1. OOP : Lập trình hướng đối tượng- 4 tính chất:

1. Tính đóng gói: thể hiện qua access modifier

-Public: có thể truy cập từ bất cứ đâu

-Protected: các phương thức và thuộc tính đc phép truy xuất trong cung 1 lớp và các lớp con, có thể khác package

-Defauft (k có access modifier): lớp và các thành phần của lớp đc truy xuất ở những nơi trong cùng 1 package,k thể truy xuất ở lớp con

-Private: các phương thức và thuộc tính chỉ đc truy xuất trong cùng 1 lớp

-Static: đc sử dụng để khai báo các thuộc tính và phương thức của lớp, có thể truy xuất các thành phần static bằng các sử dụng lớp hoặc đối tượng, k cần khởi tạo đối tượng vẫn có thể sử dụng các thành phần static: giảm sử dụng bộ nhớ.

2. Tính đa hình: thể hiện qua overload, overriding

-Override: xuất hiện khi trong class con tồn tại một method được định nghĩa với cùng tên và cùng số lượng tham số với method của class cha, có cùng kiểu trả về.

-Overload: Overloading xuất hiện khi trong một class tồn tại một method được định nghĩa với cùng tên nhưng khác số lượng tham số đầu vào với method của class cha

3. Tính trừu tượng:thể hiện qua abstract class và interfac

Abstract class:

-Abstract class có thể có các thuộc tính và phương thức bình thường

-không thể khởi tạo đối tượng của lớp abstract

• Lớp có phương thức abstract thì bắt buộc phải abstract

-1 lớp k phải abstract kê thừa từ 1 lớp cha abstract thì bắt buộc phải triển khai các phương thức abstract của lớp cha

-lớp/phương thức abstract thì k thể là final

- Lớp kế thừa lớp abstract thì phải triển khai toàn bộ các phương thức  
abstract

Interface:

-interface chỉ chứa các hằng số và abstract method

- Từ khoá implement được sử dụng để triển khai interface

-interface quy định các hành vi chung cho các lớp triển khai nó

- Interface có thể kế thừa interface khác

\* Lựa chọn Interface hay Abstract class• Trong nhiều trường hợp, có thể chuyển đổi giữa việc sử dụng  
Interface và Abstract class  
• Nếu có sự gần gũi, rõ ràng giữa các lớp về mối quan hệ cha-con thì  
nên sử dụng lớp (mối quan hệ is-a)  
• Ví dụ: Apple is a Fruit (Táo là một Quả)• Nếu không có mối quan hệ gần gũi thì nên chọn interface (mối quan  
hệ can-do)  
• Ví dụ: Bird can fly (Chim có thể bay)

**SO SÁNH GIỮA COMPARABLE VÀ COMPARATOR**

-Giống nhau: đều là interface đc sử dụng để sắp xếp các phần tử của collection

-Khác nhau:

\* Comparable:

+ bạn phải implement interface comparable cho lớp đối tượng cần so sánh

+ cung cấp phương thức compareTo() để sắp xếp các phần tử

+ sắp xếp các phần tử bởi phương thức collection.sort(list)

\*Comparator:

+ bạn không phải implement interface comparator cho lớp đối tượng cần so sánh

+ cung cấp phương thức compare() để sắp xếp các phần tử

+ sắp xếp các phần tử bởi phương thức collection.sort(list,Comparator)

4. Tính kế thừa: extends

Kế thừa là cơ chế cho phép một lớp thừa hưởng các đặc điểm và hành vi của  
một lớp khác  
• Lớp được kế thừa gọi là lớp cha, lớp kế thừa gọi là lớp con  
• Ghi đè phương thức là hình thức lớp con viết lại các phương thức đã có của lớp  
cha  
• Sử dụng mũi tên rỗng để biểu diễn mối quan hệ kế thừa giữa các lớp  
• Java không hỗ trợ đa kế thừa  
• Từ khoá final được sử dụng để ngăn chặn việc kế thừa từ một lớp và việc ghi đè  
phương thức  
• Đa hình là cơ chế cho phép một biến kiểu cha có thể trỏ đến các đối tượng kiểu  
con  
• Lớp Object là lớp cha của tất cả các lớp trong Java  
• Phương thức toString() được sử dụng để trả về một chuỗi mô tả đối tượng  
• Ép kiểu là hình thức chuyển đổi tham chiếu đối tượng từ một kiểu này sang  
tham chiếu đối tượng thuộc kiểu khác

