Bạn đã dùng Framework để quản lý trạng thái (vd Redux) chưa?

ngrx, ngxs

1. Làm sao Angular phát hiện được sự thay đổi giá trị binding?

change detection: so sánh giá trị hiện tại với giá trị trước đó, dùng thư viện zone.js

**Java Core & OOP**

1. **Các kiểu biến nào một lớp có thể chứa?**

Một lớp có thể gồm biến local, biến instance, và biến lớp.

1. JVM là gì? Tại sao Java được gọi là “Platform Independent Programming Language”?

Máy ảo Java (JVM) là một máy ảo tiến trình có thể thực thi Java bytecode.

1. java.util.regex gồm các lớp nào?

Ba lớp: lớp Pattern, lớp Matcher và lớp PatternSyntaxException.

1. Liệt kê một số từ khóa trong Java (không giống từ khóa trong C, C++)?

Một số từ khóa trong Java là import, super, finally, …

1. Biến lớp là gì?

Đây là các biến được khai báo với một lớp, bên ngoài bất cứ phương thức nào, với từ khóa static

1. Liệt kê 5 đặc điểm bất kỳ của Java?

Một số đặc điểm của Java là:

Hướng đối tượng

Độc lập nền tảng

Thông dịch

Đa luồng (Multi-thread)

Robust (mạnh mẽ)

Portable (khả chuyển)

1. Giải thích Runtime Exception?

Nó là một Exception mà có thể được tránh bởi lập trình viên. Trái ngược với Checked Exception, các Runtime Exception bị bỏ qua tại thời điểm biên dịch.

1. Tại sao Java được xem như là Dynamic?

Nó được thiết kế để thích nghi với môi trường phát triển. Các chương trình Java có thể mang một lượng lớn thông tin run-time có thể được sử dụng để kiểm tra và xử lý các truy cập tới các đối tượng tại runtime.

1. Bạn hiểu gì về Đối tượng?

Đối tượng là một thực thể tại runtime, trạng thái của nó được lưu trữ trong các trường và hành vi được thể hiện thông qua các phương thức. Các phương thức vận hành trên trạng thái nội tại của một đối tượng và đóng vai trò như là kỹ thuật sơ cấp để giao tiếp giữa các đối tượng với nhau.

1. Lớp Exception có các lớp con nào?

Lớp Exception có hai lớp con chính là: lớp IOException và lớp RuntimeException.

1. Sự khác nhau giữa hai lớp StringBuffer và StringBuilder?

Sử dụng StringBuilder bất cứ khi nào có thể bởi vì nó nhanh hơn StringBuffer. Nhưng, nếu an toàn luồng (Thread Safety) là cần thiết, thì bạn nên sử dụng các đối tượng StringBuffer.

1. Các tính chất của lập trình hướng đối tượng trong Java?

Có 4 tính chất:

tính trừu tượng

tính đóng gói

tính kế thừa

tính đa hình

1. Phương thức finalize() làm gì?

Có thể định nghĩa một phương thức mà sẽ được gọi ngay trước khi hủy đối tượng bởi Garbage Collector (Trình dọn rác). Phương thức này được gọi là finalize(), và nó có thể được sử dụng để bảo đảm rằng đã hoàn toàn kết thúc một đối tượng.

1. Liệt kê ba bước để tạo một đối tượng cho một lớp?

Đầu tiên, một đối tượng được khai báo, sau đó thuyết minh và cuối cùng là khởi chạy.

1. Java Server Page (JSP) là một tài liệu văn bản chứa hai loại văn bản:

Static data và

JSP elements

1. Exception là gì?

Một Exception là một vấn đề được tạo ra trong khi thực thi một chương trình.

Các Exception được bắt bởi Handler được xác định cùng với lời gọi phương thức của Thread.

1. Giá trị mặc định của kiểu dữ liệu byte trong Java?

Giá trị mặc định của kiểu dữ liệu byte là 0.

1. Biến local hay biến cục bộ là gì?

Các biến được định nghĩa bên trong phương thức, constructor hoặc các khối được gọi là biến local. Biến này sẽ được khai báo và khởi tạo bên trong phương thức và nó sẽ bị hủy khi phương thức kết thúc.

1. Biến local hay biến cục bộ là gì?

Các biến được định nghĩa bên trong phương thức, constructor hoặc các khối được gọi là biến local. Biến này sẽ được khai báo và khởi tạo bên trong phương thức và nó sẽ bị hủy khi phương thức kết thúc.

1. Sự khác biệt giữa JDK và JRE là gì?

Java Runtime Environment (JRE) về cơ bản là Java Virtual Machine (JVM) nơi các chương trình Java của bạn đang được thực thi. Nó cũng bao gồm các plugin trình duyệt để thực thi applet.

Java Development Kit (JDK) là Software Development Kit đầy đủ tính năng cho Java, bao gồm JRE, các trình biên dịch và tools (như JavaDoc và Java Debugger), để người dùng phát triển, biên dịch và thực thi các ứng dụng Java.

1. Hỏi về collection Framework ( đa phần sẽ gặp khi đi phỏng vấn)?

Sự khác nhau giữa Set với Map.

Sự khác nhau giữa Linklist với Arraylist; Vector với Arraylist; HashTable với HashMap.

1. Liệt kê một số JDE của Java?

Netbeans, Eclipse, …

1. Biến instance là gì?

Biến instance là các biến bên trong một lớp nhưng bên ngoài bất cứ phương thức nào. Những biến này được thuyết minh khi lớp được tải.

1. Servlet là gì?

Servlet là một class ngôn ngữ lập trình Java được sử dụng để xử lý các requests từ client và tạo ra nội dung web động. Servlet chủ yếu được sử dụng để xử lý hoặc lưu trữ dữ liệu được gửi bởi một biểu mẫu HTML, cung cấp nội dung động và quản lý thông tin state không tồn tại trong giao thức stateless HTTP.

1. Hai loại Exceptions trong Java là gì? Sự khác biệt giữa chúng?

Java có hai loại exceptions: unchecked exceptions và checked exceptions.

Unchecked exceptions không cần phải được khai báo trong một phương thức hoặc mệnh đề throws của phương thức khởi tạo, nếu chúng có thể được throw bởi việc thực thi phương thức hoặc hàm tạo và lan truyền bên ngoài phương thức hoặc ranh giới của hàm tạo.

Mặt khác, Checked exceptions phải được khai báo trong một phương thức hoặc mệnh đề throws của phương thức khởi tạo.

1. Mô tả về Contructor?

Contructor được triệu hồi khi một đối tượng mới được tạo. Mỗi lớp có một Constructor.

Nếu chúng ta không viết một constructor một cách tường minh cho một lớp, thì Java Compiler sẽ xây dựng một Constructor mặc định cho lớp đó.

1. Giá trị mặc định của kiểu dữ liệu float và double trong Java?

Giá trị mặc định của kiểu dữ liệu float và double là khác với trong C/C++.

Mặc định của float là 0.0f và của double là 0.0d.

1. Lập trình hướng đối tượng là gì?

Lập trình hướng đối tượng là 1 kỹ thuật lập trình, cho phép lập trình viên trừu tượng hóa các đối tượng thực tế thành các đối tượng trong code.

1. Sự khác nhau giữa JDK,JRE và JVM?

JVM

JVM (viết tắt của Java Virtual Machine) là một thiết bị trừu tượng (ảo) có thể giúp máy tính chạy các chương trình Java. Nó cung cấp môi trường runtime mà trong đó Java Bytecode có thể được thực thi.

JRE

JRE (là viết tắt của Java Runtime Environment) được sử dụng để cung cấp môi trường runtime. Nó là trình triển khai của JVM. JRE bao gồm tập hợp các thư viện và các file khác mà JVM sử dụng tại runtime. Trình triển khai của JVM cũng được công bố bởi các công ty khác ngoài Sun Micro Systems.

JDK

JDK (là viết tắt của Java Development Kit) bao gồm JRE và các Development Tool.

1. Sự khác nhau giữa bộ nhớ heap và stack trong java?

Stack là một vùng nhớ được sử dụng để lưu trữ các tham số và các biến local của phương thức mỗi khi một phương thức được gọi ra.

Heap là một vùng nhớ trong bộ nhớ được sử dụng để lưu trữ các đối tượng khi từ khóa new được gọi ra, các biến static và các biến toàn cục (biến instance).

1. Các từ delete, next, main, exit và null có phải là từ khóa trong java không?

Không

1. Nếu không cung cấp bất kỳ đối số nào trên command line, thì mảng String của hàm main là empty hay null?

Mảng String là empty, không phải null.

1. Chuyện gì xảy ra nếu khai báo static public void thay vì public static void?

Chương trình được biên dịch và run đúng.

1. Giá trị mặc định của các biến local là gì?

Các biến local không được khởi tạo với bất kỳ giá trị mặc định nào, bất kể là nguyên thủy hay tham chiếu đối tượng.

1. Sự khác biệt giữa ngôn ngữ lập trình hướng đối tượng và ngôn ngữ lập trình dựa trên đối tượng là gì?

Ngôn ngữ lập trình dựa trên đối tượng thực hiện theo tất cả các tính năng của OOPs ngoại trừ tính kế thừa.

1. Giá trị khởi tạo của biến tham chiếu đối tượng được định nghĩa là biến instance là gì?

Trong java, các tham chiếu của đối tượng được khởi tạo là null.

1. Constructor là gì?

Constructor giống như một phương thức được sử dụng để khởi tạo trạng thái của một đối tượng. Nó được gọi ra vào thời điểm tạo ra đối tượng.

1. Mục đích của constructor là gì?

Constructor mặc định cung cấp các giá trị mặc định cho các đối tượng. Trình biên dịch java tạo ra một constructor mặc định chỉ khi không có constructor nào trong lớp.

1. Constructor trả về kiểu giá trị gì?

Constructor trả về thể hiện của lớp hiện tại.

1. Constructor được kế thừa không?

Không, constructor không được kế thừa.

1. Có thể tạo constructor final không?

không, constructor không thể là final.

1. Biến static là gì?

Biến static có thể được sử dụng để tham chiếu thuộc tính chung của tất cả đối tượng (mà không là duy nhất cho mỗi đối tượng), ví dụ như tên công ty của nhân viên, tên trường học của các sinh viên, …

Biến static lấy bộ nhớ chỉ một lần trong Class Area tại thời gian tải lớp đó.

1. Phương thức static là gì?

Một phương thức static thuộc lớp chứ không phải đối tượng của lớp.

Một phương thức static gọi mà không cần tạo một instance của một lớp.

Phương thức static có thể truy cập biến static và có thể thay đổi giá trị của nó.

1. Tại sao phương thức main là static?

Bởi vì không cần thiết phải tạo đối tượng để gọi phương thức static. Nếu nó là phương thức non-static, JVM đầu tiên tạo đối tượng và sau đó gọi phương thức main() mà có thể gây ra vấn đề về cấp phát bộ nhớ bộ nhớ phụ.

1. Chuyện gì xảy ra khi phương thức main không có static?

Chương trình được biên dịch, nhưng gặp lỗi ngoại lệ tại runtime là "NoSuchMethodError".

1. this trong java là gì?

this là từ khóa trong java tham chiếu đến đối tượng hiện tại.

1. Kế thừa là gì?

Kế thừa là một cơ chế trong đó một đối tượng được thừa hưởng tất cả thuộc tính và phương thức của đối tượng khác của lớp khác. Nó được sử dụng để tái sử dụng code và ghi đè phương thức

1. Các 4 tính chất quan trọng trong OOP (Oject Oriented Programming)

* Tính kế thừa(Inheritance): Cho phép một lớp con chỉ kể thừa các thuộc tính và phương thức của lớp cha.Lớp con có thể mở rộng hoặc mở rộng lại (override) các phương thức của lớp cha.Kế thừa giúp tái sử dụng mã nguồn, tăng tính linh hoạt và giảm sự trùng lặp trong mã.
* Tính đa hình(Polymorphism): Tính đa hình cho phép một đối tượng có thể biểu diễn dưới nhiều hình thái khác nhau.Đối tượng của lớp con có thể được coi là đối tượng của lớp cha, cho phép gọi các phương thức của lớp cha.Đa hình giúp tăng tính linh hoạt và mở rộng trong việc thiết kế và sử dụng đối tượng.
* Tính đóng gói(Encapsulation): Tính đóng gói đảm bảo rằng dữ liệu và mã nguồn được bảo vệ và chỉ được truy cập thông qua các phương thức công khai (public methods).Dữ liệu trong một lớp được ẩn đi và chỉ có thể được truy cập thông qua các phương thức getter và setter.Đóng gói giúp bảo mật dữ liệu và tạo ra các phương thức giao tiếp chuẩn.
* Tính trừu tượng(Abstraction): Tính trừu tượng giúp tạo ra các lớp và đối tượng trừu tượng không có sự liên quan trực tiếp đến các đối tượng cụ thể.Một lớp trừu tượng có thể chứa các phương thức trừu tượng (abstract methods) mà không có phần thân.Trừu tượng giúp tách biệt trách nhiệm và tập trung vào các khía cạnh quan trọng của đối tượng.

