1. **Lập trình hướng đối tượng là gì:**

* Lập trình hướng đối tượng là chia bài toán thành các đối tượng, mỗi đối tượng gồm thuộc tính và phương thức. để xử lý bài toán ta đi xây dựng mối quan hệ giữa các đối tượng và cho các đối tượng thực thi phương thức của chính nó.

1. **Các mối quan hệ của lập trình hướng đối tượng là gì?**

* Kế thừa: kế thừa là đối tượng con thừa hưởng phương thức và thuộc tính của đối tượng cha.
* phụ thuộc: 1 đối tượng được khai báo là 1 thuộc tính của đối tượng khác thì đó là mqh đó là phụ thuộc
* song song: 2 đối tượng k liên quan gì tới nhau

1. **Cách xác định đối tượng, thuộc tính, phương thức xong bài toán**

* Đối tượng là danh từ k có miền giá trị
* thuộc tính là danh từ có miền xác định
* phương thức là động từ mô tả hành động của đối tượng.

1. **Các tính chất của lập trình hướng đối tượng là gì. Giải thích từng tính chất**:

Có 4 tính chất:

* Đóng gói: Tạo ra các gói để chứa các class có tính chất giống nhau, có 4 access modify để bảo vệ
* Kế thừa: Class con kế thừa thuộc tính và phương thức của class cha mà class cha cho phép
* Đa hình: 1 class có nhiều phương thức cùng tên nhưng khác số lượng tham số hoặc kiểu tham số. hoặc class con có thể có phương thức giống với phương thức ở class cha. Thể hiện trong java bằng overload và override.

1. đối tượng có thể được khởi tạo bằng nhiều thể hiện khác, ví dụ

Tugiac tg=new TuGiac();

TuGiac tg2=new Hinhvuong();

TuGiac tg3=new HinhChuNhat();

* Trừu tượng: Trừu tượng hóa đối tượng, Tập trung vào những thông tin cần thiết của bài toán

Tính trừu tượng giúp bạn tập trung vào những cốt lõi cần thiết của đối tượng thay vì quan tâm đến cách nó thực hiện.

Tính trừu tượng được thể hiện qua abstract và interface trong java

1. **Overloading and overriding là gì**

* Overloading(Nạp chồng) là việc 1 class có nhiều phương thức nhưng khác tham số hoặc khác kiểu dữ liệu của tham số, nó thể hiện tính đa hình lúc biên dịch.
* Overriding(Ghi đè) là việc cài đặt lại các phương thức đã có ở lớp cha, các phương thức override ở lớp con phải giống hệt lớp cha, nó thể hiện tính đa hình lúc runtime

1. **Abstract class là gì?**

Abstract class là 1 lớp trừu tượng trong java, nó có thể chứa phương thức trừu tượng hoặc không trừu tượng.

Abstract class dùng để miêu tả đối tượng có chung thuộc tính và phương thức. nhưng có 1 vài phương thức có cụ thể hóa khác nhau. Nên ta cần có abstract method.(ví dụ ngựa và chim có chung bản chất là di chuyển nhưng cách di chuyển khác nhau)

1. **Interface là gì?**

Interface là 1 bộ hành vi chung của các đối tượng, Nó bao gồm các method trừu tượng.

1. **Abstract và interface là gì và khác nhau ntn?**

|  |  |
| --- | --- |
| Abstract class có phương thức **abstract**(không có thân hàm) và phương thức **non-abstract** (có thân hàm), constructor. | Interface chỉ có phương thức **abstract**. Từ java 8, nó có thêm **các phương thức default và static** |
| Abstract class **không hỗ trợ đa kế thừa**.( Dùng từ khóa extend và chỉ đc extend 1 class) | Interface **có hỗ trợ đa kế thừa** (Dùng từ khóa implement để thực thi, có thể implememnt nhiều interface) |
| Abstract class có các biến **final, non-final, static and non-static**. | Interface chỉ có các biến **static và final**. |
| Access modifier có thể là private, default, protected, public | Access modifier Chỉ có thể là public |
| Abstract class **có thể cung cấp nội dung cài đặt cho phương thức của interface**. | Interface **không thể cung cấp nội dung cài đặt cho phương thức của abstract class**. |
| Sử dụng từ khóa abstract để khai báo | Sử dụng từ khóa implement. |

