**CHECK LIST CÁC CÂU HỎI CƠ BẢN VỀ JAVA**

**1. Câu hỏi cơ bản về Java:**

a. Java ra đời năm nào?

Java ra đời năm 1995

b. Giới thiệu đơn giản về JVM, JRE, JDK.

+JVM là viết tắt của Java Virture machine. Nó là môi trường thực thi cho các chương trình Java. Cho phép chương trình được thực thi trên nhiều hệ điều hành khác nhau

+JRE là viết tắt của Java Runtime Enviroment là một phần mềm chạy trên hệ điều hành máy tính và cung cấp các thư viên và các tài nguyên khác mà một chương trình Java cần có để chạy.

+JDK(java Development Kit) là bộ công cụ cung cấp môi trường phát triển để viết và chạy chương trình Java. Nó bao gồm JRE + JVM và các công cụ hỗ trợ khác

**2. Biến, mảng trong Java:**

a. Biến trong Java là gì?

Biến là tên của một vùng nhớ được khai báo

b. Mảng trong Java là gì? Cách khai báo một mảng trong Java?

Mảng là kiểu dữ liệu có cấu trúc bao gồm tập hợp cố định số lượng cố định các phân tử có số lượng cố định

-Cách khai báo mảng

[Kiều\_dũ\_liệu] tên\_mảng[];

[Kiểu\_dữ\_liệu][] tên mảng;

**3. Các kiểu dữ liệu trong Java:**

a. Liệt kê các kiểu dữ liệu nguyên thủy (primitive) trong Java.

-Có 8 kiểu dữ liệu nguyên thủy trong Java

+Kiểu số nguyên: byte,short, int, long

+Kiểu sô thực: float, double

+Kiểu ký tự: Char

+Kiểu logic: boolean

b. Giải thích khái niệm kiểu Wrapper trong Java và đưa ra ví dụ.

Trong Java ứng với mỗi kiểu dữ liệu cơ sở thì chúng ta sẽ có 1 kiểu dữ liệu Wrapper, wrapper gói các kiểu dữ liệu cơ sở vào 1 đối tượng của nó

VD tương ứng với kiểu Boolean thì kiểu wrapper cũng là boolean

c. So sánh primitive với wrapper.

Ưu viêt của Wrapper so với primitive

+ Có thể sử dụng lớp Wrapper trong collection

+ Có thể sử dụng các biến có giá trị NULL

+ Với Wrapper chúng ta có thể nhận được NullPointerException khi một cái gì đó đang được thiết lập không chính xác.

+ Với kiểu nguyên thủy các dữ liệu được khỏi tạo mặc định 0, False ... điều này có thể không chính xác

Hạn chế của wrapper so với kiểu nguyên thủy

+ Các kiểu dữ liệu nguyên thủy có thể nhanh hơn so với Wrapper tương ứng

+ Phép toán so sánh == có thể cho kết quả không như mong đợi do nó so sánh tham chiếu,

**4. String, StringBuilder, StringBuffer:**

a. Tại sao String là bất biến trong Java?

Hiệu năng: do các chuổi trong thực tế lặp lại. Thay vì tạo ra nhiều instance khác nhau cho một chuỗi thì sử dụng strong pool chỉ tạo ra 1 instance duy nhất

Bảo mật: khi sử dụng driver hay tạo một kết nối URL nào đó chung ta thường xuyên phải truyền đối tượng dưới dangj String. Nêu String không phải là immutable thì có thể gây ra vấn đề bảo mật là Pandora box

b. Làm thế nào để lưu trữ một đối tượng String trong bộ nhớ?

+ Sử dụng literal

+ Sử dụng từ khóa New

c. Sự khác nhau giữa String s = "abc"; và String s = new String("abc");?

- s = “abc” Giá trị lưu trong string pool. JVM xác đinh chuỗi đã tồn tại trong string pool chưa. Nêu đã tồn tại JVM sẽ tham chiếu dến string pool. Nếu chưa tồn tại sẽ JVM sẽ tạo ra Object chứa chuỗi này đưa vào strong pool rồi tham chiếu s đến

- String s = new String (“abc”) Giá trị lưu ở vùng nhớ Heap space. JVM luôn tạo ra 1 object trong bộ nhơ heap

d. So sánh sử dụng String, StringBuilder và StringBuffer trong Java.