1. Các khái niệm trong java 🎉

* Class: là một bản thiết kế cho đối tượng cho một đối tượng cự thể. Nó định nghĩa các thuộc tính (properties) và phương thức (methods)
* Interface: là một tập hợp các phương thức trừu tượng (abstract methods) mà một lớp có thể triển khai
* Record: Là một loại đăcj biết của lớp giới hạn được thiệu từ java 14. Nó giup đinh nghĩa một cách ngắn gọn chỉ chứ dữ liệu ko thay đổi(imutable java). Tụ đọng getter, setter, hasCode,..
* Enum: Là tập hợp các hàng số đuọcw liêtj kê, mỗi giá trị là một đối tượng duy nhất của enum
* Annotation: các chỉ thị đặc biêt hay đánh dấu để cung câp thông tin bổ sung và metadata trong java. Chúng đánh dấu các lớp, phương thức, trường. Nó được thực thi trong qua trình biện dịch

1. Các tính năng trọng java 8: Lambda expressions, Functional Interface, StreamAPI, DefaultMethod(interface), Method references, Optional class, Date Time API, As
2. Lớp nào là lớp cha cho tất cả các lớp.

Lớp object

1. Tại sao đa kế thừa không được hỗ trợ trong java.

Để giảm thiểu sự phức tạp và đơn giản hóa ngôn ngữ, đa kế thừa không được support trong java.

1. Tại sao java không support con trỏ?

Con trỏ là một biến tham chiếu tới địa chỉ ô nhớ. Nó không được sử dụng trong java vì nó không an toàn và phức tạp

1. super trong java là gì?

Nó là một từ khóa mà tham chiếu trực tiếp đến đối tượng của lớp cha.

1. Có thể sử dụng cả this() và super() trong một constructor?

Không, vì this() gọi đến một constructor khác trong lớp hiện tại, còn super() gọi constructor của lớp cha.

1. Overloading (nạp chồng) phương thức là gì?

Nếu một lớp có nhiều phương thức có tên giống nhau nhưng các tham số khác nhau, được gọi là overloading phương thức (nạp chồng phương thức). Nó giúp code rõ ràng, dễ hiểu hơn.

1. Tại sao overloading phương thức không xảy ra khi thay đổi kiểu giá trị trả về?

Bởi vì đó là sự không rõ ràng, không biết gọi phương thức nào khi runtime.

1. Có thể overload phương thúc main() không?

Có, có thể overload phương thức main().

1. Ghi đè (overriding) phương thức là gì?

Nếu lớp con có phương thức giống lớp cha được gọi là ghi đè (overriding) phương thức trong java.

Nói cách khác, nếu lớp con cung cấp sự cài đặt cụ thể cho phương thức đã được cung cấp bởi một lớp cha của nó được gọi là ghi đè (overriding) phương thức trong java.

1. Có thể ghi đè phương thức static không?

Bạn không thể ghi đè phương thức static vì chúng thuộc về class chứ không thuộc về đối tượng.

1. Tại sao không thể ghi đè phương thức static?

Bởi vì phương thức static là một phần của lớp và nó bị ràng buộc với lớp, trong khi phương thức instance là bị ràng buộc với đối tượng, static được lưu trong vùng nhớ Class và instance được lưu trong bộ nhớ heap.

1. Có thể ghi đè phương thức đã nạp chồng?

Có

1. Biến final là gì?

Nếu bạn tạo một biến với từ khóa final, bạn sẽ không thể thay đổi được giá trị của biến đó (hằng số).

1. Phương thức final là gì?

Phương thức final không thể được ghi đè.

1. Lớp final là gì?

Lớp final không thể được kế thừa.

1. Biến final blank là gì?

Một biến final không được khởi tạo giá trị lúc khai báo được gọi là biến final blank.

1. Có thể khởi tạo giá trị cho biến final blank không?

Có, nếu biến đó là non-static thì chỉ khởi tạo được trong constructor. Nếu biến đó là static thì chỉ khởi tạo được trong khối static.

1. Có thể khai báo phương thức main là final không?

Có, giống như: public static final void main(String[] args){}

1. Trừu tượng là gì?

Tính trừu tượng là một tiến trình ẩn các cài đặt chi tiết và chỉ hiển thị tính năng tới người dùng.

1. Sự khác nhau giữa trừu tượng và đóng gói là gì?

Trừu tượng là ẩn đi cài đặt chi tiết còn đóng gói là gói code và data vào một khối duy nhất.

1. Lớp trừu tượng là gì?

Một lớp được khai báo với từ khóa abstract là lớp trừu tượng trong Java. Cần có một lớp khác kế thừa nó và cài đặt phương thức của nó. Nó không thể là thể hiện cụ thể.

1. Có phương thức trừu tượng không nằm trong lớp trừu tượng không?

Không, nếu có bất kỳ phương thức trừu tượng nào trong lớp, thì lớp đó phải là lớp trừu tượng.

1. Có thể sử dụng cả abstract và final cho một phương thức không?

Không, vì phuong thức trừu tượng (abstract) cần phải được ghi đè, trong khi đó không thể ghi đè được phương thức final.

1. Interface là gì?

Một Interface trong Java là một bản thiết kế của một lớp. Nó chỉ có các phương thức trừu tượng. Interface là một kỹ thuật để thu được tình trừu tượng hoàn toàn và đa kế thừa trong Java.

1. Có thể khai báo một phương thức của interface với từ khóa static không?

Không. Vì mặc định các phương thức của một interface là trừu tượng, từ khóa static và abtract không thể được sử dụng chung với nhau.

1. Một interface có thể là final không?

Không. Vì cài đặt phải được cung cấp bởi một lớp khác.

1. Có thẻ định nghĩa private hoặc protected cho các biến trong interface không?

Không. chúng phải là public.

1. Package là gì?

Một package (gói) trong java là một nhóm các kiểu tương tự của các lớp, giao diện và các package con .

1. Có cần import package import java.lang không? tại sao?

Không. Vì nó được tải gầm bởi JVM.

1. Có thể import package/lớp giống nhau hai lần không? JVM sẽ tải package hai lần khi chạy không?

Chúng ta có thể import package/lớp giống nhau nhiều lần, trình biên dịch và JVM sẽ không phàn nàn gì về vấn đề này. Nhưng JVM sẽ tải lớp đó một lần.

1. Static import là gì?

Với static import, chúng ta có thể truy cập đến các thành viên của một lớp một cách trực tiếp. Không cần phải sử dụng tên class cũng có thể truy cập.

1. Xử lý ngoại lệ (handling exception) là gì?

Exception Handling trong java hay xử lý ngoại lệ trong java là một cơ chế mạnh mẽ để xử lý các lỗi runtime để có thể duy trì luồng bình thường của ứng dụng.

1. Sự khác biệt giữa checked exception và unchecked exception là gì?

2.1. Checked exception

Các lớp được extends lớp Throwable ngoại trừ RuntimeException và Error được gọi là các checked exception. Ví dụ: IOException, SQLException, ... Checked exception được kiểm tra tại thời gian biên dịch.

2.2. Unchecked exception

Các lớp được extends lớp RuntimeException được gọi là unchecked exception. Ví dụ như ArithmeticException, NullPointerException, ... Unchecked exception không được kiểm tra tại thời gian biên dịch.

1. Có phải mỗi khối try phải đi kèm với một khối catch?

Nó không phải là cần thiết mà mỗi khối thử phải được theo sau bởi một khối catch. Nó phải được theo sau bởi một khối catch hoặc một khối finally. Và bất kỳ trường hợp ngoại lệ có thể bị ném sẽ được khai báo với từ khóa throws của phương thức.

1. Khối finally là gì?

Khối lệnh finally trong java được sử dụng để thực thi các lệnh quan trọng như đóng kết nối, đóng cá stream,…

Khối lệnh finally trong java luôn được thực thi cho dù có ngoại lệ xảy ra hay không.

Khối lệnh finally trong java được khai báo sau khối lệnh try hoặc sau khối lệnh catch.

1. Khối finally có thể được sử dụng mà không cần khối catch không?

Có. Bởi khối try. Khối finally phải theo sau khối try hoặc catch.

1. Có trường hợp nào khối finally không được thực thi không?

Khối finally không được thực thi nếu chương trình bị thoát (bằng cách gọi System.exit() hoặc lỗi phần cứng.

1. Sự khác nhau giữa throw và throws là gì?

|  |  |  |
| --- | --- | --- |
| 1) | Từ khóa throw trong java được sử dụng để ném ra một ngoại lệ rõ ràng. | Từ khóa throws trong java được sử dụng để khai báo một ngoại lệ. |
| 2) | Ngoại lệ checked không được truyền ra nếu chỉ sử dụng từ khóa throw. | Ngoại lệ checked được truyền ra ngay cả khi chỉ sử dụng từ khóa throws. |
| 3) | Sau throw là một instance. | Sau throws là một hoặc nhiều class. |
| 4) | Throw được sử dụng trong phương thức. | Throws được khai báo ngay sau dấu đóng ngoặc đơn của phương thức. |
| 5) | Bạn không thể throw nhiều exceptions. | Bạn có thể khai báo nhiều exceptions, Ví dụ: public void method()throws IOException,SQLException. |

1. Có thể khai báo phương thức overriding của lớp con một ngoại lệ nếu phương thức của lớp cha không throw một ngoại lệ?

Được nhưng chỉ đối với unchecked exception còn checked exception thì không.

1. Việc tuyên truyền ngoại lệ là gì?

Việc chuyển tiếp đối tượng ngoại lệ sang phương thức gọi được gọi là sự truyền ngoại lệ.

1. Ý nghĩa của immutable (bất biến) trong String là gì?

Ý nghĩa đơn giản của immutable là không thể sửa đổi hoặc không thể thay đổi được. Một khi đối tượng String đã được tạo ra, giá trị của nó không thể thay đổi.

1. Tại sao các đối tượng String trong java là immutable?

Bởi vì java sử dụng khái niệm về string literal. Giả sử có 5 biến tham chiếu, tất cả các tham chiếu đến một đối tượng "hello". Nếu một biến tham chiếu thay đổi giá trị của đối tượng, nó sẽ bị ảnh hưởng đến tất cả các biến tham chiếu khác. Đó là lý do tại sao đối tượng String trong java immutable (bất biến).

1. Có bao nhiêu cách để tạo ra một đối tượng String trong java?

Có 2 cách để tạo đối tượng String:

Sử dụng string literal

Sử dụng từ khóa new

1. Có bao nhiêu đối tượng String được tạo ra trong đoạn code sau?

String s1="Hello";

String s2="Hello";

String s3="World";

Chỉ có 2 đối tượng được tạo ra.

1. Tại sao java sử dụng khái niệm string literal?

Vì điều này khiến Java sử dụng bộ nhớ hiệu quả hơn (vì không có đối tượng String mới được tạo ra trong string constant pool).

1. Có bao nhiêu đối tượng được tạo ra trong đoạn code sau?

String s = new String("Welcome");

Có 2 đối tượng được tạo ra. Một đối tượng được tạo ra trong string constant pool và một đối tượng được tạo ra trong bộ nhớ heap bởi từ khóa new.

1. Sự khác nhau giữa String và StringBuffer là gì?

String là một đối tượng immutable (bất biến). StringBuffer là môt đối tượng mutable (có thể biến đổi).

1. Sự khác nhau giữa StringBuffer và StringBuilder là gì?

StringBuffer là đồng bộ còn StringBuilder là KHÔNG đồng bộ.

1. Làm thế nào để tạo lớp immutable trong java?

Chúng ta có thể tạo ra lớp immutable như lớp String bằng cách định lớp với từ khóa final và

1. Mục đích của phương thức toString() trong java là gì?

Phương thức toString() trả về biểu diễn chuỗi của bất kỳ đối tượng nào

1. Nested class (lớp lồng nhau) là gì?

Một lớp được khai báo bên trong một lớp khác được gọi là lớp lồng nhau. Có 4 loại lớp bên trong lớp lồng nhau: member inner class, local inner class, annonymous inner class và static nested.

1. Có sự khác nhau giữa nested class và inner class không?

Có. Inner class là non-static nested class nghĩa là inner class là một kiểu nested class.

1. Nested interface là gì?

Bất kỳ interface nào được khai báo bên trong interface hoặc class, được biết đến nested interface. Nó có thể là static hoặc default.

1. Có thể khai báo interface trong class không?

Có. Nó được biết đến như nested interface.

1. Có thể khai báo một class trong interface không?

Có. Nhưng chúng không liên quan đến nhau.

