**1.Đặc điểm của List(List,ArrayList,LinkedList,Vector,Stack)**

-List là một collection trong Java cho phép lưu trữ các phần tử trùng lặp

-Các phần tử lưu trữ theo thứ tự mà chúng đc them vào

-Có thể truy cập các phần tử trong List dựa vào index với chỉ số bắt đầu từ 0

-Kích thước của List có thể thay đổi,thêm, xóa linh hoạt

\*\*\*\*\*\*

**2.Các triển khai của List**

**1.ArrayList:**

**- Đặc điểm:**

* Dựa trên mảng (array) động, kích thước có thể thay đổi.
* Truy cập phần tử nhanh do truy cập trực tiếp qua chỉ số.
* Chèn, xóa phần tử giữa danh sách chậm hơn do phải dịch chuyển các phần tử sau vị trí chèn/xóa.
* Sử dụng cần truy cập nhanh đến các phần tử bằng chỉ số,ít thay đổi về dữ liệu

**2.LinkedList:**

**- Đặc điểm:**

* Dựa trên cấu trúc danh sách liên kết,mỗi phần tử lưu trữ con trỏ đến phần tử tiếp theo và trước đó
* Truy cập chậm hơn so với arrayList
* Thêm,xóa phần tử bất kì ở vị trí nào nhanh hơn
* Sử dụng khi cần thêm,xóa phần tử trong danh sách thường xuyên

**3.Vector:**

**-Đặc điểm:**

* Giống với ArrayList nhưng các phương thức của Vector được đồng bộ,an toàn với môi trường đa luồng
* Hiệu xuất chậm hơn ArrayList
* Sử dụng khi cần danh sách an toàn cho đa luồng mà không muốn tự quản lý đồng bộ hóa,tuy nhiên Vector cũng ít đc sử dụng do hiệu xuất kém

**4.Stack:**

**- Đặc điểm:**

* Là triển khai kiểu ngăn xếp dựa tren vector,với các phương thức đặc trưng như push,pop,peek
* Được đồng bộ hóa
* Sử dụng khi cần cấu trúc ngăn xếp

\*\*\*\*\*\*\*\*\*

**Các trường hợp sử dụng:**

* **ArrayList:** sử dụng khi cần truy cập ngẫu nhiên vào phần tử bằng chỉ số,thích hợp cho việc đọc và lấy dữ liệu hơn là cập nhật danh sách.vd:Lưu danh sách các đối tượng học sinh,sinh viên,..
* **LinkedList:**lưu đối tượng theo cặp liên kết đôi,thích hợp khi muốn thêm sửa xóa ở đầu,giữa,cuối bảng
* **Vector:**như ArrayList nhưng an toàn trong môi trường đa luồng,tuy nhiên đã lỗi thời và ít được sử dụng

**Một vài phương thức trong List:**

* add(E e): Thêm một phần tử vào cuối danh sách.
* add(int index, E element): Chèn một phần tử vào vị trí chỉ định.
* remove(int index): Xóa phần tử tại vị trí chỉ định.
* remove(Object o): Xóa phần tử đầu tiên tìm thấy trong danh sách.
* get(int index): Trả về phần tử tại vị trí chỉ định.
* set(int index, E element): Thay đổi phần tử tại vị trí chỉ định.
* size(): Trả về số lượng phần tử trong danh sách.
* isEmpty(): Kiểm tra xem danh sách có rỗng hay không.
* contains(Object o): Kiểm tra xem danh sách có chứa phần tử chỉ định hay không.
* indexOf(Object o): Trả về chỉ số của phần tử đầu tiên tìm thấy trong danh sách hoặc -1 nếu không tìm thấy.
* clear(): Xóa tất cả các phần tử trong danh sách.
* iterator(): Trả về một Iterator để duyệt qua danh sách.

**1. Iterator**

* **Iterator** là một interface trong Java, được sử dụng để lặp qua các phần tử của một Collection. Iterator cung cấp cách thức truy cập tuần tự đến các phần tử trong Collection mà không cần quan tâm đến cách dữ liệu được lưu trữ bên trong.
* Iterator thường được sử dụng với List, Set, và Queue.

**Các phương thức chính của Iterator:**

* boolean hasNext(): Kiểm tra xem còn phần tử nào trong Collection để lặp hay không.
* E next(): Trả về phần tử tiếp theo trong Collection.
* void remove(): Xóa phần tử hiện tại khỏi Collection (không phải tất cả các Collection đều hỗ trợ phương thức này).