+ String không thể thay đổi và không cho phép có class con

+ StringBuffer và StringBuilder có thể thay đổi

+ Khi sử lý văn bản nhiều luồng việc sử dụng StringBuffer tránh gây xung đột các luồng

StringBuilder > StringBuffer > String

**5. Ép kiểu:**

a. Khái niệm ép kiểu dữ liệu trong Java là gì? Đưa ra ví dụ.

Chuyển đổi một kiểu dữ liệu qua kiểu dữ liệu khác

VD: float c = 35.8f;

Int b = int(c);

Đầu tiên c sẽ được ép kiêu thành 35 rồi gán vào biến b

b. Khái niệm ép kiểu đối tượng trong Java là gì? Đưa ra ví dụ.

Là chuyên tham chiếu chứa đối tượng thuộc loại này thành tham chiếu đối tượng loại khác

+ép từ kiểu từ lớp con lên lớp cha(ngầm định thực hiện)

+ép kiểu từ lớp cha xuống lớp con

**6. Constructor:**

a. Constructor là gì trong Java?

Giống như là một khuôn mẫu để có thể tạo ra các đối tượng

b. Có mấy loại constructor?

Có 2 loại Constructor

+ Constructor mặc định(không đối số)

+ Constructor có tham số

c. Cách nào để tạo constructor nhanh trong IDE Eclipse hoặc Intellij?

Sử dụng tính năng tự động tạo constructor bằng phím tắt ALT + Insert

**7. Access modifier:**

a. Liệt kê và giải thích các phạm vi truy cập (access modifier) trong Java: private, default, protected, public.

+ Private: chỉ có thể truy cập từ class đó

+ default truy cập các class cùng gói

+ protected các class cùng gói và các lớp con

+ public được truy cập từ tất cả mọi nơi

b. Khi nào chúng ta sử dụng private, default, protected và public trong việc khai báo các thành viên lớp?

+private: khi dữ liệu quan trọng

+default: khi dữ liệu có thể được sử dụng ở các tính năng khác trong gói

+protected: khi có lớp khác sẽ kế thừa những phương thức của, thuộc tính của lớp này và có chỉ được sử dụng trong lớp kế thừa đó

**8. Non-access modifier:**

a. Liệt kê và giải thích các non-access modifier: abstract, static, final, synchronized, transient, volatile (khi kết hợp với field, method hoặc class).

- abstract:

+ Khi class đinh nghĩa với abstract, class khác không thể khởi tạo đối tượng cho class này

+ Đinh nghĩa các yêu cầu nhưng không thực thi và triển khai trong các class con

+ abstract không thể sử dụng với biến

+ static chỉ đinh này cho phép tạp ra các phương thức, thuộc tính dùng chung có thể thay đổi tư nhiều nơi

+final nó như gán chết giá trị, thuộc tính nào đó và không thể thay đổi

+ synchronized áp dụng cho method. tránh method bị truy câp cập cùng lúc từ nhiều luồng

+ transient sử dụng trong Serialization có tác dụng giữ trạng thái của biến lưu trong qua trinh Serialization

**9. Exception và xử lý exception:**

a. Khái niệm về exception trong Java là gì?

Là một trạng thái bất thường làm gián đoạn luông bình thường của chương trình

b. Làm thế nào để xử lý exception trong Java?

Sử dụng khối Try Catch

**10. Abstract và Interface:**

a. Khái niệm abstract class và abstract method trong Java là gì?

- abstract class là 1 lớp không thể khởi tạo trực tiếp mà chỉ làm lớp cơ sở cho lớp khác

- abstract method là phương thức không có thân hàm. Các phưng thức được ghi đè trong lớp con

b. Khái niệm interface trong Java là gì?

- Là một lớp trìu tựong định nghĩa các phương thức mà lớp con cần thực thi(không chưa thân hàm)

c. Sự khác nhau giữa abstract class và interface trong Java.

- Trong interface dữ liệu phải là hằng số, abstract class có thể là biến

- Trong interface phương thức chỉ có tên và không thực hiện. Trong abstract class có phương thức đầy đủ

-Interface hỗ trợ đa kế thừa còn abstract class thì không

**11. OOP trong lập trình:**

a. Khái niệm OOP (Object-Oriented Programming) trong lập trình là gì?

- Là phương pháp lập trình tiếp cận theo hướng chia chương trình thành các đối tượng và lớp và cách thức các lớp và đối tượng tương tác vơi nhau giup code dễ quản lý và bảo trì

b. Tại sao OOP được coi là một phương pháp lập trình mạnh mẽ?

