Trần Trung Hiếu – Báo cáo thực tập

26 – 30/12/2022

Java Collection

Java Collection là một framework cung cấp kiến trúc để lưu trữ và thao tác với nhóm đối tượng.

Java Collection có thể thực hiện tất cả các thao tác mà bạn thực hiện trên dữ liệu như tìm kiếm, sắp xếp, chèn, thao tác và xóa.

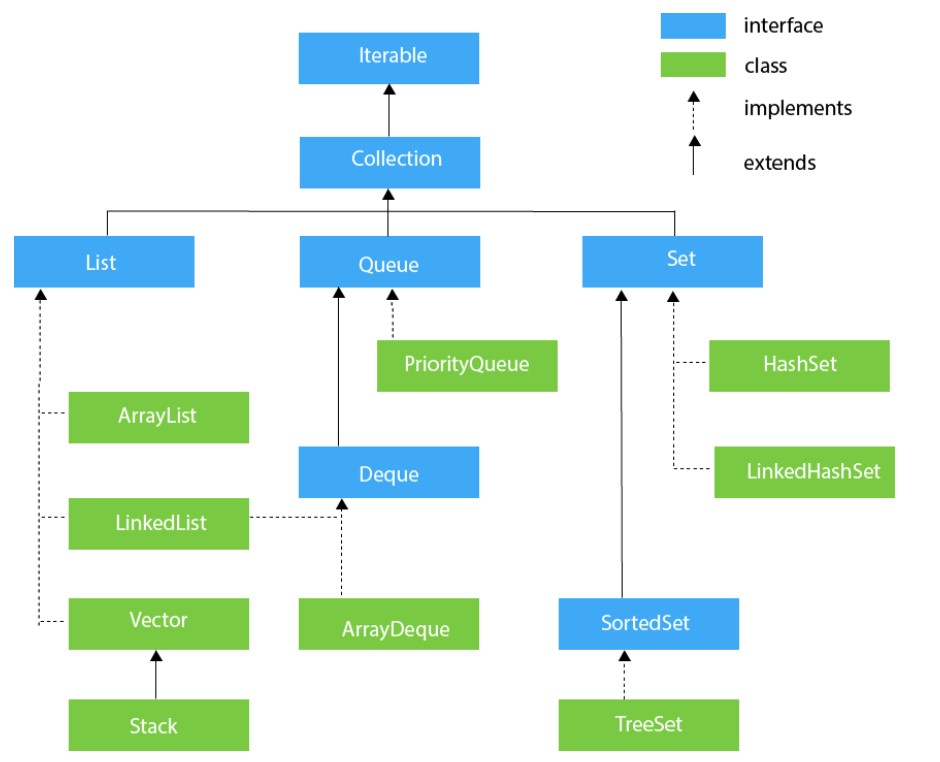
Java Collection framework cung cấp nhiều interface (Set, List, Queue, Deque) và các lớp (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).

Framework trong Java là gì

* Cung cấp kiến trúc làm sẵn.
* Đại diện cho một tập hợp các lớp và interface.
* Optional, không bắt buộc.

Gói java.util chứa tất cả các lớp và interface cho Java Collection framework.

Hệ thống java collection:



1. Iterable Interface

Iterable là interface gốc cho tất cả các lớp Java Collection. Java Collection interface mở rộng Iterable interface và do đó tất cả các lớp con của Java Collection interface cũng triển khai (implement) Iterable interface.

Iterable Interface chỉ chứa một phương thức trừu tượng,

Iterator<T> iterator()

Trả về iterator trên các phần tử của kiểu T.

1. Collection Interface

Collection Interface là interface được triển khai bởi tất cả các lớp trong Java Collection framework. Collection Interface xây dựng nền tảng mà Collection framework phụ thuộc vào.

Một số phương thức Collection Interface là Boolean add ( Object obj), Boolean addAll ( Collection c), void clear(), v.v. được triển khai bởi tất cả các lớp con của Collection framework.

1. List Interface

List Interface là interface con của Collection Interface. Nó bao gồm các cấu trúc dữ liệu kiểu danh sách trong đó chúng ta có thể lưu trữ tập đối tượng được sắp xếp theo thứ tự. Nó có thể có các giá trị trùng lặp.