1. **Có mấy loại java. Tại sao java lại viết 1 lần chạy mọi nơi.**

* Có 3 loại java, JavaSE, JavaEE, JavaME.
* Java viết 1 lần chạy mọi nơi vì nó có 1 máy ảo JVM và code được biên dịch ra bytecode thay vì mã máy…

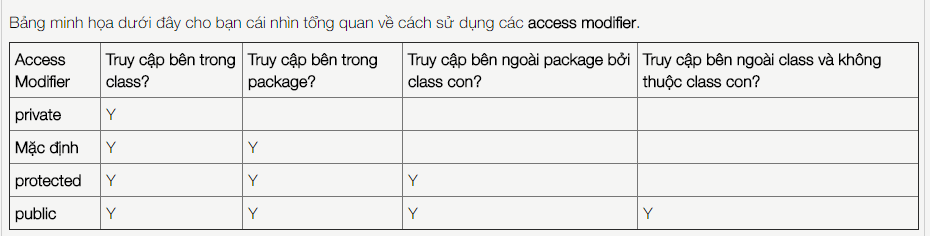
1. **Kiểu dữ liệu là gì và có những loại dữ liệu nào trong java? Có mấy kiểu dữ liệu cơ bản, là những loại nào?**

* Kiểu dữ liệu là một sự quy định về cấu trúc và miền giá trị dữ liệu và tập hợp các phép toán tác động lên miền giá trị đó.
* Có 2 loại kiểu dữ liệu: Kiểu dữ liệu nguyên thủy và kiểu dữ liệu có cấu trúc.
* Kiểu dữ liệu cơ bản có 8 loại: boolean, byte, short, int, long, float, double, char

1. **Access modify là gì. Có những loại nào và tính chất của từng loại.**

* Các access modifiers trong java xác định độ truy cập (Phạm vi) vào dữ liệu của của các trường, phương thức, class.

Có 4 access modifier:



1. **Có những loại cấu trúc rẽ nhánh nào. Và khi nào thì nên dùng loại nào?**

* Có 2 loại là If/else và switch/case
* If else dùng khi có nhỏ hơn 4 nhánh, còn lại dùng switch case.

1. **Có những loại vòng lặp nào. Khi nào dùng loại nào?**

* For, While, Do/while
* For dùng khi biết trước số lần lặp
* While dùng khi chưa biết số lần lặp
* Do/while dùng khi chưa biết số lần lặp và cần chạy ít nhất 1 lần

1. **Break và continue dùng như thế nào?**

* Break dùng để dừng vòng lặp.
* Continue dùng để bỏ qua lần lặp hiện tại và sang lần lặp tiếp theo

1. J**ava có đa kế thừa không? Dùng đa kế thừa trong java NTN?**

* Java không có đa kế thừa mà java hỗ trợ interface để thực hiện đa kế thừa

1. **Có mấy vùng nhớ trong java. Đặc điểm của từng vùng**.

* Có 4 vùng nhớ. Code, data, stack, head.
* **Vùng nhớ code (code segment)**, theo như tên gọi của nó, tất cả mã chương trình (machine code) được lưu ở đây khi chương trình được thực thi.
* **Vùng nhớ data (data segment)**, đây là nơi lưu trữ những dữ liệu chung của chương trình như các biến static, constant,… những biến dữ liệu mà được sinh ra khi chương trình bắt đầu thực thi và chỉ được giải phóng khi chương trình kết thúc.

Ví dụ: private static String TAG=“Main”;

private final int NUMBER=1

* **Vùng nhớ stack (stack segment)**, đây sẽ là nơi lưu trữ các biến thuộc nhóm **kiểu dữ liệu cơ sở** (primitive data type như là boolean, int, char,…) và địa chỉ của ô nhớ (memory address).
* **Vùng nhớ heap (heap segment)**, đây là nơi lưu trữ tất cả các **đối tượng** (object) được sinh ra **trong thời gian thực thi chương trình** (run time).