1. Sự khác nhau giữa ArrayList và Vector là gì?

|  |  |  |
| --- | --- | --- |
| **No.** | **ArrayList** | **Vector** |
| 1) | ArrayList là KHÔNG synchronized. | Vector là synchronized. |
| 1) | ArrayList không phải là legacy class. | Vector là legacy class. |
| 2) | ArrayList tăng kích thước của nó bằng 50% kích thước mảng. | Vector tăng kích thước của nó bằng cách nhân đôi kích thước mảng. |

1. Sự khác nhau giữa ArrayList và LinkedList là gì?

|  |  |  |
| --- | --- | --- |
| **No.** | **ArrayList** | **LinkedList** |
| 1) | ArrayList sử dụng một mảng động. | LinkedList sử dụng danh sách liên kết doubly. |
| 2) | ArrayList không hiệu quả với thao tác vì cần nhiều chuyển đổi. | LinkedList là hiệu quả cho thao tác. |
| 3) | ArrayList là tốt hơn để lưu trữ và lấy dữ liệu. | LinkedList là tốt hơn để thao tác dữ liệu. |

1. Sự khác nhau giữa Iterator và ListIterator là gì?

Iterator duyệt các phần tử chỉ theo một chiều hướng là chuyển tiếp, trong khi ListIterator duyệt các phần tử theo hai hướng là chuyển tiếp và ngược lại.

Iterator có thể được sử dụng trong List, Set và Queue.

ListIterator chỉ có thể được sử dụng trong List.

1. Sự khác nhau giữa List và Set là gì?

List có thể chứa các phần tử trùng lặp (dublicate), trong khi Set chỉ chứa các phần tử duy nhất

1. Sự khác nhau giữa HashSet và TreeSet là gì?

HashSet **không** duy trì **thứ tự nào**, trong khi TreeSet duy trì **thứ tự tăng dần**.

1. Sự khác nhau giữa Set và Map là gì?

Set chỉ chứa giá trị, trong khi Map chứa cặp key và value.

1. Sự khác biệt giữa HashSet và HashMap là gì?

HashSet chỉ chứa giá trị, trong khi HashMap chứa cặp key và value.

1. Sự khác nhau giữa HashMap và TreeMap là gì?

HashMap duy trì **không có thứ tự**, trong khi TreeMap duy trì **thứ tự tăng dần**

1. Sự khác nhau giữa HashMap và Hashtable là gì?

|  |  |  |
| --- | --- | --- |
| **No.** | **HashMap** | **Hashtable** |
| 1) | HashMap là KHÔNG synchronized. | Hashtable là synchronized. |
| 2) | HashMap có thể chứa một khóa null và nhiều giá trị null. | Hashtable không thể chứa bất kỳ khóa null hoặc giá trị null. |

1. Sự khác nhau giữa Collection và Collections là gì?

Collection là một interface, trong khi Collections là một lớp. Collecion interface cung cấp các chức năng về cấu trức dữ liệu cho List, Set, Queue. Nhưng lớp Collections là để sắp xếp và đồng bộ các phần tử Collection.

1. Sự khác nhau giữa Comparable và Comparator là gì?

|  |  |  |
| --- | --- | --- |
| **No.** | **Comparable** | **Comparator** |
| 2) | Nó cung cấp phương thức compareTo(). | Nó cung cấp phương thức compare(). |
| 3) | Nó được đặt trong java.lang package. | Nó được đặt trong java.util package. |
| 4) | Nếu chúng ta một lớp được implement Comparable interface, thì lớp đó phải được sửa đổi. | Lớp không bị sửa đổi. |

1. Phương thức hashCode() là gì?

Phương thức hashCode() trả về một giá trị mã băm (một số nguyên).

Phương thức hashCode() trả về cùng số nguyên, nếu hai keys (bằng phương thức equals()) giống nhau.

Tuy nhiên, có thể hai mã băm có thể có các keys khác nhau hoặc giống nhau.

1. Tại sao chúng ta phải nghi đè phương thức equals()?

Phương thức equals() được sử dụng để kiểm tra xem hai đối tượng có giống nhau hay không. Nó cần phải được ghi đè nếu chúng ta muốn kiểm tra các đối tượng dựa trên thuộc tính của chúng.

1. Lợi ích của generic collection là gì?

Nếu chúng ta sử dụng lớp generic, chúng ta không cần typecasting. Nó là typesafe và kiểm tra tại thời gian biên dịch.

1. Sự khác nhau giữa Array và ArrayList là gì?

|  |  |
| --- | --- |
| **Array** | **ArrayList** |
| 1) Kích thước **cố định**. | Kích thước có thể **thay đổi được**. |
| 2) Có thể lưu trữ dữ liệu kiểu **nguyên thủy** và **đối tượng**. | Chỉ có thể lưu trữ dữ liệu kiểu **đối tượng**. Kể từ Java 5, kiểu nguyên thủy được tự động chuyển đổi trong các đối tượng được gọi là **auto-boxing**. |
| 3) Tốc độ lưu trữ và thao tác **nhanh hơn**. | Tốc độ lưu trữ vào thao tác **chậm hơn**. |
| 4) Chỉ có thuộc tính **length**. | Có nhiều phương thức để thao tác với dữ liệu. |

1. Hibernate là gì?

Hibernate là một công cụ ORM mã nguồn mở, được sử dụng để lưu trữ, thao tác và truy xuất dữ liệu từ cơ sở dữ liệu.

1. ORM là gì?

ORM là từ viết tắt của Object/Relational mapping có nghĩa là ánh xạ đối tượng/quan hệ. Đây là một kỹ thuật lập trình để ánh xạ đối tượng với dữ liệu được lưu trữ trong cơ sở dữ liệu. Nó đơn giản hóa việc tạo dữ liệu, thao tác dữ liệu và truy cập dữ liệu.

1. ORM bao gồm những gì?

Giải pháp ORM bao gồm 4 phần như sau:

API để thực hiện các thao tác CRUD cơ bản.

API để thể hiện các truy vấn tham chiếu đến các lớp.

Phương tiện để chỉ định Metadata.

Tối ưu hóa phương tiện: dirty checking,lazy associations fetching.

1. Tại sao bạn cần công cụ ORM như Hibernate?

Ưu điểm chính của ORM như Hibernate là nó bảo vệ các nhà phát triển khỏi SQL lộn xộn. Ngoài ra, ORM cung cấp các lợi ích sau:

1. ORM có những ưu điểm gì so với JDBC?

ORM có những ưu điểm sau so với JDBC:

Phát triển ứng dụng nhanh.

Quản lý giao dịch.

Tạo khóa tự động.

Chi tiết về các truy vấn SQL được ẩn đi.

1. Kể tên 2 thành phần chính của đối tượng cấu hình Hibernate.

Database Connection

Class Mapping Setup

1. HQL là gì?

HQL là viết tắt của Hibernate Query Language - Ngôn ngữ truy vấn Hibernate được biết đến như một ngôn ngữ truy vấn hướng đối tượng. Nó giống như một ngôn ngữ truy vấn có cấu trúc (SQL).

Ưu điểm chính của HQL so với SQL là:

Không cần học SQL.

Độc lập cơ sở dữ liệu.

Đơn giản để viết một truy vấn.

1. Có bao nhiêu loại mapping trong Hibernate?

Có thể có 4 loại:

One to One

One to Many

Many to One

Many to Many

1. Có thể thực hiện mapping collection với One-to-One và Many-to-One không?

Không, chỉ có thể thể thực hiện mapping collection với One-to-Many and Many-to-Many.

1. lazy loading trong Hibernate là gì?

Nó tải các đối tượng con theo yêu cầu, được sử dụng để cải thiện hiệu suất.

### persistent class là gì?

Các lớp có đối tượng được lưu trữ trong bảng cơ sở dữ liệu.

1. Session interface đóng vai trò gì trong Hibernate

Nó là một đối tượng single-threaded, short-lived biêủ diễn giao tiếp giữa ứng dụng và persistent store, Nó cho phép bạn tạo các đôis tượng truy vấn để lấy các đối tượng được lưu trữ trong store.

1. Stack vs Queue?

Stack hoạt động theo nguyên lý LIFO – Last In First Out

Queue thì là FIFO – First In First Out

1. Đệ quy là gì

một phương thức gọi lại chính nó một cách liên tiếp

**JSP/Servlet**

1. **JSP là gì? Giải thích cách JSP hoạt động.**

**JSP (JavaServer Pages) là một công nghệ trong Java cho phép nhúng mã Java vào các trang HTML để tạo ra các trang web động. JSP được sử dụng để phát triển giao diện người dùng trong ứng dụng web Java.**

**Cách hoạt động JSP:**

1. **Khi một yêu cầu HTTP được gửi từ client tới máy chủ, máy chủ nhận yêu cầu và xác định rằng nó là 1 yêu cầu JSP**
2. **Máy chủ tạo một đối tượng JSP Servlet(chỉ được tạo một lần) để xử lý yêu cầu.**
3. **Đối tượng JSP Servlet biên dịch mã JSP thành mã Servlet tương ứng, được thực thi bơi máy áo JVM**
4. **Mã servlet tạo ra kêts quả HTML hoặc xml được gửi trở lại trinh duyệt**
5. **Kết quả được hiện thị trên trình duyệt client**

**2. Servlet là gì? Giải thích cách Servlet hoạt động.**

**Servlet là một thành phần của công nghệ Java để xây dựng ứng dụng web. Nó được sử dụng để xử lý các yêu cầu từ trình duyệt web và tạo ra các phản hồi tương ứng. Servlet là các lớp Java được triển khai trên máy chủ web để xử lý các yêu cầu HTTP và tạo ra các phản hồi HTTP.**

**Cách hoạt động:**

1. **Server nhận request từ client và xác định nó là một yêu cầu servlet**
2. **Server tìm và tạo 1 đối tượng servlet tương ứng từ lớp Servlet đã được triển khai**
3. **Đối tượng Servlet được tạo 1lần và tái sử dụng**
4. **Đối tượng Servlet nhận yêu cầu thông qua phương do Servlet Api như doGet(), doPost(), doPut(), doDelete()**
5. **Response từ server tới client dưới dạng HTTP, chẳng hạn như HTML, XML, JSON**
6. **Client hiển thị phản hồi cho ngươi dùng**

### 3. Có bao nhiêu đối tượng của một servlet được tạo ra

Chỉ có 1 đối tượng được tạo ra tại thời điểm request đầu bới servlet hoặc web container

### Vòng đời của một servlet là gì?

Một vòng đời của servlet là toàn bộ quá trình từ khi toạ ra đến khi huỷ:

Servlet được khởi tạo bằng cách gọi phương thức init()

Phương thức servlet service() được gọi để xử lý yêu câu cầu client

Servlet được huỷ bằng cách gọi phươngư thưc destroy()

Cuối cùng, servlet được thu thập bởi sưu tập rác của JVM

### Ai có trách nhiệm tạo đối tượng servlet?

Web container hoặc servlet container

### Đối tượng servlet được tạo ra khi nào?

Khi có request đầu tiên

1. Sự khác nhau phương thưc GET và Post

|  |  |
| --- | --- |
| **GET** | **POST** |
| 1) Số lượng dữ liệu được gửi cho mỗi request bị giới hạn, vì nó được gửi qua header. | Số lượng lớn dữ liệu được gửi cho mỗi request, vì nó gửi qua body. |
| 2) Không an toàn vì người dùng nhìn thấy dữ liệu rõ ràng trên URL. | An toàn vì người dùng không nhìn thấy dữ liệu trên URL. |

1. Sự khác nhau giữa PrintWriter and ServletOutputStream

PrintWriter là 1 lớp character-stream. có thể ghi các thông tin character-based

ServletOutputStream là 1 lớp byte-stream. có thi các giá trị nguyên thuỷ cx như character-based

1. Sự khác nhau giữa GenericServlet và HttpServlet là gì ?

GênericServlet là giao thức độc lập.

HttpServlet là giao thức HTTP cự thể cung cấp các chức năng bổ sung như quản lý state.

1. Servlet Collaboration là gì?

là một servlet giao tiếp với một servlet khác, các tạo giao tiếp : RequestDispacher interface hoặc sendRedirect()...

1. Mục đich của giao diện RequestDispacher là gì?

cung cấp khả năng gửi yêu cầu tới nguyền khác có thể html, servlet hoặc jsp

1. Sự khác nhau giữa ServletConfig và ServletContext là gì?

Container tạo đối tượng ServletConfig cho mỗi servlet trong khi đối tượng của ServletContext đuọc tạo ra cho mỗi ứng dụng web

### Sự khác nhau giữa phương thức forward() và sendRedirect() là gì?

|  |  |
| --- | --- |
| **forward()** | **sendRedirect()** |
| 1) forward() gửi yêu cầu tương tự tới một tài nguyên khác. | 1) sendRedirect() luôn luôn gửi yêu cầu mới vì nó sử dụng thanh URL của trình duyệt. |
| 2) forward() hoạt động ở phía server. | 2) sendRedirect() hoạt động ở phía client. |
| 3) forward() chỉ hoạt động trong server. | 3) sendRedirect() hoạt động bên trong và bên ngoài server. |

1. Cookies là gì?

Cookies là các tập tin văn bản được lưu trữ trên client. Mục đich của cookie là để theo dõi các thông tin khác nhau. ví dự remember login

1. Sự khác nhau giữa cookies và session là gì?

Cookies hoạt động ở phía client.

