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

ArrayList là lớp triển khai của Interface List. Nó cho phép tạo ra các mảng động (dynamic) có thể tự động thay đổi kích thước khi chạy. Điều này có nghĩa là bạn có thể thêm hoặc xóa các phần tử trong mảng một cách linh hoạt mà không cần phải lo lắng về kích thước của mảng.

**Sự giống và khác nhau giữa ArrayList và Array:**

**Giống nhau:**

* Cả ArrayList và Array đều được sử dụng để lưu trữ các phần tử.
* Các phần tử trong cả hai đều có thể được truy cập bằng chỉ số (index).

**Khác nhau:**

* **Kích thước:**
  + Array: Có kích thước cố định. Khi khai báo một mảng, bạn phải chỉ định kích thước của nó và không thể thay đổi kích thước sau này.
  + ArrayList: Có kích thước thay đổi. Bạn có thể thêm hoặc xóa các phần tử một cách linh hoạt, và kích thước của nó sẽ tự động điều chỉnh.
* **Hiệu suất:**
  + Array: Có hiệu suất cao hơn khi thực hiện các thao tác truy cập và cập nhật phần tử.
  + ArrayList: Có hiệu suất thấp hơn khi so sánh với Array, do phải duy trì tính linh hoạt của kích thước.
* **Phương thức:**
  + Array: Không có nhiều phương thức hỗ trợ. Bạn phải tự viết các phương thức để thực hiện các thao tác thêm, xoá, tìm kiếm, sắp xếp, v.v.
  + ArrayList: Cung cấp nhiều phương thức hỗ trợ như add(), remove(), get(), set(), contains(), size(), và nhiều phương thức khác để thực hiện các thao tác trên danh sách.

**2. Các method phổ biến khi thao tác với ArrayList**

* **Add (E e):** Thêm phần tử vào cuối danh sách.
* **Add (int index, E e):** Thêm phần tử vào vị trí chỉ định trong danh sách.
* **Clear ():** Xóa tất cả các phần tử khỏi danh sách.
* **Contains (Object o):** Kiểm tra danh sách có chứa phần tử được chỉ định hay không.
* **Get (int index):** Trả về phần tử tại vị trí chỉ định trong danh sách.
* **indexOf (Object o):** Trả về chỉ số của sự xuất hiện đầu tiên của phần tử được chỉ định trong danh sách, hoặc -1 nếu danh sách không chứa phần tử.
* **isEmpty ():** Kiểm tra danh sách có rỗng hay không.
* **Remove (int index):** Xóa phần tử tại vị trí chỉ định trong danh sách.
* **Remove (Object o):** Xóa sự xuất hiện đầu tiên của phần tử được chỉ định khỏi danh sách, nếu có.
* **Set (int index, E e):** Thay thế phần tử tại vị trí chỉ định trong danh sách với phần tử được chỉ định.
* **Size ():** Trả về số phần tử trong danh sách.
* **ToArray ():** Trả về một mảng chứa tất cả các phần tử trong danh sách theo đúng thứ tự.
* **TrimToSize ():** Cắt giảm dung lượng của mảng nội bộ của danh sách xuống mức cần thiết để lưu trữ số lượng phần tử hiện tại của danh sách.
* **EnsureCapacity (int minCapacity):** Tăng kích thước của mảng nội bộ của danh sách, nếu cần, để đảm bảo nó có thể chứa ít nhất minCapacity phần tử.