**Chú ý:**

* Với vùng nhớ **Code và Data**, khi chương trình thực thi sẽ **cấp phát** một không gian có **kích thước không đổi**.
* Còn đối với vùng nhớ **Stack và Heap**, kích thước của nó sẽ **thay đổi** (hoặc mở rộng khi tạo thêm biến hoặc đối tượng, hoặc thu hẹp khi một biến hoặc đối tượng được giải phóng) và nếu cần mở rộng thêm, nó sẽ **lấy không gian từ Free memory.**
* Ngoài ra, trong bộ nhớ máy tính, **mỗi ô nhớ** sẽ **được đánh địa chỉ** để xác định chính xác vị trí của nó trong bộ nhớ. Vì vậy trong mỗi ô nhớ sẽ bao gồm 2 thành phần là **Value** và **Address**

.

1. **String là gì, nó có tính chất gì và dùng String để làm gì?**

* String là một đối tượng trong java và nó chứa các kí tự.
* String là đối tượng Immutable(ví dụ khi nối chuỗi nó sẽ giữ nguyên chuỗi cũ và tạo chuỗi mới) và dùng để xử lý công việc liên quan đến chuỗi…
* Nó có tính nguyên thủy và tính đối tượng

1. **Các cách khai báo String và khác nhau giữa các cách khai báo?**

* String a = new String(“Hello”);
* String a = “Hello”;
* Khác nhau giữa 2 cách là cách thức cấp phát bộ nhớ và nơi lưu trữ.

Khi khai báo các a thì luôn tạo ra đối tượng mới và lưu trữ tại head,

Khi khai báo cách b thì sẽ kiểm tra ở string pool trong stack nếu tồn tại thì k tạo mới mà chỉ gán con trỏ tới vùng nhớ đó, nếu chưa có thì tạo mới trong stack và trỏ con trỏ vô đó

1. **Để so sánh String ta dùng gì, tại sao lại dùng nó?**

Để so sánh String ta dùng equals vì nếu dùng == nó sẽ so sánh địa chỉ ô nhớ… có thể dẫn tới khác nhau.

1. **Kể tên 1 số phương thức của String.**

CompareTo, Equal, substring. Indexof…vv

1. **StringBuffer và StringBuilder**

StringBuffer và StringBuilder đều là đối tượng mutable và dùng để xử lý chuỗi. Khi dùng StringBuffer và StringBuilder thì nó sẽ k tạo ra đối tượng mới khi thao tác chuỗi

* nó có phương thức append và insert.

1. **Khác nhau giữa StringBuffer, StringBuilder, String, String pool.**

* **StringBuilder(không đồng bộ)**và **StringBuffer(đồng bộ)**là giống nhau, nó chỉ khác biệt tình huống sử dụng có liên quan tới đa luồng (Multi Thread).Nếu xử lý văn bản sử dụng nhiều luồng (Thread) bạn nên sử dụng **StringBuffer**để tránh tranh chấp giữa các luồng.

Nếu xử lý văn bản sử dụng 1 luồng (Thread) nên sử dụng **StringBuilder**.

Nếu so sánh về tốc độ xử lý **StringBuilder**là tốt nhất, sau đó **StringBuffer**và cuối cùng mới là **String**.

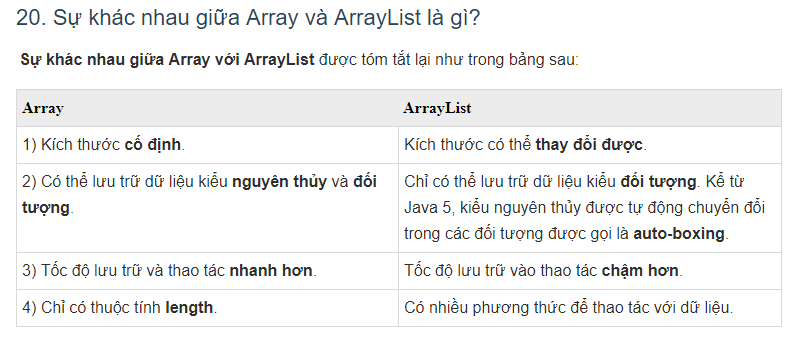
* String khác vs StringBuilder,Buffer là immutable và mutable
* Stringpool là 1 vùng nhớ lưu các chuỗi ở bộ nhớ stack