Session hoạt động ở phía server.

1. Filter là gì?

Là một đối tượng được gọi ở quá trình tiền xử lý hoặc sau khi xử lý một yêu cầu

1. Nhược điểm của cookie là gì?

Nó sẽ không hoạt động nêu cookie bị tắt từ trình duyệt hoặc trình duyệt không hỗ trợ cookie.

1. Các anotation được trong Sevlet là gì?

Có 3 annotation chủ yếu : @Webservlet @WebListener, @Wefilter

1. Sử dụng thuộc tính welcome-file-list trong web.xml để làm gì?

để chỉ định trang index của dự án web

1. Vòng đời của JSP giống trong servlet
2. JSTL là gì?

JSTL(JavaServer Pages Standard Tag Library) là một thư viện thẻ tiêu chuẩn trong JSP để làm cho việc phát triẻn ứng web Java dễ dàng và hiệu quả hơn. Nó cung cấp các thẻ tiện ích mạnh mẽ trong thao tác dự liệu, kiểm soát luòng và hiện thị giao diện lên trang JSP

1. Câc đổi tượng ấn trong JSP:

JSP cung cấp 9 đối tượng ẩn theo mặc định:

|  |  |
| --- | --- |
| **Đối tượng** | **Kiểu** |
| 1) out | JspWriter |
| 2) request | HttpServletRequest |
| 3) response | HttpServletResponse |
| 4) config | ServletConfig |
| 5) session | HttpSession |
| 6) application | ServletContext |
| 7) pageContext | PageContext |
| 8) page | Object |
| 9) exception | Throwable |

**Spring**

1. **Spring MVC là gì? Giải thích kiến trúc và nguyên tắc hoạt động của Spring MVC.**

**Spring MVC(Model - View - Controller) là một phần trong Spring Framework cugn cấp một các tiếp cận phát triển ứng dụng web dựa trên Mô hình MVC. Nó giúp tách logic ứng dụng và giao diện người dùng, cung cấp một cấu trúc tổ chức rõ ràng và một quy tắc chuẩn để phát triển các ứng dụng web linh hoạt, dễ bảo trì và kiểm thử**

**Kiến trúc Spring MVC chia làm 3 phần:**

**Model: Đại diện cho dữ liệu và quy tác kinh doanh của ứng dụng. Model này đảm nhận trách nhiệm cập nhạt và xử lý dữ liệu.**

**View: đại diện cho phần giao diện người dùng. nó hiển thị dữ liệu từ model vag thu thâp thông tin từ người dùng. View có thể được dụng JSP, Thymealeaf Template,..**

**Controller: Bộ điều khiển: đại diện cho thành phần xử lý các yêu càu và điều huong luồng của ứng dụng. Controller nhận yêu cầu từ người dùng thông qua giao diện. và gọi các phương thưc trong model để xử lý dữ liệu và chọn view phù hợp. Ở controller xử liên quan tối xác thực, uỷ quyền và xử lý lỗi**

1. **Spring Boot và Spring MVC khác nhau như thế nào?**

**Spring MVC là một phần trong spring framework tập trung vào phát triển ứng dụng web dựa trên mô hình MVC.**

**Spring Boot là một dự án con của spring framework tập trung vào việc giảm bớt công việc cấu hình và triển khai ứng dụng Java, giúp đơn giản hóa quá trình phát triển.**

1. **Servlet và Spring khác gì nhau?**

**Servlet và Spring là hai công nghệ khác nhau trong việc phát triển ứng dụng web.**

**Sự khác nhau:**

**Về phạm vi ứng dụng: Servlet là một công nghệ cơ bản của Java để phát triển ứng dụng web, Nó tập vào xử lý yêu cầu và phản hồi HTTP, cung cấp các phương để xây dụng ứng dụng web động. Spring là framework phát úng dụng Java toàn diện, Spring cung cấp một giải pháp linh hoạt và mạnh mẽ để phát triển ứng dụng web và ứng dụng doanh nghiệp**

**Về kiến trúc và cú pháp: Servlet sử dụng các lớp và phương thức để xử lý yêu cầu và phản hồi HTTP. Nó được triển khải thông qua việc viết mã Java và cấu hình trong các tệp cấu hình xml. Spring cung một 1 hình phát triển linh hoạt hơn, sử dụng Dependency Injection DI và Inversion of Control (IOC) để quản lý các thành phần và giảm sự phụ thuọc vào các API servlet cơ bản.**

**Về tiện ích và tính năng: Servlet cung các tiện ích cơ bản để xử lý yêu cầu HTTP, nhu đọc và ghi thông tin, xử lý phiên, quản lý trạng thái, và điều huong yêu cầu đến các tài nguyên khác. Spring cung câp một loatj các tính năng hữu tiền lõi bào Denpendency Injection , AOP, quản lý transaction, xác thực và uỷ quyền, caching, kiẻm thư, nhiều hơn nữa.**

1. **Spring là gì?**

Spring là một khung framework mã nguồn mở, dựa trên Java. Nó được thiết kế để giúp các nhà phát triển Java tạo các ứng dụng web cụ thể.

**Câu 1: IoC là gì và lợi ích của nó?**

IoC(Inversion Of Control) có nghĩa là đảo ngược điều khiển, nó giúp làm thay đổi luồng điều khiển chương trình 1 cách đơn giản và làm cho lượng code được tối giản, dễ test hơn. Có 2 loại IOC: Application Context và Bean Factory

**Câu 2: DI là gì?**

DI(Dependency injection) là tiêm phụ thuộc được quản lý bởi Spring Container và IoC để có thể writing vào các object để ứng dụng trở nên linh hoạt, dễ bảo trì, dễ mở rộng hơn.

**Câu 3: Beans là gì?**

Beans là những Java Object được quản lý bởi Spring Container.

Khi sử dụng @Beans là đang khai báo 1 kiểu Object để cho Spring Container quản lý. Điều này giúp cho ứng dụng nhẹ hơn, dễ quản lý và mở rộng ứng dụng.

**Câu 4: @Autowired là gì?**

Để tiêm các Object được quản lý bởi Spring Container vào class muốn sử dụng object đó.

Cau 4.1 : Scope trong Bean

Cos

**Câu 5: Framework thường sử dụng để ứng dụng java kết nối với database?**

Spring JDBC Framework , Hibernate, JPA

**Câu 6: MVC là gì? Phân tích tác dụng từng thành phần của mô hình?**

MVC là mô hình thiết kế đươc viết tắt từ 3 thành phần là Model – View – Controller.

Model: Là bộ phận để lưu trữ và xử lý dữ liệu

View: Giao diện hiển thị, tương tác với người dùng

Controller: điều khiển luồng thực thi của chương trình

**Câu 7: Luồng đi trong mô hình MVC khi có 1 request được gửi đến server là gì?**

Chúng ta sẽ có các bước thực hiện như sau:

Bước 1: Request gửi đến server sẽ vào controller

Bước 2: Controller phân tích cần lấy dữ liệu ở đâu nên sẽ gọi đến model

Bước 3: Model phân tích yêu cầu đề lấy dữ liệu cần thiết và trả về lại cho controller

Bước 4: Controller lấy dữ liệu đó gửi cho view để lấp đầy những chỗ trống của dữ liệu còn thiếu và hiển thị giao diện cho client

**Câu 8: @Controller là gì?**

Khai báo 1 class là controller như 1 lớp điều khiển

**Câu 9: @RequestMapping là gì?**

Để map 1 URL với 1 class hoặc 1 phương thức

**Câu 10: Kiến trúc Repository là gì?**

Là 1 design patten, thiết kế thêm 1 tầng repository để tương tác với database. Giúp database được bảo mật và an toàn hơn.

1. ‘Boot’ là gì?

Spring Boot là một phiên bản của Spring mà mục đích để làm cho ứng dụng tạo ra quá trình sắp xếp hợp lý hơn.

Câu 11: Boot “initilizr" sử dụng để làm gì?

* Là công cụ để tạo ra 1 khuôn mẫu project ban đầu cho dự án spring

Câu 12: Có bao nhiêu cachs đẻ thực hiện Denpendency Ịnjection

* Có 3 cách làm: Constructor injection , setter Injection, Interface injection

Câu 13: Có nhiêu giao thức trong HTTP?

* Có 3 : HTTP/1.1, HTTP/2, HTTP/3

Câu 14: Có nhiêu phương thức trong HTTP/3

* Có 5 Phương thưc thương sử dụng: GET, POST, PUT, DELETE, PATCH
* GET: yêu cầu lấy thông tin tài nguyên từ máy chủ.
* POST: yêu cầu gửi dữ liệu đến máy chủ để xử lý.
* PUT: yêu cầu cập nhật tài nguyên có sẵn trên máy chủ.
* DELETE: yêu cầu xóa tài nguyên có sẵn trên máy chủ.
* PATCH: yêu cầu chỉnh sửa một phần tài nguyên có sẵn trên máy chủ.

Câu 15: **Spring JPA** là gì và nó được sử dụng để làm gì?

Spring JPA (Java Persistence API) là một phần mền của Spring Framework, là một công nghệ tương tác với các hệ quản trị CSDL NHƯ MySQL, SQL Server

Spring JPA là một công nghệ quan trọng trong việc phát triển các ứng dụng Java liên quan đến cơ sở dữ liệu, giúp cho việc tương tác với cơ sở dữ liệu trở nên đơn giản và dễ dàng hơn.

Câu 16: Các annotation quan trọng được sử dụng trong Spring JPA?

* @Entity: Định nghĩa một entity.
* @Table: Xác định tên bảng tương ứng cho một entity.
* @Id: Xác định trường đại diện cho khoá chính của entity.
* @GeneratedValue: Xác định cách sinh ra giá trị cho khoá chính của entity.
* @Column: Xác định tên cột tương ứng cho một trường của entity.
* @ManyToOne, @OneToMany, @ManyToMany, @OneToOne: Xác định quan hệ giữa các entity.
* @Query: Định nghĩa câu truy vấn để lấy dữ liệu từ cơ sở dữ liệu.
* @Transactional: Xác định phạm vi của một giao dịch.

Câu 17: Các cách để thực hiện các truy vấn tùy chỉnh trong Spring JPA?

* Sử dụng method naming conventions: findbyName
* Sử dụng @Query annotation : truy vấn câu lệnh bằng JPQL hoặc SQL
* Sử dụng Named Queries
* Sử dụng Criteria API
* Sử dụng QueryDSL

Câu 18: Các tính năng khác nhau của EntityManager và JpaRepository?

1. EntityManager: Đây là một giao diện quản lý thực thể và chu kỳ cuộc gọi của đối tượng trong cơ sở dữ liệu. EntityManager được sử dụng để thực hiện các thao tác cơ bản với cơ sở dữ liệu, chẳng hạn như truy vấn, thêm, sửa và xóa thực thể.
2. JpaRepository: Đây là một giao diện mở rộng của Repository, cung cấp các phương thức tiện ích để thao tác với cơ sở dữ liệu. JpaRepository cung cấp các phương thức đặc biệt hơn so với EntityManager, chẳng hạn như tìm kiếm theo tiêu chí, sắp xếp, phân trang và truy vấn tự động.

Câu 19: **Spring Security** là gì và nó được sử dụng để làm gì?

Spring Security là một framework bảo mật mạnh mẽ cho ứng dụng Java, được sử dụng để bảo vệ các ứng dụng web và RESTful. Nó cung cấp các tính năng bảo mật như xác thực, phân quyền và bảo vệ chống tấn công CSRF, XSS và SQL Injection.

Spring Security được sử dụng để xác định danh tính người dùng, xác thực người dùng và kiểm soát quyền truy cập đối với các tài nguyên của ứng dụng. Nó cũng cung cấp các tính năng bảo mật cho việc xác thực các API RESTful và các kết nối WebSocket.

Câu 20: Làm thế nào Spring Security giúp bảo vệ ứng dụng web?

