### ****1. Lập trình hướng đối tượng là gì?(****Object Oriented Programming****)****

Trả lời: 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

vì sao chọn java:

Một trong những lý do hàng đầu mà em nên học ngôn ngữ lập trình Java đó là vì ứng dụng của java khá là rộng, là **ngôn ngữ lập trình bậc cao, hướng đối tượng và giúp bảo mật mạnh mẽ,**

### Ví dụ ứng dụng của java: **Java làm phần mềm phát triển**, **Java làm ứng dụng web**

### **Lớp và đối tượng:**

Đối tượng trong OOP bao gồm 2 thành phần chính:

* Thuộc tính (Attribute): là những thông tin, đặc điểm của đối tượng
* Phương thức (Method): là những hành vi mà đối tượng có thể thực hiện

**Đối tượng(Object) bao gồm hai thành phần chính là thuộc tính và phương thức, là một thể hiện cụ thể đặc điểm hành vi của một lớp(Class),** đối tượng là các thể hiện (kết quả) của một lớp.

**Một lớp** là khuôn mẫu chứa một nhóm đối tượng có các thuộc tính chung. Nó là một mẫu hoặc thiết kế từ đó các đối tượng được tạo ra. Lớp cũng bao gồm thuộc tính và phương thức.

* Thuộc tính (Attribute): là những thông tin, đặc điểm của đối tượng
* Phương thức (Method): là những hành vi mà đối tượng có thể thực hiện

**4 đặc điểm của java:**

## 1. Tính chất trừu tượng (abstract)

Là từ các mô tả của chương trình tìm ra các đặc trưng, hành động để trừu tượng hóa thành các đối tượng các class.

Ví dụ viết 1 chương trình nhập thông tin sinh viên gồm họ tên, tuổi, ngày sinh, lớp.  
Từ yêu cầu bài toán ta rút ra được các danh từ: “sinh viên”, “họ tên”, “tuổi”, “ngày sinh”, “lớp” –> “sinh viên” là 1 đối tượng, “họ tên”, “tuổi”, “ngày sinh”, “lớp” là các thuộc tính, đặc trưng của “sinh viên” do đó ta trừu tượng hóa thành class: SinhVien

## 2. Tính chất đóng gói (encapsulation)

Đóng gói ở đây là đóng gói các biến, method thành các class; đóng gói các class thành 1package…  
Việc đóng gói giúp che giấu thông tin, đảm bảo sự toàn vẹn dữ liệu.

Ví dụ: tính đóng gói mà ta hay dùng nhất đó là qua phạm vi truy cập (access modifier): public, private, protected, default

* **Private**: Chỉ có thể truy cập trong cùng class.
* **Default**: Có thể truy cập trong cùng class và cùng package.
* **Protected**: Có thể truy cập trong cùng class, package và ngoài package bởi lớp con.
* **Public**: Có thể truy cập ở bất cứ đâu.

## 3. Tính chất kế thừa (inheritance)

Tính chất này thì đơn giản hơn, đó là sử dụng lại các thuộc tính, method sẵn có từ các class khác mà không phải xây dựng từ đầu

## 4. Tính chất đa hình (polymorphism)

Tính đa hình trong lập trình OOP cho phép các đối tượng khác nhau thực thi chức năng giống nhau theo những cách khác nhau.

4.1. Phân biệt override và overload?

* Overriding (ghi đè) là viết lại, định nghĩa lại method mà nó thừa kế từ lớp cha
* Overloading (nạp chồng): là sử dụng các method có tên giống nhau nhưng tham số đầu vào khác nhau:

**Sự khác nhau giữa abstract class và interface :**

**Abstract(trừu tượng)** nghĩa là một cái gì đó không hoàn toàn cụ thể, nó chỉ là một ý tưởng hoặc ý chính của một cái gì đó mà không có bản triển khai cụ thể. Vì vậy Class abstract chỉ là một cấu trúc hoặc hướng dẫn được tạo cho các class cụ thể khác.

**Interface (Giao diện)** là một hình thức, giống như một hợp đồng, nó không thể tự làm bất cứ điều gì. Nhưng khi có một class ký kết hợp đồng (implement Interface) này, thì class đó phải tuân theo hợp đồng này.

1. Implementation: Class abstract có thể implement các Interface. Trong khi Interface thì không thể implement class abstract.
2. Inheritance: Class abstract có thể kế thừa được một class khác. Trong khi Interface có thể kế thừa được nhiều Interface khác.
3. Accessibility: các thành viên trong Interface kiếu mặc định là public. Trong khi class abstract thì lại có thể là private, protected,..
4. Diagram

   Description automatically generated with low confidence

**Khi nào nên dùng?**

1. Tạo một class abstract khi bạn đang cung cấp các hướng dẫn cho một class cụ thể.
2. Tạo Interface khi chúng ta cung cấp các hành vi bổ sung cho class cụ thể và những hành vì này không bắt buộc đối với clas đó.