- Giúp dễ chia chương trình vì mỗi đối tượng trong thực tế là một đối tượng

- Dễ tái sử dụng code

- Dễ dàng tìn lổi

**12. Đóng gói trong Java:**

a. Java triển khai tính đóng gói như thế nào?

Java triển khai tính đóng gói bẳng cách ẩn đi dữ liệu thông qua các chỉ định truy suất như private, protected

sử dụng hàm để truy cập vào dữ liệu

b. Getter và Setter trong Java được sử dụng để làm gì?

để truy cập vào dữ liệu trong đối tượng mà không phá vỡ tính đóng gói của phương pháp lập trinh hướng đối tượng

**13. Kế thừa trong Java:**

a. Khái niệm kế thừa trong Java là gì?

Là một lớp có thể kế thừa lại các phương thức và thuộc tính của lớp khác thay vì định nghĩa lại

b. Từ khoá this và super trong Java được sử dụng để làm gì?

this là tham chiếu đến đối tượng của lớp hiện tại

super là tham chiếu đến đối tượng của lớp cha

c. Liệt kê các loại kế thừa. Java có đa kế thừa hay không? Tại sao?

đơn kế thừa, đa kế thừa, kế thừa phân cấp

Java không có đa kế thừa vì để giảm thiểu phức tạp và đơn giản hóa ngôn ngữ

d. Sự khác nhau giữa đơn kế thừa, kế thừa nhiều cấp và kế thừa phân cấp trong Java.

Đơn kế thừa: kế thừa từ 1 lớp cha trục tiếp

Đa kế thừa: một lớp con kế thừa từ nhiều lớp cha

Kế thừa phân cấp: khi một lớp con kế thừa 1 lớp comn khác

**14. Đa hình trong Java:**

a. Đa hình trong compile time (overloading) là gì? Đưa ra ví dụ.

là 1 phương thức có cùng tên nhưng khác nhau về đối số

Public class DaHinh{

Public int sum(int a, int b){

return a + b;

}

Public float sum(float a, float b){

return a \* b;

}

Public static void main(String args[]){

System.out.println(sum(2,3))

}

} kế quả là 5 vì 2 số truyền vào là 2 số nguyên

b. Đa hình trong run time (overriding) là gì? Đưa ra ví dụ.

là gọi các phương thức đa được ghi đè các thuộc tính khác trong thời gian thực hiện chương trinh

Class Animal {

void run(){

System.out.println(“running”)

}

}

Class Lion extends Animal {

void run(){

system.out.println(“lion runnning”)

}

}

public static void main(String ảrgs[])

{

Animal a = new Lion()

b.run()

}

**15. Trừu tượng trong Java:**

a. Trừu tượng trong java là gì? Thể hiện như thế nào?

Trìu tượng trong java là tính trình ẩn chi tiết cách thức hoạt động của một chức năng và chỉ hiện tính năng cho người dùng

Có 2 cách để thể hiện

dùng abstract

dùng interface

**16. Collection:**

a. Collection trong Java được định nghĩa như thế nào?

là 1 interface trong java Framework để biểu diễn và quản lý nhóm các đối tượng.

b. Các thành phần chính của Collection là gì?

List, Set, Map và Queue

c. Hệ thống cấp bậc của Collection bao gồm những gì?

**17. Iterable và Iterator:**

a. Iterable là gì trong Java? Vai trò của nó là gì?

Iterable là một interface trong java đại diện cho bộ sưu tập có thể được lặp qua. nó chưa phưogn thức iterable() trả về 1 đối tượng iterable

vai trò: bất cứ lớp nào triển khai iterable cung cấp 1 phương thức iterable() tra về đối tượng interface. Đối tượng này cho hép lặp qua các phần tử 1 cách tuần tự

b. Làm thế nào để sử dụng Iterator để lặp qua các phần tử trong một Collection?

+B1; tạo collection

+B2: lấy iterator qua phương thức iterator()

+B3: dùng vòng lặp cùng với hasNext() và next()

**18. Interface List và các lớp cài đặt:**

a. Interface List trong Java được sử dụng để làm gì?

để làm 1 tập hơp các phần tử có thể truy cập theo chỉ số và có thể trùng lặp

b. Các lớp cài đặt của Interface List trong Java bao gồm những gì? Hãy nêu một số đặc điểm và sử dụng của mỗi lớp.