* Xác thực người dùng: Xác thực bằng CSDL, OAuth, OpenID Connect
* Phân quyền người dung
* Bảo vệ tại nguyên
* Kiểm tra và giám sát hoạt động:

Câu 21: Các tính năng chính của Spring Security bao gồm:

1. Xác thực (Authentication): xác thực người dùng và đảm bảo rằng họ có quyền truy cập vào các tài nguyên được yêu cầu.
2. Phân quyền (Authorization): quản lý các quyền và vai trò người dùng để đảm bảo rằng họ chỉ có thể truy cập vào các tài nguyên cần thiết và được ủy quyền.
3. Quản lý phiên (Session Management): quản lý phiên của người dùng để đảm bảo rằng các phiên được xử lý đúng cách và đủ an toàn.
4. Bảo vệ CSRF (Cross-Site Request Forgery): đảm bảo rằng các yêu cầu từ một trang web được tạo bởi người dùng được xác minh và có thẩm quyền để tránh các cuộc tấn công giả mạo yêu cầu giữa các trang web.
5. Bảo vệ XSS (Cross-Site Scripting): đảm bảo rằng dữ liệu nhập vào từ người dùng được lọc và mã hóa để tránh các cuộc tấn công XSS.
6. Bảo vệ lưu trữ (Secure Data Storage): đảm bảo rằng các thông tin nhạy cảm được lưu trữ một cách an toàn và được mã hóa đúng cách để tránh các cuộc tấn công lưu trữ.
7. Bảo vệ truyền thông (Secure Communication): đảm bảo rằng tất cả các truyền thông giữa máy khách và máy chủ được mã hóa và đủ an toàn để tránh các cuộc tấn công gián đoạn truyền thông.

Câu 22: Các thành phần cơ bản trong Spring Security gồm:

1. Authentication: Xác thực thông tin đăng nhập của người dùng.
2. Authorization: Xác định quyền truy cập của người dùng sau khi được xác thực.
3. UserDetailsService: Tìm kiếm thông tin người dùng, bao gồm thông tin đăng nhập và quyền truy cập.
4. UserDetails: Đại diện cho thông tin người dùng, bao gồm tên người dùng, mật khẩu và danh sách các quyền truy cập.
5. PasswordEncoder: Mã hóa mật khẩu của người dùng để lưu trữ an toàn hơn trong cơ sở dữ liệu.
6. Filter: Các bộ lọc được sử dụng để xác thực và phân quyền truy cập của người dùng.
7. AccessDecisionManager: Xác định quyền truy cập của người dùng dựa trên các quyết định từ các quy tắc cấp phép.
8. SecurityContextHolder: Lưu trữ thông tin về xác thực và phân quyền truy cập của người dùng trong phiên làm việc.
9. HttpSession: Lưu trữ thông tin phiên của người dùng sau khi xác thực thành công.

Câu 23: Các hình thức xác thực phổ biến trong Spring Security bao gồm:

1. Xác thực dựa trên form: người dùng đăng nhập bằng cách nhập thông tin đăng nhập (tên đăng nhập và mật khẩu) vào một trang đăng nhập đặc biệt được cung cấp bởi ứng dụng web. hay còn gọi là form-based authentication
2. Xác thực dựa trên token: trong hình thức này, người dùng đăng nhập và được cấp một mã thông báo (token) đại diện cho quyền truy cập của họ. Các yêu cầu sau đó sử dụng token để xác thực người dùng.
3. Xác thực dựa trên OAuth: OAuth là một giao thức xác thực phổ biến được sử dụng bởi các ứng dụng web để cho phép người dùng đăng nhập và chia sẻ thông tin từ các nguồn khác nhau mà không cần cung cấp tên đăng nhập và mật khẩu.
4. Xác thực dựa trên LDAP: LDAP (Lightweight Directory Access Protocol) là một giao thức xác thực được sử dụng bởi nhiều hệ thống quản lý người dùng để quản lý thông tin người dùng và phân quyền.
5. Xác thực dựa trên SSO: SSO (Single Sign-On) là một hình thức xác thực cho phép người dùng đăng nhập một lần và truy cập vào nhiều ứng dụng khác nhau mà không cần phải nhập thông tin đăng nhập lại.

Câu 24: Làm thế nào để tùy chỉnh quyền truy cập trong Spring Security?

Trong Spring Security, có thể tùy chỉnh quyền truy cập thông qua việc cấu hình các đối tượng SecurityConfigurations và sử dụng các annotation như @EnableWebSecurity, @EnableGlobalMethodSecurity và @PreAuthorize(PostAuthorize) để xác định các quyền truy cập cho các phương thức.

Câu 25: Làm thế nào để phân biệt giữa CSRF và XSS trong Spring Security?

* CSRF là một cuộc tấn công mà tin tặc sử dụng quyền truy cập của người dùng đang đăng nhập để thực hiện các hoạt động không mong muốn trên trang web, ví dụ như thực hiện một giao dịch tài chính hoặc thay đổi thông tin cá nhân. Các tấn công CSRF có thể được phòng ngừa bằng cách sử dụng token CSRF để xác thực các yêu cầu được gửi đến máy chủ từ người dùng.
* XSS là một cuộc tấn công mà tin tặc cố gắng chèn mã độc vào trang web để lấy thông tin người dùng, hoặc thực hiện các hành động khác nhau trên trang web. Các tấn công XSS có thể được phòng ngừa bằng cách sử dụng các kỹ thuật lọc đầu vào và mã hóa đầu ra để đảm bảo rằng dữ liệu nhập vào được xử lý đúng cách trước khi được hiển thị trên trang web.
* Trong Spring Security, CSRF và XSS đều được đối phó bằng cách sử dụng các tính năng bảo mật tích hợp sẵn. Ví dụ, Spring Security cung cấp tính năng CSRF Token để bảo vệ khỏi tấn công CSRF và cung cấp tính năng mã hóa đầu ra để bảo vệ khỏi các tấn công XSS. Ngoài ra, bạn có thể thực hiện các biện pháp bảo mật khác nhau để đối phó với các cuộc tấn công này, chẳng hạn như sử dụng thư viện mã hóa bên thứ ba và các kỹ thuật lọc đầu vào.

Câu 26: CORS là gì

* CORS (Cross-Origin Resource Sharing) là một cơ chế cho phép các trang web trong một domain (origin) yêu cầu tài nguyên từ một domain khác mà không vi phạm Same-Origin Policy, một chính sách an ninh của trình duyệt web. CORS cho phép một domain khác (origin) cung cấp dịch vụ tài nguyên (như API) cho các domain khác để sử dụng trong các ứng dụng web và di động. CORS được sử dụng rộng rãi trong các ứng dụng web hiện đại để cho phép giao tiếp giữa các domain khác nhau.

Câu 27: **Sự khác biệt chính giữa RequestMapping và GetMapping là gì?**

**RequestMapping có thể được sử dụng với GET, POST, PUT và nhiều phương thức yêu cầu khác bằng cách sử dụng thuộc tính phương thức tại annotation. Trong khi GetMapping chỉ là một phần mở rộng của RequestMapping, giúp bạn cải thiện độ rõ ràng của các yêu cầu**

**Câu 28: Transaction trong spring**

**Có nghĩa là giao dịch là khái niểm đảm boả tính nhất quán và an toàn của các thao tác trên cơ sỡ dự liệu. Spring cung cấp 1 cơ chế quản lý giao dịch mãnh mẽ và linh hoạt thông qua Spring Transaction Management**

**Câu 29: Thư viện và framework được sử dụng để kết nối cơ sở dữ liệu (Database) với Java:**

1. **JDBC (Java Database Connectivity): Đây là một API cung cấp các lớp và giao diện để kết nối và thao tác với cơ sở dữ liệu quan hệ thông qua SQL. JDBC là một tiêu chuẩn của Java để kết nối cơ sở dữ liệu và được hỗ trợ bởi hầu hết các hệ quản trị cơ sở dữ liệu.**
2. **Hibernate: Hibernate là một framework ORM (Object-Relational Mapping)phổ biến trong cộng đồng Java. Nó cung cấp cơ chế ánh xạ đối tượng vào cơ sở dữ liệu quan hệ. Hibernate giúp giảm sự phức tạp và công việc lập trình khi làm việc với cơ sở dữ liệu và cung cấp các tính năng như lazy loading, caching, và transaction management.**
3. **Spring Data JPA: Là một phần của Spring Framework, Spring Data JPA cung cấp các tính năng dễ dùng để làm việc với cơ sở dữ liệu. Nó sử dụng JPA (Java Persistence API) để thực hiện các thao tác trên cơ sở dữ liệu. Spring Data JPA tự động tạo các câu truy vấn dựa trên tên phương thức và cung cấp cách tiếp cận mạnh mẽ và linh hoạt cho việc truy vấn dữ liệu.**

**Câu 27: Lợi ích của việc sử dụng Docker là gì?**

**Đóng gói ứng dụng, Độc lập và cô lập, Dễ di chuyển và triển khai, Tiết kiệm tài nguyên, Tích hợp và mở rộng.**

**Câu 28: Docker khác gì vời máy ảo (virtual machine)**

**Docker dùng để chứa các tài ngoài là các suystem, máy chứa tài nguyên để chạy 1 hệ điều hành.**

**Câu 29: Kubernetes là gì và tại sao nó quan trọng trong việc triển khai ứng dụng?**

**Kubernates là một nền tảng mã nguồn mở cho việc triển khai, quản lý và mở rộng ứng dụng dựa trên container.Kubernates giúo tự đingj hoá quá trình triển khai và quản lý các ứng dụng được đóng gói trong container trên một môi trường phân tán.**

**Câu 30:Các khai niệm quan trong trong kubernetes: Pod, Node, Deployment, Service, ReplicaSet.**

**Câu 28: Docker là gì và nó được sử dụng để làm gì?**

**Docker là một nền tảng containerization mã nguồn mở được sử dụng để đóng gói, vận chuyển và chạy các ứng dụng trong các container độc lập với nền tảng cơ sở hạ tầng. Docker giúp giảm sự phụ thuộc vào các môi trường cài đặt và cấu hình trên máy chủ, giúp đơn giản hóa việc phát triển, triển khai và quản lý ứng dụng. Bằng cách sử dụng Docker, các ứng dụng có thể được đóng gói và chạy trên bất kỳ máy tính nào có cài đặt Docker, từ máy tính cá nhân đến các môi trường điện toán đám mây.**

**Câu 29: Các khái niệm quan trọng trong Docker như image, container, và Dockerfile có ý nghĩa gì?**

1. **Image: Một image là một gói phần mềm đóng gói tất cả các thành phần cần thiết để chạy một ứng dụng nhất định, bao gồm cả các thư viện, tập lệnh, mã nguồn, tệp cấu hình, vv. Các image được sử dụng để tạo ra các container.**
2. **Container: Một container là một thực thể thực thi của một image. Nó chứa tất cả các thành phần cần thiết để chạy ứng dụng, bao gồm cả các quy trình, hệ điều hành, các tệp dữ liệu, vv.**
3. **Dockerfile: Một Dockerfile là một tập tin văn bản chứa các chỉ thị và lệnh cần thiết để tạo ra một image Docker. Nó được sử dụng để định nghĩa môi trường của ứng dụng, bao gồm các tệp cấu hình, các yêu cầu phụ thuộc và các lệnh để cài đặt các gói phần mềm và thư viện cần thiết.**

**Câu 30: Ưu điểm của micro service?**

* **Nó làm cho sự phát triển nhanh chóng và dễ dàng.**
* **Tương thích với tất cả các container.**
* **Giảm thời gian sản xuất.**
* **Đây là một mô hình nhẹ hỗ trợ một ứng dụng kinh doanh lớn.**

**Câu 31: Spring Batch**

**Spring Boot Batch cung cấp khả năng tái sử dụng mã rất quan trọng khi làm việc với số lượng lớn bản ghi, bao gồm quản lý giao dịch, logging, skipping, thống kê xử lý công việc và khởi động lại công việc.**

**Câu 31: Giải thích về caching**

**Caching là bộ nhớ lưu trữ tạm thời dữ liệu được truy cập thường xuyên mà nếu lấy hoặc tính toán thì rất tốn kém.**

**Câu 32: Ưu điểm của Spring Boot**

* **Giúp bạn tạo ứng dụng độc lập, có thể bắt đầu bằng java.jar.**
* **Nó cung cấp cấu hình POM được xác định chính xác cho Maven.**
* **Cho phép bạn nhúng trực tiếp Undertow, Tomcat hoặc Jetty.**
* **Giúp bạn tự động cấu hình bất cứ khi nào có thể.**

**Câu 33: Quá trình bạn cần làm theo để chạy ứng dụng Spring Boot trên cổng tùy chỉnh là gì?**

**Để chạy ứng dụng, bạn cần đặt thuộc tính server.port trong application.properties.**

**Ví dụ: server.port = 8050.**

**Câu 34: Spring Boot starter là gì? Nó có hữu ích như thế nào?**

**Spring Boot có nhiều starters. Chúng là tập hợp tất cả các convenient dependency descriptors. Starter cho phép bạn include nhiều descriptors trong file pom.xml (Maven) hoặc build.gradle (Gradle).**

