**Các đặc điểm chính của List:**

Thứ tự (Order): List duy trì thứ tự thêm của các phần tử. Khi duyệt qua, phần tử được in ra theo thứ tự đã chèn vào.

Chỉ số (Index): Mỗi phần tử trong List đều được gán một index bắt đầu từ 0, cho phép truy cập ngẫu nhiên thông qua chỉ số.

Cho phép trùng lặp: Một List có thể chứa nhiều phần tử giống nhau.

Hỗ trợ các phương thức thao tác: Thêm, xóa, cập nhật phần tử theo index và các phương thức tìm kiếm như indexOf() và lastIndexOf().

Triển khai từ Collection: List kế thừa từ interface Collection, do đó nó cũng có các phương thức cơ bản như add(), remove(), và size().

**Các class triển khai từ List Interface**

ArrayList: Sử dụng mảng động để lưu trữ các phần tử, hiệu quả khi truy cập theo chỉ số.

LinkedList: Sử dụng danh sách liên kết kép, thích hợp khi cần thêm/xóa nhiều phần tử.

Vector: Giống ArrayList nhưng đồng bộ hóa (thread-safe), phù hợp trong môi trường đa luồng.

Stack: Kế thừa từ Vector, cung cấp thêm các thao tác cho cấu trúc ngăn xếp (LIFO).

**1. ArrayList**

Đặc điểm:

* Dùng mảng động để lưu trữ.
* Truy cập phần tử theo chỉ số nhanh
* Khi thêm/xóa ở giữa mảng, cần dịch chuyển các phần tử, làm giảm hiệu suất.

Khi nào dùng?

* Khi cần truy cập nhanh vào các phần tử theo chỉ số.
* Dữ liệu ít thay đổi, không cần chèn hoặc xóa thường xuyên.

2. LinkedList

Đặc điểm:

* Sử dụng danh sách liên kết kép.
* Thêm, xóa phần tử ở đầu hoặc giữa nhanh hơn
* Truy cập phần tử theo chỉ số chậm hơn vì phải duyệt tuần tự.

Khi nào dùng?

* Khi cần thêm/xóa nhiều phần tử ở các vị trí khác nhau trong danh sách.
* Khi dữ liệu có thể thay đổi thường xuyên.

3. Vector

Đặc điểm:

* Giống ArrayList nhưng đồng bộ hóa (synchronized), an toàn trong môi trường đa luồng.
* Tuy nhiên, vì đồng bộ hóa, hiệu suất thấp hơn so với ArrayList.

Khi nào dùng?

* Khi cần danh sách an toàn trong môi trường đa luồng.

4. Stack

Đặc điểm:

* Kế thừa từ Vector, hỗ trợ các thao tác cho ngăn xếp (LIFO - Last In, First Out).
* Cung cấp các phương thức đặc biệt như push(), pop(), và peek() để thao tác với ngăn xếp.

Khi nào dùng?

* Khi cần ngăn xếp để quản lý dữ liệu theo kiểu LIFO, chẳng hạn như kiểm tra dấu ngoặc trong biểu thức hoặc duyệt cây nhị phân.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | Cấu trúc | Truy cập | Thêm/Xóa | Đồng bộ hóa | Khi nào dùng |
| Array List | Mảng động | Nhanh | Chậm khi chèn/xóa | Không | Khi cần truy cập nhanh theo index |
| Linked List | Danh sách liên kết kép | Chậm | Nhanh | Không | Khi cần thêm/xóa nhiều phần tử |
| Vector | Mảng động | Nhanh | Chậm khi chèn/xóa | Có | Khi cần an toàn trong đa luồng |
| Stack | Ngăn xếp (kế thừa Vector) | Theo LIFO | Nhanh với push/pop | Có | Khi cần cấu trúc ngăn xếp LIFO |