Collection Framework và Generic

1. **Collection Framework**

Collection Framework là một tập hợp các lớp và giao diện giúp thao tác trên các cấu trúc dữ liệu, như danh sách (list), tập hợp (set), hàng đợi (queue), và bản đồ (map). Các cấu trúc này giúp lưu trữ và quản lý các đối tượng một cách hiệu quả.

1. Các giao diện (Interfaces) trong Collection Framework

* Collection: Là giao diện gốc cho tất cả các tập hợp (collections) khác.
* List: Một tập hợp có thứ tự, có thể chứa các phần tử trùng lặp. Ví dụ: ArrayList, LinkedList.
* Set: Tập hợp không có thứ tự và không chứa phần tử trùng lặp. Ví dụ: HashSet, LinkedHashSet, TreeSet.
* Queue: Được sử dụng để lưu trữ các phần tử theo thứ tự FIFO (First In First Out). Ví dụ: PriorityQueue, ArrayDeque.
* Map: Lưu trữ các cặp khóa-giá trị, mỗi khóa chỉ xuất hiện một lần. Ví dụ: HashMap, TreeMap, LinkedHashMap.

1. Các lớp (Classes) trong Collection Framework

* ArrayList: Một danh sách dựa trên mảng, có thể tăng kích thước khi cần.
* LinkedList: Một danh sách liên kết kép, giúp cho việc chèn và xóa phần tử nhanh chóng.
* HashSet: Một tập hợp không chứa các phần tử trùng lặp và không theo thứ tự.
* TreeSet: Một tập hợp sắp xếp các phần tử theo thứ tự tự nhiên hoặc theo bộ so sánh được chỉ định.
* HashMap: Lưu trữ các cặp khóa-giá trị và không đảm bảo thứ tự.
* TreeMap: Sắp xếp các cặp khóa-giá trị theo thứ tự khóa.

1. Một số phương thức quan trọng

* add(E e): Thêm phần tử vào tập hợp.
* remove(Object o): Xóa phần tử khỏi tập hợp.
* contains(Object o): Kiểm tra xem tập hợp có chứa phần tử chỉ định không.
* size(): Trả về số lượng phần tử trong tập hợp.
* iterator(): Trả về một Iterator để duyệt qua các phần tử trong tập hợp.

1. **Generic**

Generic cho phép các lớp, giao diện, và phương thức làm việc với các kiểu dữ liệu khác nhau mà không cần chỉ định kiểu cụ thể tại thời điểm viết mã, mà sẽ chỉ định tại thời điểm sử dụng. Điều này giúp đảm bảo tính an toàn về kiểu dữ liệu (type safety) và giảm sự dư thừa mã.

Lợi ích khi dùng generic :

* Type Safety: Đảm bảo rằng chỉ những kiểu dữ liệu tương thích mới có thể được thêm vào tập hợp.
* Code Reusability: Viết mã một lần và sử dụng với nhiều loại dữ liệu khác nhau.

Ví dụ :

public <T> void printArray(T[] array) {

for (T element : array) {

System.out.println(element);

}

}

Hoặc sử dụng generic trong arraylish:

ArrayList<String> list = new ArrayList<>();

list.add("Java");

list.add("Collection");