**Câu 35: Spring Boot Actuator là gì?**

**Spring Boot Actuator cho phép bạn theo dõi và quản lý ứng dụng của mình khi bạn muốn đưa nó vào sản xuất. Nó giúp bạn kiểm soát ứng dụng của mình bằng cách sử dụng các điểm cuối HTTP.**

**Câu 36: Làm thế nào bạn có thể truy cập một giá trị được xác định trong ứng dụng? File properties trong Spring Boot là gì?**

**Sử dụng annotation @Value để truy cập các properties được xác định trong file application – properties**

**Câu 37: Embedded containers support by Spring là gì?**

**Spring Boot hỗ trợ ba vùng chứa nhúng chính:**

* **Tomcat.**
* **Jetty.**
* **Undertow.**

**Theo mặc định, nó sử dụng Tomcat làm vùng chứa nhúng.**

**Câu 38: @pathVariable là gì?**

**Annotation @PathVariable giúp bạn trích xuất thông tin trực tiếp từ URL.**

**Câu 39: Swagger2 là gì?**

**Swagger được sử dụng để mô tả cấu trúc của các API. Swagger 2 là một dịch vụ mã nguồn mở được cung cấp trong Spring Boot, giúp máy tìm ra cấu trúc của các API như dịch vụ Web RESTful dễ dàng hơn.**

**Câu 40: Các environments(môi trường) khác nhau để phát triển ứng dụng doanh nghiệp là gì?**

* **Dev**
* **QA**
* **Stage**
* **Production**

**Câu 41: Ý nghĩa của Aspect-Oriented Programming (AOP) là gì?**

**Aspect-Oriented Programming (AOP) bổ sung Object-Oriented Programming (OOP) nhằm mục đích tăng tính modularity. AOP chia logic chương trình thành nhiều phần khác nhau, được gọi là concerns(mối quan tâm).**

**Câu 42: Handle exception trong Spring Boot như thế nào?**

**[Spring Boot](https://hocspringboot.net/2021/04/15/spring-boot-swagger2/) cung cấp một cách rất hữu ích để xử lý các ngoại lệ bằng cách sử dụng annotation @ControllerAdvice.**

**Câu 43: Apache Kafka**

**Apache Kafka là một open-source messaging platform. LinkedIn phát triển nó. Apache Kafka cho phép người dùng xây dựng các ứng dụng phân tán và xử lý các nguồn cấp dữ liệu thời gian thực. Kafka thích hợp cho cả nhắn tin ngoại tuyến và trực tuyến.**

**Câu 44: Giải thích về Spring MVC.**

**Nó là một web application framework truyền thống giúp bạn xây dựng một ứng dụng web. Framework này tương tự như framework Struts.**

**Câu 45: Bạn hiểu gì về ý nghĩa của aspect(khía cạnh)?**

**Nó là một tập hợp các API cung cấp các yêu cầu xuyên suốt.**

**Câu 46: Q**

**Database SQL**

1. **CHECK Constraint - Ràng buộc CHECK là gì?**

Một ràng buộc CHECK được sử dụng để giới hạn các giá trị hoặc loại dữ liệu có thể được lưu trữ trong một cột. Nếu bản ghi không đáp ứng được điều kiện này, thì sẽ không được lưu trữ vào trong bảng.

1. Primary Key (Khóa chính) là gì?

Một ràng buộc (constaint) Primary Key là một mã định danh duy nhất cho một row trong bảng cơ sở dữ liệu.

Mỗi bảng nên có một ràng buộc primary key để xác định duy nhất từng row và chỉ một ràng buộc primary key có thể được tạo cho mỗi bảng. Các ràng buộc primary key được sử dụng để thực thi tính toàn vẹn của thực thể.

1. Sự khác biệt giữa các lệnh TRUNCATE, DELETE và DROP?

DELETE xóa một hoặc tất cả các hàng từ một bảng dựa trên điều kiện và có thể được phục hồi lại.

TRUNCATE xóa tất cả các hàng từ một bảng bằng cách phân bổ các trang bộ nhớ và không thể phục hồi lại

DROP xóa hoàn toàn một bảng từ cơ sở dữ liệu.

1. Thứ tự của SQL SELECT?

Thứ tự các mệnh đề SQL SELECT là: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY. Trong đó SELECT và FROM là bắt buộc.

1. Trường dữ liệu BOOLEAN có giá trị nào?

Đối với trường dữ liệu BOOLEAN, có hai giá trị: 1 (TRUE) và 0 (FALSE).

1. Sự khác nhau giữa UNIQUE và PRIMARY KEY constraints là gì?

Một bảng có thể chỉ có một PRIMARY KEY và có thể không có hoặc có mộ hay nhiều UNIQUE keys.

PRIMARY KEY không thể chứa giá trị Null , UNIQUE có thể chứa giá trị Null.

1. Trigger là gì ?

Trigger là một thủ tục dược thực thi từ phía máy chủ CSDL khi một sự kiện bảng xảy ra (Chèn, cập nhật hoặc xóa lệnh thực hiện đối với một bảng cụ thể) .

1. View là gì?Lợi ích của Views là gì?

Một khung nhìn là một bảng ảo chứa dữ liệu từ một hoặc nhiều bảng. Lượt xem hạn chế quyền truy cập dữ liệu của bảng bằng cách chỉ chọn các giá trị được yêu cầu và thực hiện các truy vấn phức tạp một cách dễ dàng.

Ưu điểm của Views: Chế độ xem hạn chế quyền truy cập vào dữ liệu vì chế độ xem có thể hiển thị các cột được chọn từ bảng. Có thể sử dụng chế độ xem để truy vấn các kết quả tìm kiếm phức tạp.

1. Cổng TCP / IP nào mà SQL Server chạy?

Mặc định SQL Server chạy trên cổng 1433.

1. Thủ tục lưu trữ (stored procedure) là gì?

Một thủ tục lưu trữ là một tập hợp các truy vấn SQL có thể lấy đầu vào và gửi lại đầu ra.

1. ROWID nghĩa là gì?

Đó là một cột giả dài 18 ký tự gắn liền với mỗi hàng của một bảng.

1. Một bảng có thể có nhiều hơn một khoá ngoại?

Đúng, một bảng có thể có nhiều khóa ngoài và chỉ có một khóa chính.

1. Hãy định nghĩa về một temp table?

Temp table (bảng tạm) là một cấu trúc lưu trữ tạm thời. Điều đó nghĩa là gì? Về cơ bản, bạn có thể sử dụng một Temp table để lưu trữ dữ liệu tạm thời để bạn có thể thao tác và thay đổi nó như một table trong database.

1. View có chứa dữ liệu không?

Không, View là cấu trúc ảo.

1. DML và DDL là gì?

DML là viết tắt của Ngôn ngữ Thao tác Dữ liệu ( Data Manipulation Language): INSERT, UPDATE và DELETE là các câu lệnh DML.

DDL là viết tắt của Ngôn ngữ Định nghĩa Dữ liệu (Data Definition Language): CREATE, ALTER, DROP, RENAME là các câu lệnh DDL.

1. IDENTITY trong SQL là gì?

Một cột IDENTITY trong SQL sẽ tự động sinh ra các giá trị số tự tăng. Có thể định nghĩa giá trị bắt đầu và gia tăng của cột nhận dạng.

1. Index là gì?

Index (Chỉ mục) là bảng tra cứu đặc biệt mà Database Search Engine có thể sử dụng để tăng nhanh thời gian và hiệu suất thực hiện các truy vấn. Index có thể được tạo ra trên một hoặc nhiều cột của một bảng.

1. Định nghĩa JOIN và các loại JOIN khác nhau?

Từ khóa JOIN được sử dụng để nạp dữ liệu từ hai hay nhiều bảng liên quan.

Khi cần truy vấn các cột dữ liệu từ nhiều bảng khác nhau để trả về trong cùng một tập kết quả, cần sử dụng từ khóa "JOIN"

Các loại JOIN là:

INNER JOIN (Hoặc JOIN)

LEFT OUTER JOIN (Hoặc LEFT JOIN)

RIGHT OUTER JOIN (Hoặc RIGHT JOIN)

FULL OUTER JOIN (Hoặc OUTER JOIN)

CROSS JOIN

SELF JOIN

1. Làm thế nào để lấy ra các hàng ngẫu nhiên từ một bảng?

Sử dụng mệnh đề SAMPLE chúng ta có thể chọn các hàng ngẫu nhiên. Ví dụ: SELECT \* FROM table\_name SAMPLE (10);

1. SQL là gì?

Viết tắt của Structured Query Language – ngôn ngữ truy vấn cấu trúc. Nó được thiết kế để quản lý dữ liệu trong một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System). SQL là ngôn ngữ cơ sở dữ liệu, được dùng để tạo, xóa, lấy các hàng và sửa đổi các hàng.

1. Câu lệnh để chọn tất cả bản ghi từ table?

Cú pháp: Select \* from table\_name

1. Cú pháp để thêm bản ghi vào 1 bảng là gì?

Sử dụng cú pháp INSERT để thêm bản ghi vào 1 bảng

Ví dụ: INSERT into table\_name VALUES (value1, value2,…)

1. Làm thế nào để bạn thêm 1 cột vào 1 bảng?

Để thêm một cột khác vào bảng, sử dụng cú pháp:

ALTER TABLE table\_name ADD (column\_name)

1. Xác định câu lệnh Delete SQL

Câu lệnh này được sử dụng để xóa hàng hoặc các hàng từ một bảng dựa trên điều kiện được chỉ định.

Cú pháp:

DELETE FROM table\_name<br>WHERE<Condition>

1. Khóa ngoại (FOREIGN KEY) là gì?

Khi một trường khóa chính của một bảng được thêm vào các bảng có liên quan để tạo ra trường phổ biến có liên quan đến 2 bảng, nó được gọi là khóa ngoại trong các bảng khác. Các ràng buộc khóa ngoại thực thi toàn vẹn tham chiếu.

1. Giao dịch (transaction) là gì?

Mỗi giao dịch là một dãy mã chạy trên cơ sở dữ liệu.

1. Sự khác nhau giữa UNIQUE và PRIMARY KEY constraints là gì?

Một chỉ có thể có 1 PRIMARY KEY nhưng có thể không có hoặc có nhiều UNIQUE KEY.

PRIMARY KEY không thể chứa giá trị NULL, UNIQUE thì có thể có giá trị NULL

1. Khóa tổng hợp (Composite primary key) là gì?

Khóa chính được tạo trên nhiều cột được gọi là khóa chính tổng hợp.

1. Subquery là gì?

Truy vấn con (còn được gọi là truy vấn phụ hay truy vấn lồng nhau) là một truy vấn bên trong truy vấn SQL khác và được nhúng bên trong mệnh đề WHERE.

1. Tối ưu hóa truy vấn là gì?

Là một quá trình, trong đó hệ thống cơ sở dữ liệu so sánh các chiến lược truy vấn khác nhau và chọn truy vấn với chi phí thấp nhất.

1. Làm thế nào chúng ta có thể tránh trùng lặp hồ sơ trong một truy vấn?

Sử dụng từ khóa INSTINCT thì việc sao chép hồ sơ trong một truy vấn có thể tránh được.

1. Giải thích sự khac nhau giữa đổi tên (Rename) và Bí danh (Alias)?

Đổi tên là một tên thường xuyên cho một bảng hoặc cột. Bí danh là tên tạm thời cho một bảng hoặc cột.

1. Schema là gì?

Biểu đồ là tập hợp các đối tượng cơ sở dữ liệu của người dùng

1. Bảng là gì?

Một bảng là đơn vị cơ bản của lưu trữ dữ liệu trong hệ thống quản lý cơ sở dữ liệu. Dữ liệu bảng được lưu trữ trong hàng và cột.

1. Chế độ xem có thể dựa trên một chế độ xem khác không?

Chế độ xem dựa trên một chế độ xem khác

1. Sự khác nhau giữa mệnh đề Having và mệnh đề Where?

Cả hai mệnh đề đều chỉ định điều kiện tìm kiếm nhưng mệnh đề Having chỉ được sử dụng với câu lệnh SELECT và thường được sử dụng với mệnh đề GROUP BY. Nếu không có mệnh đề GROUP BY thì Having sử dụng giống mệnh đề WHERE.