1. **Collections(List, Set,Map, QUEUE)**

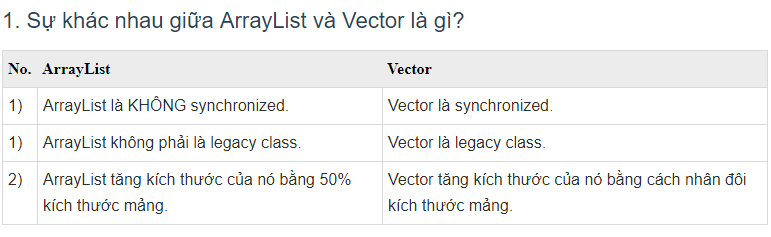
* **Set:**là một collection không thể chứa 2 giá trị trùng lặp. Set được sử dụng để biểu diễn các bộ, chẳng hạn như bộ tú lu khơ, thời khóa biểu của học sinh, các tiến trình đang chạy trên máy tính…(lưu trữ danh sách duy nhất, Nếu trùng nó sẽ lấy giá trị thằng đứng trước trong mảng)
* **List:**là một collection có thứ tự (đôi khi còn được gọi là một chuỗi). List có thể chứa các phần tử trùng lặp. Thường có quyền kiểm soát chính xác vị trí các phần tử được chèn vào và có thể truy cập chúng bằng chỉ số (vị trí của chúng).(lưu trữ danh sách theo index)
* **Queue (hàng đợi):**là một collection được sử dụng để chứa nhiều phần tử trước khi xử lý. Bên cạnh các thao tác cơ bản của collection, Queue cung cấp các thao tác bổ sung như chèn, lấy ra và kiểm tra. Queue có thể được sử dụng như là FIFO (first-in, first-out – vào trước, ra trước)
* **Map:**là một đối tượng ánh xạ mỗi key tương úng với một giá trị. Map không thể chứa giá trị trùng lặp. Mỗi key có thể ánh xạ đến nhiều nhất một giá trị.(lưu trữ theo key value)



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



1. **ArrayList và Vector là gì?**

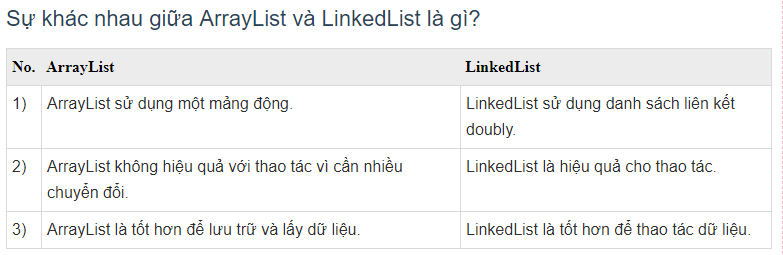


Arraylist: bất đồng bộ còn vector là đồng bộ

1. **Hashmap và Hashtable**

* hashmap cho phép truyền 1 key null và nhiều value null, hashtable không cho phép null, hashmap là bất đồng bộ,hashtable thì đồng bộ

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



Chúng hiệu quả bởi vì khi them sửa xóa ở 1 phần tử. linkedlist chỉ cần đổi 1 vị trí. Còn arraylist sẽ phải di chuyển hết các phần tử

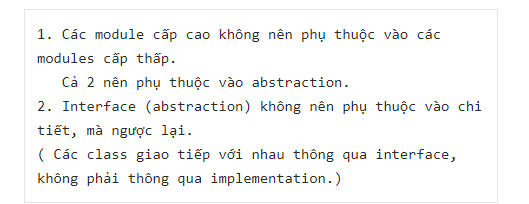
1. **List và Set là gì?**

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

1. **Solid**

Nguyên lý để code dễ đọc,dễ test,rõ ràng hơn, đặc biệt **maintainace code dễ hơn nhiều**