1. **CÁC KIỂU DỮ LIỆU TRONG JAVA.SỰ KHÁC NHAU CỦA CÁC KIỂU ĐÓ**

Java hỗ trợ nhiều kiểu dữ liệu khác nhau

• Các câu lệnh điều khiển giúp điều hướng luồng thực thi của ứng  
dụng  
• Các vòng lặp được Java hỗ trợ: for, for-each, while  
• Các khái niệm của mảng: Tên mảng, kiểu dữ liệu, kích thước, phần  
tử, chỉ số  
• Tên của mảng tuân theo quy tắc của tên biến  
• Chỉ số của phần tử đầu tiên là 0  
• Chỉ số của phần tử cuối cùng là length – 1  
• Có thể sử dụng vòng lặp for và for-each để duyệt mảng

Có 2 kiểu dữ liệu:

-Kiểu dữ liệu nguyên thủy: byte, short, int, long, char, float, double, boolean

giá trị của kiểu dữ liệu nguyên thủy nằm trong bộ nhớ stack

-Kiểu dữ liệu tham chiếu: biến mảng là 1 kiểu tham chiếu, object, string

giá trị của dữ liệu tham chiếu là nó tham chiếu đến giá trị nằm trong vùng nhớ heap

**. PHÂN BIỆT THAM TRỊ VÀ THAM CHIẾU**

-Tham trị (pass by value):giá trị của biến đc lưu trên stack, khi chương trình thực thi thì sẽ tạo ra 1 giá trị mới bằng cách copy giá trị gốc và mình chỉ thao tác vs giá trị copy này.khi chúng ta thay đổi các giá trị copy này thì không ảnh hưởng đến giá trị gốc.Thường các biến nguyên thủy là theo kiểu tham trị

-Tham chiếu (pass by reference): địa chỉ của biến đc lưu trên stack nhưng nó tham chiếu đến giá trị bên vùng nhớ heap.khi thay đổi các giá trị này thì giá trị của biến cũng thay đổi theo.Thường mảng là kiểu tham chiếu

1. **Java Collection framework**

* Cung cấp đầy đủ các interface và class  
  thực hiện các thao tác với cấu trúc dữ liệu thông dụng
* Cấu trúc dữ liệu là hình thức tổ chức dữ liệu và cung cấp các phương  
  thức để thao tác với các phần tử

• Cấu trúc Set chứa các phần tử không trùng lặp:

-Có thể sử dụng các lớp:

+**hashSet**: lưu trữ các phần tử của nó trong bảng băm, tuy nhiên nó không đảm bảo về thứ tự các phần tử chèn vào

+**treeSet**: lưu trữ các phần tử của nó trong một cây, sắp xếp các phần tử của nó dựa trên các giá trị của chúng, về cơ bản là chậm hơn HashSet.

+**LinkedHashSet**: được triển khai dưới dạng bảng băm với có cấu trúc dữ liệu danh sách liên kết, sắp xếp các phần tử của nó dựa trên thứ tự chúng được chèn vào tập hợp (thứ tự chèn).

• Cấu trúc List chứa các phần tử cho phép trùng lặp  
  
• ArrayList lưu trữ các phần tử trong mảng :

- Arraylist là 1 mảng để lưu trữ các phần tử bên trong, có thể điều chỉnh đc kích thước tùy theo số lượng các phần tử bên trong.

- Có thể lưu được nhiều kiểu dữ liệu

-Arraylist đc sử dụng khi muốn truy xuất (nhanh) các phần tử ngẫu nhiên bên trong nhiều và ít thêm, xóa các phần tử(chậm)

• LinkedList lưu trữ các phần tử theo cơ chế liên kết giữa các phần tử:

-LinkedList lưu trữ các phần tử sử dụng cơ chế liên kết theo các node

-LinkedList đc sử dụng khi muốn thêm hay xóa các phần tử bên trong (nhanh) nhìêu và ít truy xuất ngẫu nhiên(chậm)

Generic là cơ chế cho phép truyền kiểu dữ liệu vào như là tham số  
cho các lớp, interface và phương thức

-Stack là 1 cấu trúc dữ liệu danh sách, trong đó việc thêm và lấy các phần tử đc thực hiện theo quy tắc FILO(first in/last out)(vào trc/ra sau)

-Stack đc sử dụng khi chúng ta muốn lưu trữ dữ liệu theo dạng ngăn xếp

• Sử dụng ArrayList để triển khai Stack hiệu quả hơn là sử dụng  
LinkedList

-Queue là 1 cấu trúc dữ liệu danh sách, trong đó việc thêm và lấy các phần tử được thực hiện theo quy tắc FIFO(first in/firstout)(vào trc/ra tr)

-Queue đc sử dụng khi chúng ta muốn lưu trữ dữ liệu kiểu hàng đợi

• Sử dụng LinkedList để triển khai Queue hiệu quả hơn là sử dụng  
ArrayList

Map

-Map là cấu trúc dữ liệu lưu trữ danh sách các cặp key/value.