Exception:

Trong Java, Exception là một sự kiện mà phá vỡ luồng chuẩn của chương trình. Nó là một đối tượng mà được ném tại Runtime. Một exception (ngoại lệ) trong Java là một vấn đề xảy ra trong quá trình thực hiện của chương trình. Một ngoại lệ có thể xảy ra với nhiều lý do khác nhau, như dưới đây:

Người dùng nhập dữ liệu không hợp lệ.

Một file cần được mở nhưng không thể tìm thấy.

Collection:

Collections trong java là một khuôn khổ cung cấp một kiến trúc để lưu trữ và thao tác tới nhóm các đối tượng. Tất cả các hoạt động mà bạn thực hiện trên một dữ liệu như tìm kiếm, phân loại, chèn, xóa,... có thể được thực hiện bởi Java Collections.

Collection trong java là một root interface trong hệ thống cấp bậc Collection. Java Collection cung cấp nhiều interface (Set, List, Queue, Deque vv) và các lớp (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet vv).

Dưới đây là mô tả những interface chính của Collection

* **Set:**là một collection không thể chứa 2 giá trị trùng lặp
* **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).
* **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)
* **Deque:**là một collection được sử dụng để chứa nhiều phần tử trước khi xử lý. Ngoài các thao tác cơ bản của collection, một Deque cung cấp các thao tác bổ sung như chèn, lấy ra và kiểm tra. Deques có thể được sử dụng như là FIFO (first-in, first-out - vào trước, ra trước) và LIFO (last-in, first-out - vào sau, ra trước). Trong một Deque, tất cả các phần tử mới có thể được chèn vào, lấy ra và lấy ra ở cả hai đầu.
* **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.

**Generic: viết code và tái sử dụng cho nhiều kiểu khác nhau, khi muốn sử dụng với kiểu nào thì truyền kiểu tương ứng vào làm tham số của lớp của hàm của interface tương ứng**

**Sự khác nhau giữa JSP và HTML:**

**Table

Description automatically generated**

**Vector:** Vector là một trong những thành phần của collection framework implement từ List interface. Kích thước tự động co giãn khi số lượng phần tử tăng lên hoặc giảm đi. Giống như Array, các phần tử của Vector có thể truy xuất thông qua chỉ số index.

Heap và Stack:

**JDBC: chuẩn kết nối CSDL, cung cấp các interfsacce & class nhằm tạo cơ sở cho các ứng**

**dụng java tương tác với các hệ CSDL**

**Graphical user interface, text, application

Description automatically generated**

**Text

Description automatically generated with medium confidence**

**JDBC API: Là các interface, để gọi các hàm để thực thi câu lệnh**

**JDBC driver manager: có nghĩa là tương ứng với những cơ sở dữ liệu khác nhau thì những driver cho quản trị cơ sở dữ liệu đó là khác nhau**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Graphical user interface, text, email

Description automatically generated**

**Static và non-static:**

Nhận xét: vậy những thuộc tính và hàm có static thì có thể sử dụng trực tiếp mà không thông qua đối tượng.

Đối với những hàm **không có static thì chỉ có thể truy cập thông qua đối tượng.**

**Thuật toán sắp xếp :**

Merge sort là một thuật toán chia để trị. Thuật toán này chia mảng cần sắp xếp thành 2 nửa. Tiếp tục lặp lại việc này ở các nửa mảng đã chia. Sau cùng gộp các nửa đó thành mảng đã sắp xếp.

**Insert sort:** Thuật toán sắp xếp chèn thực hiện sắp xếp dãy số theo cách duyệt từng phần tử và chèn từng phần tử đó vào đúng vị trí trong mảng con(dãy số từ đầu đến phần tử phía trước nó) đã sắp xếp sao cho dãy số trong mảng sắp đã xếp đó vẫn đảm bảo tính chất của một dãy số tăng dần.

Bubble sort: Sắp xếp nổi bọt (Bubble Sort) Giải thuật sắp xếp này được tiến hành dựa trên việc so sánh cặp phần tử liền kề nhau và tráo đổi thứ tự nếu chúng không theo thứ tự.

Selection sort: Chọn phần tử nhỏ nhất đưa về vị trí đầu tiên của mảng hiện tại và không cần quan tâm đến nó nữa. Khi đó mảng chỉ còn lại n - 1 phần tử của mảng ban đầu, tiếp tục xét từ phần tử thứ 2 của mảng.

**Static:**

**Text

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

## **API là gì?**

**API** là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của **Application Programming Interface** – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

**Câu hỏi để hỏi nhà phỏng vấn:**

### 1, Anh/chị nghĩ những phẩm chất quan trọng nhất mà một ứng viên cần để thành công trong vai trò này là gì?

### 2, Em muốn gắn bó với công ty lâu dài, liệu anh có thể tư vấn và hoạch định lộ trình phát triển nghề nghiệp lâu dài với công ty hay công