* Single Responsibility principle: 1 class chỉ nên giữ 1 nhiệm vụ duy nhất(ví dụ class thêm sửa sv thì ko in ra sinh viên)
* Open/closed principle: có thể mở rộng class nhưng không được sửa class. Khi thêm chức năng viết thêm class mới kế thừa class cũ
* Liskov Substitution Principle: Trong 1 chương trình các object của class con có thể thay thế class cha mà không làm thay đổi tính đúng đắn của chương trình
* Interface Segregation Principle: Thay vì dùng 1 interface lớn ta chia nhỏ thành nhiều interface
* Dependency inversion principle:



https://toidicodedao.com/2015/03/24/solid-la-gi-ap-dung-cac-nguyen-ly-solid-de-tro-thanh-lap-trinh-vien-code-cung/

1. **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. **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.

Ví dụ, Nhanvien là một lớp có 3 thành viên dữ liệu: id, ten và luong. Nhưng, chúng ta muốn kiểm tra sự giống nhau của đối tượng nhân viên trên cơ sở tiền lương. Khi đó, chúng ta cần ghi đè bằng phương thức equals().

<http://viettuts.vn/interview/list-cau-hoi-phong-van-java-collection>

1. **Có những loại lỗi nào trong java. Giải thích từng loại?**

Có 3 loại lỗi, lỗi biên dịch, lỗi logic, lỗi runtime.

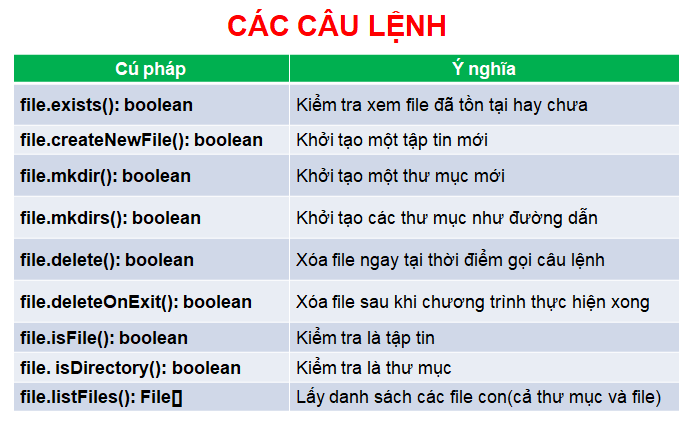
checked exception(những lỗi có thể check được khi biên dịch) và unchecked exception(Những lỗi không thể check được khi biên dịch): <http://viettuts.vn/interview/list-cau-hoi-phong-van-java-core-4> <http://viettuts.vn/exception-handling>

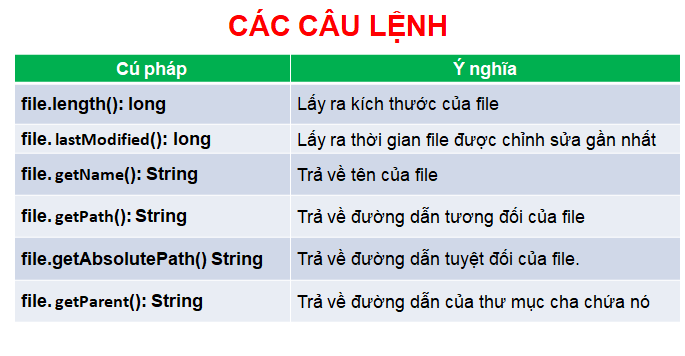
1. **Try catch là gì.**

Khi chương trình xảy ra 1 lỗi bất thường gì đó làm dừng chương trình. Try catch sẽ bắt những lỗi đó làm cho chương trình không bị dừng đột ngột.

1. **File là gì, Đối tượng file trong java là gì, các phương tức hay dùng trong File?**

* File là một tập tin có dữ liệu được lưu trữ trong máy tính.
* Định nghĩa: Đối tượng File được Java cung cấp dùng để thao tác và truy xuất các thông tin về một tập tin hoặc thư mục có trong hệ thống.
* Nằm trong gói thư viện **java.io**





1. **Các luồng đọc ghi file là gì, kể tên 1 số loại đọc ghi file? khác nhau giữa Reader/ Stream?**