-Map k cho phép 2 key trùng lặp

-Map đc sử dụng khi phải tìm kiếm, cập nhật hoặc xóa các phần tử dựa vào các key.

Map lưu trữ dữ liệu theo từng cặp key/value  
• Mỗi cặp key/value được gọi là một Entry  
• Thao tác truy xuất sử dụng key có hiệu suất cao

• Java Collection Framework cung cấp 3 lớp Map là: HashMap,  
LinkedHashMap và TreeMap

+hashMap: tối ưu cho các thao tác tìm kiếm, thêm và xóa, k hỗ trợ sắp xếp

+linkedHashMap: kế thừa hashMap, hỗ trợ sắp xếp các entry (key, value)

+treeMap: tối ưu cho thao tác duyệt qua các entry theo trật tự sắp xếp của các key

• Tree lưu trữ dữ liệu theo các node có liên kết cha-con với nhau  
• Cây nhị phân là cây mà mỗi node có thể có 0, 1 hoặc 2 cây con

Tìm kiếm

Tìm kiếm trên danh sách để tìm trong một tập hợp một phần tử  
chứa một khoá nào đó.  
• Tìm kiếm tuyến tính là hoạt động tìm kiếm liên tiếp được diễn ra  
qua tất cả các phần tử.  
• Tìm kiếm nhị phân được thực hiện trên mảng đã được sắp xếp, bằng  
cách so sánh phần tử cần tìm với phần tử tại ví trí giữa nhất của tập  
dữ liệu.  
• Để đánh giá hiệu quả của một thuật toán, có thể xét số các phép  
tính phải thực hiện khi thực hiện thuật toán  
• Thông thường số các phép tính được thực hiện phụ thuộc vào cỡ  
của bài toán, tức là độ lớn của đầu vào.

**THUẬT TOÁN SẮP XẾP**

Có 3 loại cơ bản:

-Sắp xếp nổi bọt: so sánh các cặp phần tử liền kề nhau và tráo đổi thứ tự của chúng nếu chúng k theo thứ tự

-Sắp xếp chọn: chọn phần tử nhỏ nhất/lớn nhất trong dãy các phần tử ban đầu,đưa phần tử này về vị trí đầu dãy,sau đó loại nó khỏi danh sách sắp xếp tiếp theo rồi cứ lặp lại cho đến hết dãy.

-Sắp xếp chèn: chèn thêm 1 phần tử vào danh sách con đã qua sắp xếp.phần tử đc chèn vào vị trí thik hợp sao cho vẫn đảm bảo danh sách con đó vẫn sắp xếp theo thứ tự

1. **Exception- Ngoại lệ**

**NGOẠI LỆ (EXCEPTION) LÀ GÌ?**

-Là các lỗi phát sinh trong quá trình thực thi

-Các kiểu lỗi thường xảy ra:

+Lỗi cú pháp

+Lỗi khi chạy chương trình

+Lỗi về tính logic cấu trúc của chương trình

**CHECK VÀ UNCHECK LÀ GÌ?**

-Check: lỗi đc phát sinh và đc kiểm tra trong quá trình biên dịch

-Uncheck: lỗi đc phát sinh sau khi chạy chương trình

**XỬ LÝ NGOẠI LỆ LÀ GÌ? CÓ BN CÁCH XỬ LÝ NGOẠI LỆ**

-Xử lý ngoại lệ là cách xử lý lỗi phát sinh để chương trình có thể tiếp tục chạy thay vì dừng chương trình khi gặp lỗi

-Các cách xử ly ngoại lệ:

+sử dụng khối try-catch: dùng để bắt ngoại lệ và xử lý chúng

+sủ dụng từ khóa throw: (new) dùng để ném ngoại lệ, đc dùng khi cần phát sinh ngoại lệ trong 1 phương thức

+sủ dụng từ khóa throws: dùng để ném ngoại lệ, đc dùng vs phương thức để gây ra bất cứ check hoặc uncheck exception nào và đồng thời nhường công việc xử lý ngoại lệ đó cho người gọi phương thức đó

**Ghi và đọc dữ liệu**

Các hoạt động nhập/xuất dữ liệu như nhập dữ liệu từ bàn phím, đọc  
dữ liệu từ file, ghi dữ liệu màn hình, ghi ra file, ghi ra đĩa, ghi ra máy  
in ...được gọi là stream (dòng).  
• Có 2 kiểu luồng trong Java: dòng bye (dòng nhị phân) và dòng  
character (dòng văn bản)  
• Lớp File thuộc gói java.io đại diện cho một file hoặc một thư mục.  
• Sử dụng lớp FileWriter thực hiện ghi chuỗi ký tư ra file text  
• Sử dụng lớp FileReader thực hiện đọc file text và BufferReader để  
tăng hiệu quả đọc.