ArrayList sử dụng mảng động để lưu trữ dữ liệu và có thể truy cập ngẫu nhiên 1 phàn tử

LinkedList là danh sách liên kế có thể thêm sửa xóa các phần từ ngẫu nhiên

Vector tương tự ArrayList nhưng là threa-safe được sử dụng ít hơn và hiệu suất thấp hơn

c. Tại sao cần nhiều lớp triển khai interface List? Khi nào thì nên dùng triển khai nào? Tại sao?

vi có thể sử dụng cho nhiều tinh huống khác nhau. Nếu cần truy cập ngẫu nhiên dùng ArrayList, cần thêm sủa xóa ngẫu nhiên dùng LinkedList. nếu cần threa-safe sử dụng vector

d. Nêu chi tiết hiểu biêt về ArrayList, LinkedList. Nó khởi tạo, lưu trữ, thêm, sửa, xoá, ... các phần tử như thế nào?

ArrayList tạo ra 1 mảng động các ô nhớ có index. nó truy cập và từng ô nhớ qua index và thêm sủa xóa các giá trị tương ứng

kt: ArrayList()

LinkedList sử dụng danh sách liên kết kép để liên kết 2 chiêu các node (chứa địa chỉ(con trỏ) và dũ liệu)

chúng ta duyệt qua các phần từ để đến vị trí cần tìm rồi tiến hành đọc dữ liệu. Nếu thêm phần tử dữ liệu ta gỡ liên kết với node sau và nối liên kết với node mới rồi nối node mới với node phía sau đó.

Nếu xóa ta gỡ liên kết với node cần xóa rồi gắn liên kết với node sau đó

kt: LinkedList()

**19. Queue và các lớp cài đặt:**

a. Queue trong Java có ý nghĩa gì? Đặc điểm và ứng dụng của Queue là gì?

là một ngăn xếp với đặc điểm nguyên tắc FIFO vào trước ra trước.

ứng dụng như khi xếp hành các bệnh nhân thêm bệnh nhân nào trước thì xử lý trước

b. Các lớp cài đặt của Queue trong Java bao gồm những gì? Hãy mô tả sự khác nhau giữa chúng.

LinkedList, PriorityQueue, ArrayQueue

+LinkedList triển khai trên List và Queue là lựa chọn tự nhiên cho queue thông thương

+ PriorityQueue sắp xêp các phần tử theo 1 tiêu chí ưu tiên, phần từ ưu tiên cao lấy ra trước

+ ArrayQueue là 1 biến thể của queue nó hỗ trờ ra từ cả đầu và đuôi

**20. Set và các lớp cài đặt:**

a. Set trong Java là gì? Đặc điểm và ứng dụng của Set là gì?

Set là tập hơp không chưa các phần tử trung lập

DĐ: phần tử trong set là duy nhất và không có thứ tự cụ thể

ƯD:

+loại bỏ các phần tử trùng lặp

+ kiểm tra sự tồn tại của ptu

+ phân loại các ptu duy nhất

+quản lý quyền truy cập

b. Các lớp cài đặt của Set trong Java bao gồm những gì? Hãy mô tả sự khác nhau giữa chúng.

HashSet, LinkedListSet, TreeSet

- HashSet dùng để băn dữ liệu

- LinkedListSet giữ thứ tự chèn các phần tử

Tree Set: duy trì thứ tự tăng dần

**21. Map:**

a. Map trong Java là gì? Đặc điểm và ứng dụng của Map là gì?

là tập hợp các cặp key- value

DD: giá trị các ptu có thể giông nhau

-quản lý danh bạ

-kiếm soat quyền truy cập

-quản lý cấu hình

b. Các lớp triển khai Map?

HashMap, LinkedListMap, TreeMap

c. Những hiểu biết của bạn về HashMap? Nó khởi tạo, lưu trữ, thêm, sửa, xoá, ... các phần tử như thế nào? Lưu ý nào khi dùng HashMap và nên dùng trong những trường hợp nào?

Khơi tạo Hask<K, V>map = new HashMap<>();

Lưu trữ: sd bảng băm để lưu trữ cặp key-value

Thêm ptu: put(key, value)

sủa: put(key, value)

xóa: remove(key)

nên dùng khi

+ cần thao tác truy cập nhanh

+ không cần duy trì thứ tự

+ Lưu dữ liệu lớn

**22. Collections:**

a. Collections là gì trong Java? Chức năng và sử dụng của Collections là gì?

là lớp tiện ích trong java cung cấp các phưogn thức static để thao tác với các đối tương của Java collection framework

CN: cup cấp các phưong thức sắp xếp đảo ngược hoặc tạo ban sao không thay đổi của các collection