Được implement bởi lớp ArrayList, LinkedList, Vector và Stack.

List <data-type> list1= new ArrayList();

List <data-type> list2 = new LinkedList();

List <data-type> list3 = new Vector();

List <data-type> list4 = new Stack();

3.1. ArrayList

ArrayList sử dụng một mảng động để lưu trữ phần tử trùng lặp của các kiểu dữ liệu khác nhau. Lớp ArrayList duy trì thứ tự chèn và không được đồng bộ hóa. Các phần tử được lưu trữ trong lớp ArrayList có thể được truy cập ngẫu nhiên.

3.2. LinkedList

LinkedList sử dụng một danh sách liên kết kép để lưu trữ các phần tử. Nó có thể lưu trữ các yếu tố trùng lặp. Nó duy trì thứ tự chèn và không được đồng bộ hóa. Trong LinkedList, thao tác rất nhanh chóng vì không cần dịch chuyển (shifting).

3.3. Vector

Vector sử dụng một mảng động để lưu trữ các phần tử dữ liệu. Nó tương tự như ArrayList. Tuy nhiên, Nó được đồng bộ hóa và chứa nhiều phương thức không phải là của Collection framework.

3.4. Stack

Stack là lớp con của Vector. Nó thực hiện cấu trúc dữ liệu nhập last-in-first-out. Stack chứa tất cả các phương thức của lớp Vector và cũng cung cấp các phương thức riêng của nó như boolean Push(), boolean peek(), boolean Push(object o).

1. Queue Interface

Queue Interface duy trì thứ tự first-in-first-out. Nó có thể được định nghĩa là một danh sách có thứ tự được sử dụng để chứa các phần tử sắp được xử lý. Có nhiều lớp khác nhau như PriorityQueue, Deque và ArrayDeque.

Queue<String> q1 = new PriorityQueue();

Queue<String> q2 = new ArrayDeque();

4.1. PriorityQueue

PriorityQueue chứa các phần tử hoặc đối tượng sẽ được xử lý theo mức độ ưu tiên của chúng. PriorityQueue không cho phép lưu trữ các giá trị null trong hàng đợi.

4.2. Deque Interface

Trong Deque, chúng ta có thể xóa và thêm các phần tử từ cả hai phía. Deque là viết tắt của hàng đợi hai đầu cho phép chúng ta thực hiện các thao tác ở cả hai đầu.

4.3. ArrayDeque

Lớp ArrayDeque tạo điều kiện sử dụng Deque trong array. Không giống như queue, chúng ta có thể thêm hoặc xóa các phần tử ở cả hai đầu.

ArrayDeque nhanh hơn ArrayList và Stack và không bị giới hạn dung lượng.

1. Set Interface

Set Interface trong Java có trong gói java.util. Nó đại diện cho tập hợp các phần tử không có thứ tự và không cho phép lưu trữ các mục trùng lặp. Có thể lưu trữ nhiều nhất một giá trị null trong Set. Set được triển khai bởi HashSet, LinkedHashSet và TreeSet.

5.1. Hashset

Lớp HashSet đại diện cho tập sử dụng bảng băm để lưu trữ. Băm được sử dụng để lưu trữ các phần tử trong HashSet.

5.2. LinkedHashSet

LinkedHashSet là mở rộng của lớp HashSet. Giống như HashSet, Nó cũng chứa các phần tử riêng biệt (unique items). Nó duy trì thứ tự chèn và cho phép các phần tử null.

1. SortedSet Interface

SortedSet là giao diện thay thế của Set cung cấp tổng thứ tự trên các phần tử của nó. Các phần tử của SortedSet được sắp xếp theo thứ tự tăng dần (tăng dần). SortedSet cung cấp các phương thức bổ sung ngăn chặn thứ tự tự nhiên của các phần tử.

SortedSet<data-type> set = new TreeSet();

6.1. TreeSet

Lớp Java TreeSet triển khai giao diện Set sử dụng cây để lưu trữ. Giống như HashSet, TreeSet cũng chứa các unique items. Tuy nhiên, thời gian truy cập và truy cập dữ liệu của TreeSet khá nhanh. Các phần tử trong TreeSet được lưu trữ theo thứ tự tăng dần.