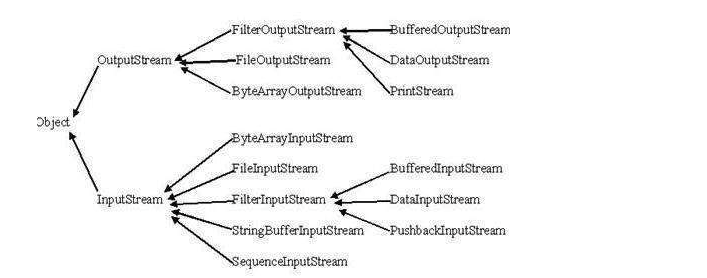
Java cung cấp các đối tượng dung để đọc ghi file. Có 2 loại đọc ghi file là đọc byte(Stream) hoặc đọc văn bản(text)

InputStream/FileInputStream, OutputStream/FileOutputStream

FileReader/BufferReader FileWriter/BufferWriter/

RandomAccessFile

ByteArrayInput/OutputStream..vv



1. **Khác nhau giữa FileInputStream/InputStream, FileOutputStream/OutputStream, khi nào dùng nó.**

InputStream/OutputStream dung để thao tác với file trên mạng.

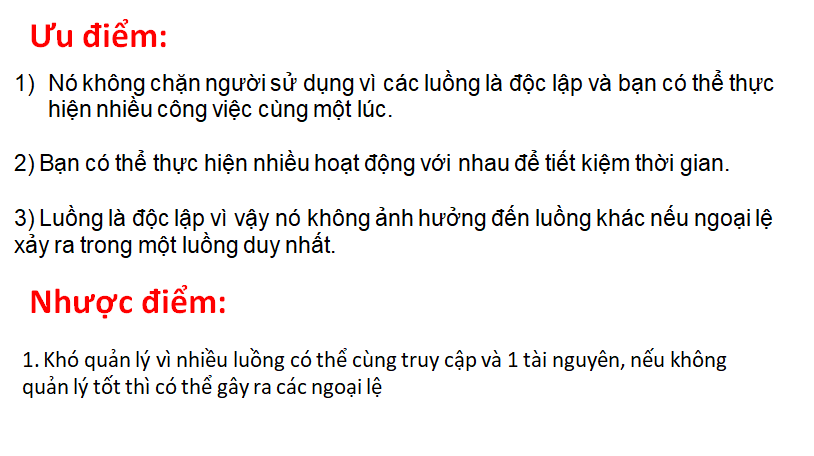
FileInputStream/FileOutputstream thường thao tác với file trên ổ cứng

1. **Khái niệm đa luồng trong java, tại sao lại cần sử dụng đa luồng**.

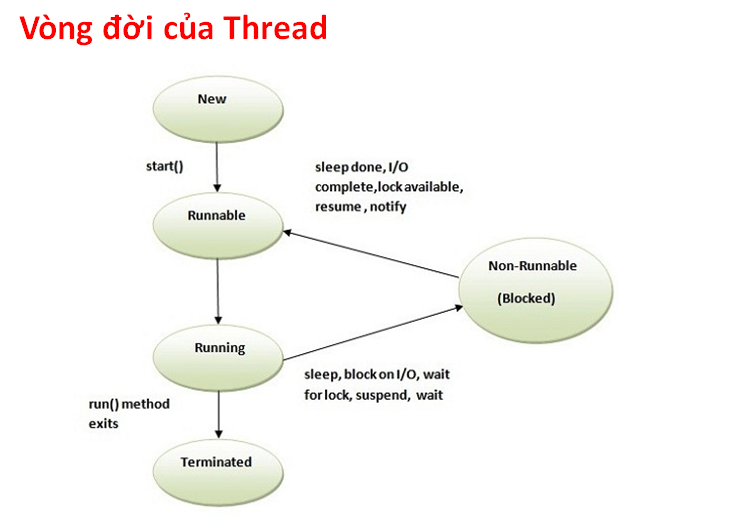
**Đa luồng (multithreading) trong java** là một tiến trình thực hiện nhiều luồng đồng thời.  
Luồng (thread) về cơ bản là một tiến trình con (sub-process). Nó là đơn vị nhỏ nhất của tiến trình và đa luồng (multithreading) được sử dụng để tạo ra hệ thống đa nhiệm (multitasking).