1. Sự khác nhau giữa bảng tạm cục bộ (Local) và bảng tạm toàn cầu (Global) là gì?

Một bảng tạm cục bộ tồn tại trong một kết nối. Khi kết thúc kết nối thì bảng tạm này tự động được xóa. Tên của bảng tạm Local được bắt đầu bằng ký tự #

Một bảng tạm toàn cầu tồn tại vĩnh viễn trong db nhưng các hàng của nó biến mất khi kết nối đóng lại. Tên bảng tạm Global được bắt đầu bằng ##

**Gradle và Maven**

1. **Hai công cụ quản lý dự án phổ biến trong phát triển phần mền Java, cả hai đều hỗ trợ việc quản lý các dependecy, build, test, triển khai. Cả hai để có khả năng quản lý phụ thuộc, tạo ra gói triển khai, chạy các tác vụ kiểm tra và triển khai dự án.**

**Gradle:**

* **Gradle là một công cụ xây dựng linh hoạt và mạnh mẽ, cho phép định nghĩa các tác vụ tùy chỉnh và quản lý dự án phức tạp.**
* **Gradle sử dụng cú pháp DSL (Domain-Specific Language) để định nghĩa các tác vụ xây dựng và quản lý dự án, giúp viết mã ngắn gọn và dễ đọc.**
* **Gradle hỗ trợ đa nền tảng, cho phép xây dựng dự án trên nhiều ngôn ngữ và môi trường.**
* **Gradle có khả năng tự động tải xuống các phụ thuộc từ các kho lưu trữ khác nhau như Maven Central, JCenter, và từ xa.**

**Maven:**

* **Maven là một công cụ quản lý dự án phổ biến và được sử dụng rộng rãi trong cộng đồng Java.**
* **Maven sử dụng cú pháp XML để định nghĩa cấu hình và quản lý dự án.**
* **Maven có một số quy ước và cấu trúc mặc định, giúp giảm sự phức tạp và đảm bảo tuân thủ chuẩn phát triển.**
* **Maven sử dụng các repository Maven để lưu trữ các phụ thuộc và tải về các thư viện từ các nguồn tài nguyên đã cấu hình.**

**Microservice**

1. Microservice là gì? Giải thích kiến trúc và nguyên tắc cơ bản của Microservice.

Microservice là một kiến trúc phần mền mà ứng dụng được phân tách thành phần nhỏ, độc lập và tự quản lý.Các nguyên tắc cơ bnar là độc lập, phân tách chức năng, giao tiếp thông qua giao thức chuẩn, quản lý độc lập, tự phục hồi và khả năng chịu lỗi, quản lý trạng thái riêng.

1. Microservices sử dụng giao thức nào để giao tiếp giữa các service?

Restful API, Messaging-based, gRPC, GraphQL

1. API Gateway vai trò?

Vai trò chính : Điều phối, bảo mật, quản lý luồng, Caching, giám sát và theo dõi

**Database NOSQL**

1. **Cơ sở dữ liệu NoSQL là gì?**

Cơ sở dữ liệu NoSQL là một Hệ thống quản lý dữ liệu không quan hệ (non-relational Data Management System) có lược đồ (schema) linh hoạt. Nó dễ mở rộng. Mục đích chính của việc sử dụng cơ sở dữ liệu NoSQL là dành cho các kho dữ liệu phân tán với nhu cầu lưu trữ dữ liệu lớn.

Collection (tương đương với table):

Document (tương đương với row):

Field (tương đương với column):

1. Những hạn chế của CSDL NoSQL

No Schema, Sự nhất quán, sự phụ thuộc,

**Cấu trúc dữ liệu và giải thuật**

1. Một sốphức tạp của thuật toán:

Tìm kiếm nhị phân O(logn), Tìm kiếm tuần tự O(n), Sắp xếp nổi bọt Duyệt 2 ma trận O(n^2), Fibonaci O(2^n)

**Testing**

**1. Unit Testing (Kiểm thử đơn vị)**

Unit testing, còn được gọi là Component testing, mức độ kiểm thử phần mềm này thường do Developer phụ trách, họ sẽ đi kiểm tra các module, các hàm, các phương thức, các lớp,… mà họ viết ra nhằm gia tăng sự tin cậy cho các chức năng mà mình viết.

Kiểm thử đơn vị nằm trong phạm vi của Kiểm thử hộp trắng (White-box testing), tức là kiểm tra code bên trong của một chức năng hoặc hệ thống để xem chức năng hoặc hệ thống đó được viết đúng chuẩn code hay chưa, đoạn code đó khi chạy hiệu năng có tốt hay không, có nhanh hay không, có tốn tài nguyên hay không,..

**2. Integration Testing (Kiểm thử tích hợp)**

Kiểm thử tích hợp là kiểm thử sự tương tác giữa các chức năng với nhau trong hệ thống và được thực hiện bởi Tester.

3. System Testing (Kiểm thử hệ thống)

Kiểm thử hệ thống là kiểm thử một hệ thống đã hoàn thành, đã tích hợp đầy đủ các chức năng nhằm kiểm tra xem hệ thống phần mềm đó có đáp ứng đầy đủ các yêu cầu chức năng theo bản đặc tả yêu cầu phần mềm (SRS) hay không. Người thực hiện test level này thường là Tester.

4. Acceptance Testing (Kiểm thử chấp nhận)

Mức độ kiểm thử phần mềm cuối cùng chính là Acceptance Test (Kiểm thử chấp nhận) – kiểm tra xem hệ thống có đáp ứng đúng nhu cầu và mong đợi của khách hàng hay không.

Acceptance test gồm 2 loại kiểm thử là:

Alpha Test, người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm, lập trình viên sẽ ghi nhận các lỗi hoặc phản hồi, và lên kế hoạch sửa chữa.

Beta Test, phần mềm sẽ được gửi tới cho người dùng để kiểm thử ngay trong môi trường thực, lỗi hoặc phản hồi cũng sẽ gửi ngược lại cho lập trình viên để sửa chữa.

1. Kiểm tra hộp đen (Black box testing) là một phương pháp kiểm thử phần mềm mà việc kiểm tra các chức năng của một ứng dụng không cần quan tâm vào cấu trúc nội bộ hoặc hoạt động của nó.
2. Kiểm thử hộp trắng (While box test) là phương pháp thử nghiệm phần mềm, trong đó các thiết kế, cấu trúc giải thuật bên trong, và việc thực hiện các công việc đều được biết đến

**Mô hình agile-scrum**

**7. Bạn có hiểu Srum?**

**Scrum là một framework giúp các nhóm nhanh nhẹn làm việc cùng nhau để phát triển, cung cấp và duy trì sản phẩm phức tạp trong thời gian ngắn nhất. Sản phẩm do nhóm scrum cung cấp trong thời gian ngắn nhất này được gọi là sprint.**

8. Các vai trò khác nhau trong Scrum là gì?

Chủ sản phẩm(Product Owner)

Điều phối viên(Scrum Master)

Nhóm phát triển(Development Team)

19. Sự khác biệt giữa cách làm việc agile và truyền thống là gì?

Cách phát triển truyền thống là theo trình tự nơi thiết kế -> phát triển -> thử nghiệm được thực hiện, trong khi, trong phát triển Agile, tất cả điều này được thực hiện trong mỗi sprint.

GraphQL và Rest API

* GraphQL là một ngôn ngữ truy vấn dữ liệu và một runtime để thực thi các truy vấn này trên các dịch vụ web. GraphQL cho phép khách hàng xác định chính xác dữ liệu cần lấy từ một API duy nhất thông qua các câu truy vấn. Khách hàng có quyền lựa chọn các trường dữ liệu cần truy vấn và cũng có thể truy vấn dữ liệu từ nhiều nguồn khác nhau trong một lời gọi duy nhất:
* Truy vấn linh hoạt
* Tương tác dữ liệu từ nhiều nguồn
* Documentation và hỗ trợ mạnh mẽ
* Versioning linh hoạt
* Rest API: là một kiểu giao diện lập trình ứng dụng (API) được sử dụng để giao tiếp và trao đổi dữ liệu giữa các hệ thống và ứng dụng phân tán. Nó dựa trên các nguyên tắc và quy ước thiết kế của kiến trúc REST
* Giao tiếp stateless
* Sử dụng phương thức HTTP
* Dữ liệu truyền qua các địa chỉ URL
* Sử dụng các định dạng dữ liệu như JSON hoặc XML

Agile/Scrum, Kanban và Waterfall là ba phương pháp quản lý dự án phổ biến trong lĩnh vực phát triển phần mềm và quản lý công việc. Dưới đây là cách phân biệt giữa chúng:

Waterfall (mô hình thác nước):

Waterfall là một phương pháp phát triển phần mềm truyền thống và tuần tự.

Dự án được chia thành các giai đoạn tuần tự như yêu cầu, thiết kế, xây dựng, kiểm thử và triển khai.

Mỗi giai đoạn chỉ bắt đầu khi giai đoạn trước đó hoàn thành.

Waterfall thích hợp cho các dự án có yêu cầu rõ ràng và ổn định, không đòi hỏi sự thay đổi lớn trong quá trình phát triển.

Agile/Scrum:

Agile là một triết lý và phương pháp quản lý dự án linh hoạt.

Scrum là một trong những framework thực hiện Agile phổ biến.

Agile tập trung vào việc cung cấp giá trị cho khách hàng nhanh chóng và đáp ứng linh hoạt với sự thay đổi.

Scrum chia dự án thành các đợt phát triển ngắn gọi là "Sprint" (thường từ 1-4 tuần).

Scrum sử dụng các sự kiện như Sprint Planning, Daily Standup, Sprint Review và Sprint Retrospective để quản lý công việc và sự tiến bộ.

Agile/Scrum phù hợp cho các dự án có yêu cầu không chắc chắn và có khả năng thay đổi, với sự tương tác thường xuyên với khách hàng.

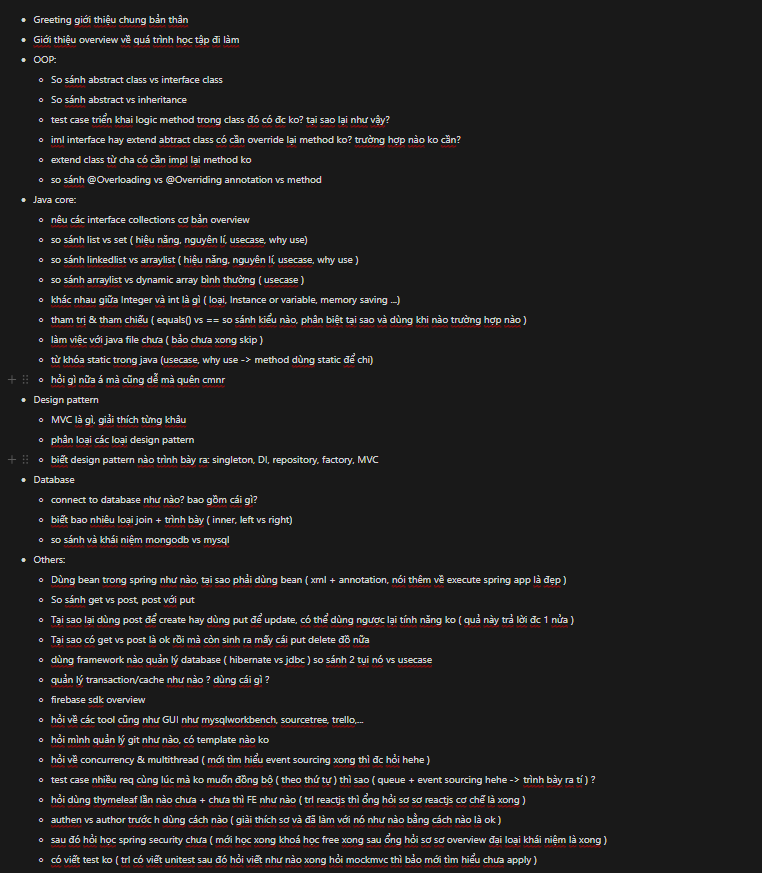
Kanban:

Kanban là một hệ thống quản lý công việc được biểu thị bằng các bảng và các thẻ (cards).

Các công việc được phân chia thành các pha trạng thái khác nhau, ví dụ: "đang chờ xử lý", "đang thực hiện", "đã hoàn thành".

Công việc di chuyển qua các trạng thái dựa trên nguyên tắc giới hạn công việc đồng thời và tối đa hóa hiệu suất làm việc.

Kanban giúp quản lý quá trình làm việc hiệu quả, tăng sự tương tác và theo dõi công việc một cách rõ ràng.



1. So sánh abtract(trường tượng) và inheritance (kế hoạch)

- Abstract tập trung vào định nghĩa hành vi chung bắt buộc lớp con phải thực hiện.

- Inheritance dùng để tái sử dụng và mở rộng chức năng từ lớp cha.

1. Có implemet interface hay abstract class có cần override lại method không ? trường hợp nào không cần?

- Có. Khi bạn implement một interface hoặc kế thức một abstract class, bạn bắt buộc phải override lại các phương thức abtract. Nếu không, liớp bạn cũng phải khai báo là abstract.

- Trường hợp nào không cần override lại phoương thức: Interface (default method), Abtract class(không cần override nếu lớp con kế thừa cũng là abstract class)

1. Có cần implemetn lại method khi extend từ class cha không?

- Không cần bắt override các phương thúc đã đầy đủ trong class cha. Tuy nhiên, vẫn có thể override lại method của lớp cha, và bắt buộc override lại method abstract của lớp

1. Overloading và Overrideing

Overloading : Định nghĩa nhiều phương thức cùng tên khác tham số trong cùng 1 lớp.

