**COLLECTION**

**Iterable là gì ?**

* Interface **Iterable** là một dại diện cho một tập hợp các đối tượng có thể lặp lại – nghĩa là có thể được lặp lại. Điều này có nghĩa là một lớp thực thi Interface **Iterable** có thể lặp lại các phần tử của nó.

**Collection là gì ?**

Trong java Collection là các interfaces và class đặc biệt dùng để lưu trữ và đọc dữ liệu theo nhiều cách khác nhau. Có thể hình dung các collection là tập hợp các object, tùy vào mục đích sử dụng, mục đích lưu trữ mà sẽ được chọn các collection phù hợp.

**List**

* Dùng để lưu trữ tập hợp các phần tử một cách có thứ tự. List có thể chứa các phần tử trùng lặp. List có thể truy cập các phần tử bằng chỉ số (Vị trí của chúng).
* **ArrayList**: được sử dụng như một mảng để lưu trữ các phần tử và các phần tử trong đấy có thể trùng lặp. Duy trì thứ tự của phần tử được thêm vào. Nó lưu trữ theo các chỉ mục.
* **Vector**: nó hoạt động như ArrayList. Tuy nhiên nó không phân biệt kiểu dữ liệu như ArrayList và khi đạt dung lượng tối đa mà thêm tiếp phần tử thì bộ nhớ sẽ tăng 100% so với bộ nhớ ban đầu.
* **LinkedList**: Lưu trữ các phần tử một cách tuần tự, các thao tác như thêm xóa rất nhanh vì không phải dịch chuyển nếu bất kỳ phần tử nào thêm/xóa khỏi danh sách.

**Set**

* là một interface kế thừa Collection interface trong java. Set trong java là một Collection không thể chứa các phần tử trùng lặp.
* **HashSet**: Lưu các phần tử không theo thứ tự. Và có thể null.
* **TreeSet**: Lưu các phần tử có thứ tự. Và không thể null.

**Queue**

* Cũng giống như hàng đợi trong thế giới thực (ví dụ: trong ngân hàng hoặc tại ATM), Hàng đợi chèn các phần tử vào cuối hàng và loại bỏ khỏi đầu hàng.
* Queue đại diện cho một danh sách các phần tử có thứ tự.

**Map**

* Lưu trữ và và truy xuất dữ liệu theo cặp key và value.
* **HashMap**: Lưu các phần tử không theo thứ tự và có thể chứa 1 key null và nhiều value null.
* **TreeMap**: Lưu các phần tử theo thứ tự và không chứa 1 key null nhưng chứa được nhiều value null.