Trong những chương trình có UI khi ta thực hiện các tác vụ nặng. nếu k sử dụng thread thì sẽ bị block UI và làm cho ứng dụng bị lỗi ANR

1. **Ưu nhược điểm của đa luồng**.



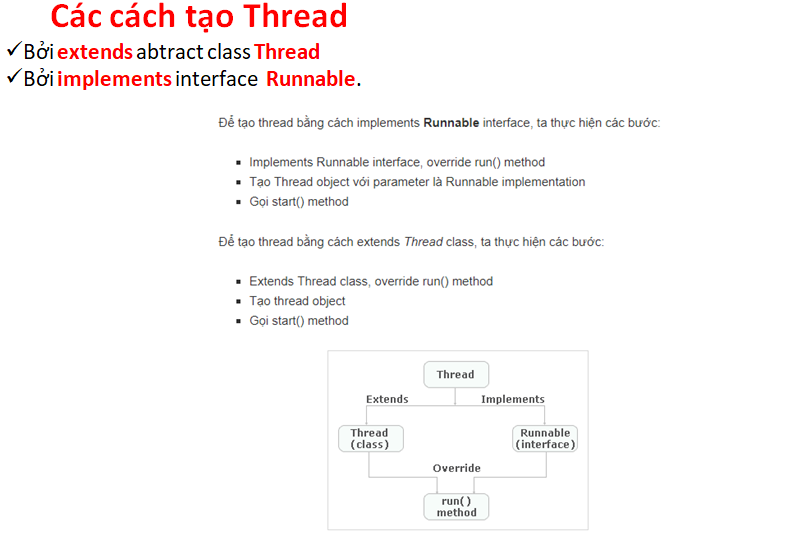
1. **Vòng đời của Thread.**

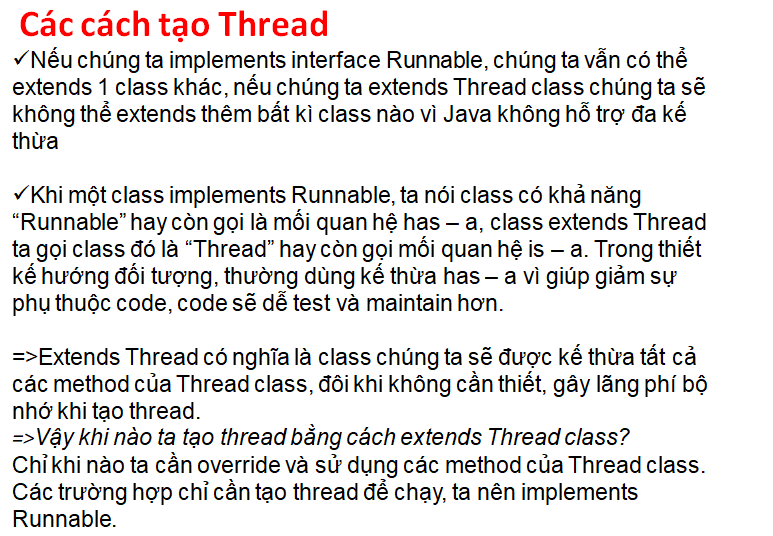




1. **Có mấy cách tạo thread. Khi nào thì dùng cách nào?**

* Có 2 cách tạo thread.