Overriding: Định nghĩa lại phương thức của lớp cha trong con để thay đổi hành vi.

1. Interface collection

Collection

│

├── List

│ ├── ArrayList

│ ├── LinkedList

│ ├── Vector

│ └── Stack

│

├── Set

│ ├── HashSet

│ ├── LinkedHashSet

│ └── SortedSet

│ └── NavigableSet

│ └── TreeSet

│

└── Queue

├── LinkedList

├── PriorityQueue

└── Deque

├── ArrayDeque

└── LinkedList

Map

├── HashMap

├── LinkedHashMap

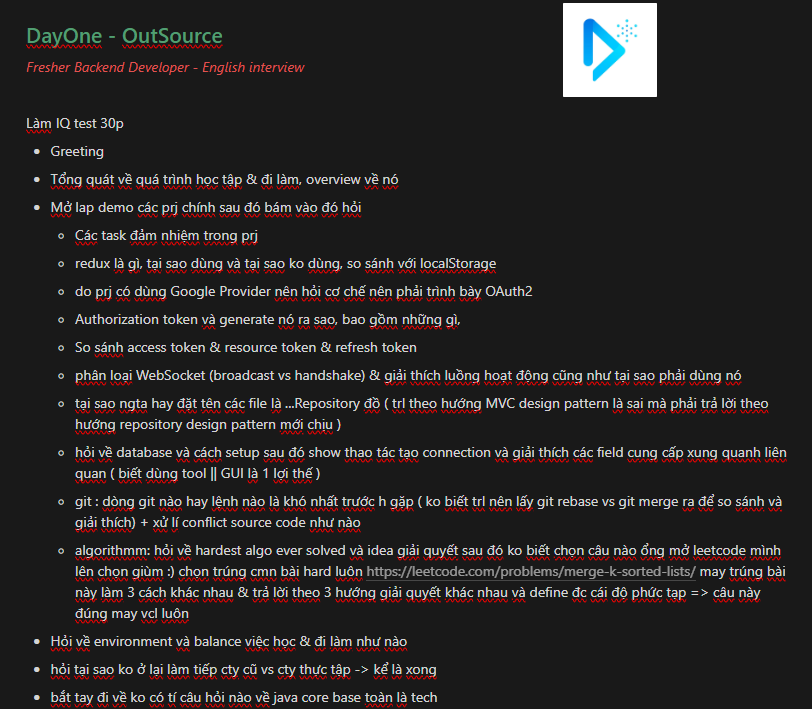
├── SortedMap

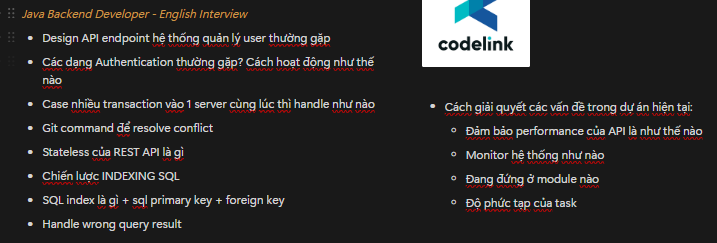
│ └── NavigableMap

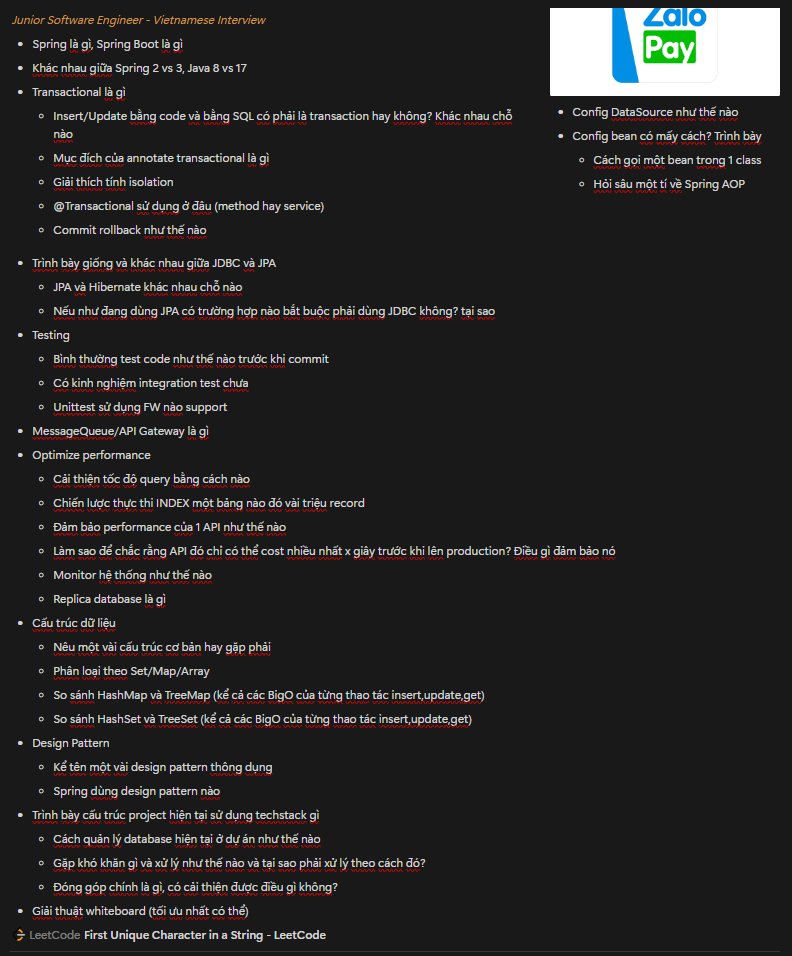
│ └── TreeMap

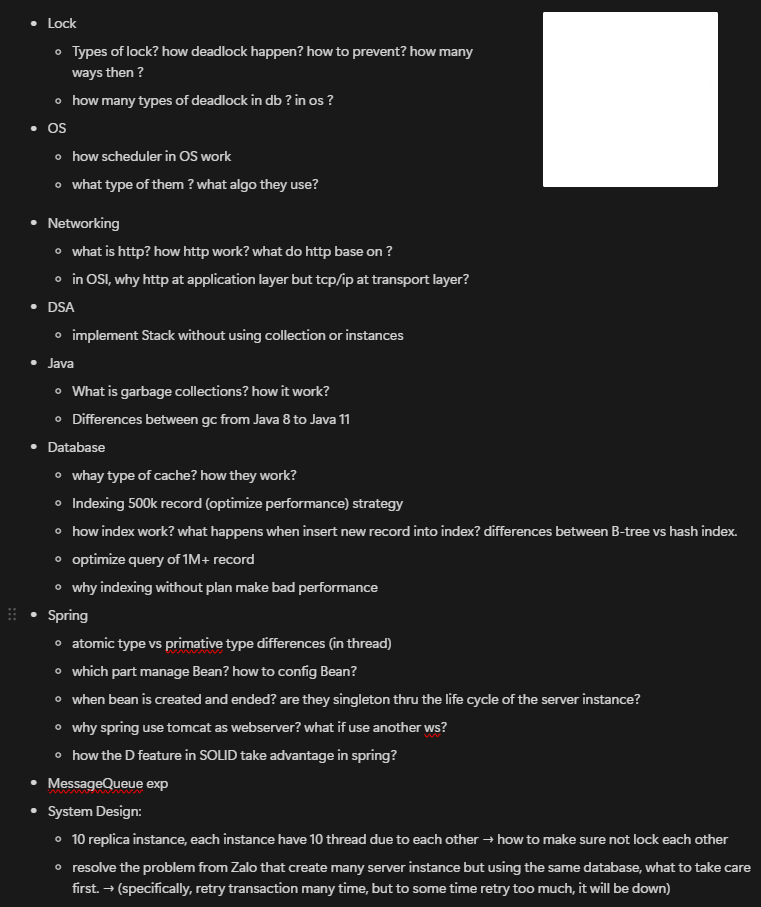
└── ConcurrentHashMap

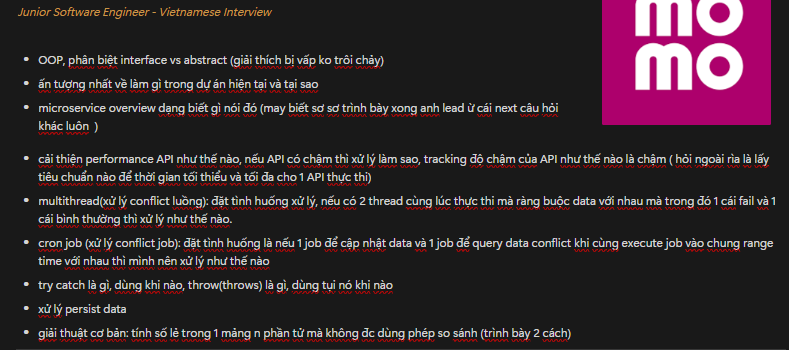












1. OOP

OOP là 1 phương pháp lập trình hướng đốit tựng dựa trên đổi khái niệm đối tượng đẻ tổ chức và quản lý mã nguồn.

1. phân biệt interface và abstract

- 1 lớp chỉ có thể kế thừa từ một abstract class, còn một lớp có thực hiện(implement) nhiều interface.

- Đối với abstract thì khoá abstract sử dụng cho lớp và phương thức, còn interface thì sử dụng từ khoá interface.

- Abstract có thể chức các biến instance, phương pháp cụ thể và phương thức trừu tượng. Interface chỉ chứa hằng số (static final) và pphương thức (abstract, default, static).

- Abstract có thể có contrructor còn inteface thì không.

1. Overview Microservice

Microcervices là 1 kiến trúc phần mềm trong đó ứng dụng được chia thành các dịch vụ nhỏ, độc lập. Mỗi service đảm nhận 1 chức năng cụ thể và giao tiếp với service khác thông qua giao thúc(http/rest, grRPC, hoặc messaging)

1. Cải thiện performance API

- 1. Tối ưu hoá cơ sở hạ tầng(Sử dụng load balancer, sử dụng CND, cải thiện tài nguyên phân cứng)

- 2. Giảm độ trễ(Latency): Caching, Compression, Http2/ hoặc Http/3, Keep-alive connections

- 3. Tối ưu hoá truy vấn và dữ liệu(Paging and Filtering; tối ưu cơ sở dữ liệu(index, giảm join, Database Caching, Bulk Operations)

- 4 Tối ưu hoá mã nguồn API (Thread Pool(Java Executor) hoặc Async ; Mimize Overhead; Connection Pooling)

- 5.Giám sát và theo dỏi ( Logging and Monitoring (ELK stack; Phân tích Bottleneck; Rate Limiting

- 6. Cải thiện giao tiếp dữ dịch vụ(cgiao thức nhanh hơn, message queues, Circuit Breaker)

- 7. Kiểm tra hiệu suât(Testing)

1. Multithread và xử lý conflict luông

Multithread là khả năng chạy nhiều luồng (threads) trong một tiến trình (proccess) để thực hiện các tác vụ đồng thời, giúp tận dụng tối đa tài nguyên CPU, cải thiện hiệu suất của ứng dụng.

- 1. Xung đột giữa các luồng (Thread Conflict)

Khi nhiều luồng truy cập hoặc sửa đổi cùng 1 một tài nguyên chung(như biến, file, hoặc database) mà không được đồng bộ đúng cách, sẽ xảy ra race condition hoặc data inconsisetency. Cách xử lý xung đột là sử dụng từ khoá synchronized (hoặc ReentrantLock)

1. Try catch là gì và dùng khi nào

Try catch là 1 có chế xử lý ngoại lệ(exception handling) để bắt và xử lý các lỗi xảy ra trong runtime mà không làm gián đoạn chương trình.

1. throw là gì

throw được dùng để ném ngoại lệ(exception) trong chương trình. Khi từ khoá throw, chương trình sẽ tạo ra một exception và chuyển nó cho trình xử lý phù hợp(try-catch)

1. throws là gì/

throws được sử dụng trong khai báo phương thức để chỉ ra rằng các phương thức có thể ném ngoại lệ (Exception). Phương thức không xử lý lỗi mà chuyển trách nhiệm xử lý cho phương thực gọi nó.

1. Xử lý persit data

Kết nói tới hệ thống lưu trữ( CSDL, File), chuyển đổi dữ liệu (Seriaalization/Deserialization; Mapping). Thảo táo luư trữ (CRUD). Tối ưu hoá. Đảm báo tính nhất quán dữ liêuk (ACID(Có quan hệ), BASE(phi quan hệ

1. giải thuật cơ bản tính số lẽ trong 1 mang n phần tử không dùng phép so sánh.

sử dụng tính chất phép toán modulo(%) và