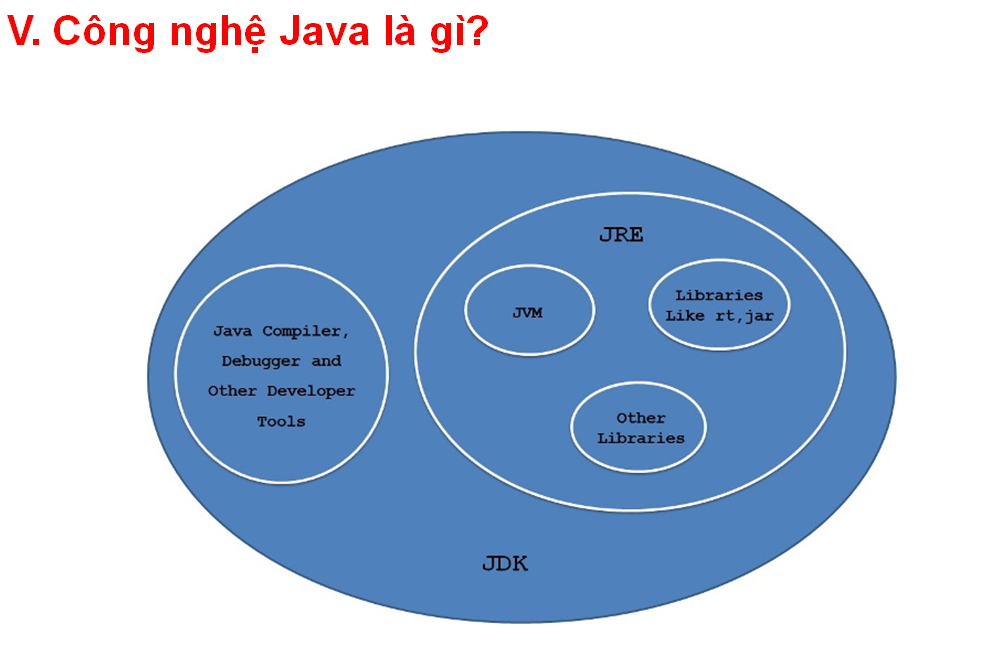
1. **Thread?**





1. **JDK, JRE, JVM**

* JVM: java virtual machine: quản lý hệ thống và cung cấp môi trường thực thi cho các ứng dụng di động, cho phép java chạy trên mọi thiết bị và tối ưu bộ nhớ
* JRE: Java runtime environment được thiết kế để chạy các phần mềm khác chứa các thư viện, trình tải và máy ảo java.Runtime environment tải các tệp lớp và đảm bảo có quyền truy cập vào bộ nhớ của hệ thống
* JDK: java development kit là nền tảng chính để xây dựng các ứng dụng, dùng để biên dịch java
* Java Development Kit (JDK) là một trong ba gói công nghệ cốt lõi được sử dụng trong lập trình Java, cùng với JVM (Java Virtual Machine) và JRE (Java Runtime Environment)
* JRE có thể được sử dụng như một thành phần độc lập để chạy các chương trình Java, nhưng nó cũng là một phần của JDK. JDK yêu cầu JRE vì chạy các chương trình Java là một phần của việc phát triển chúng.
* Định nghĩa kỹ thuật: JDK là một triển khai của đặc tả nền tảng Java, bao gồm các trình biên dịch và thư viện lớp.
* Định nghĩa hàng ngày: JDK là gói phần mềm bạn tải xuống để tạo các ứng dụng dựa trên Java.



**Tìm hiểu thêm**

**Nguyên lý SOLID:**

<https://toidicodedao.com/2015/03/24/solid-la-gi-ap-dung-cac-nguyen-ly-solid-de-tro-thanh-lap-trinh-vien-code-cung/>

<http://butchiso.com/2016/07/solid-5-nguyen-tac-cua-thiet-ke-huong-doi-tuong.html>

**Các thuật toán:**

**tìm kiếm:** Vét cạn, Backtracking, BFS, DFS(Sử dụng QUEUE and STACK tự implement), Binary search

**sắp xếp:** Bubble Sort, Selection Sort, *insertion sort*, Counting Sort, Merge sort, *heapsort*,Quick sort,Binary search

More detail.

<http://viettuts.vn/interview/list-cau-hoi-phong-van-java-core>

<https://vietjack.com/cau_hoi_phong_van_java/index.jsp>

<https://stackjava.com/uncategorized/phong-van-vi-tri-lap-trinh-vien-java.html>

<http://toando.coffee/115-cau-phong-van-java/>

<http://www.nghean-aptech.com/vieclam/2016/10-cau-hoi-kinh-dien-trong-buoi-phong-van.aspx>

1. **Supper là gì, This là gì. Dùng this và supper trong trường hợp nào, Khác nhau giữa this và supper?**

* Super là toán tử đại diện cho class cha,
* This là toán tử đại diện cho class gần nhất chứa nó.