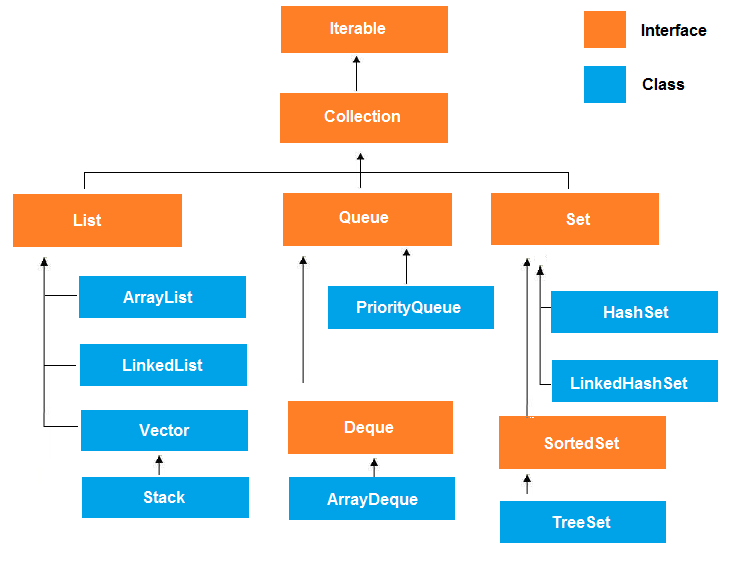
**Set và những thư viện trong SetInterface**



Set Interface trong java là 1 Collection không chứa các bản sao phần tử.

Set Interface thừa kế từ Collection và bổ xung thêm sự giới hạn về việc ngăn cấm phần tử bản sao. ( hay nói cách khác nó chỉ chứa những giá trị riêng biệt và không trùng lặp).

|  |  |
| --- | --- |
| **STT** | **Phương thức và Miêu tả** |
| 1 | **add( )**  Thêm một đối tượng tới Collection đó |
| 2 | **clear( )**  Gỡ bỏ tất cả đối tượng từ Collection đó |
| 3 | **contains( )**  Trả về true nếu một đối tượng đã cho là một phần tử bên trong Collection đó |
| 4 | **isEmpty( )**  Trả về true nếu Collection không có phần tử nào |
| 5 | **iterator( )**  Trả về một đối tượng Iterator cho Collection mà có thể được sử dụng để thu nhận một đối tượng |
| 6 | **remove( )**  Gỡ bỏ một đối tượng đã xác định từ Collection đó |
| 7 | **size( )**  Trả về số phần tử trong Collection đó |

- Các lớp triển khai của SetInterface: HashSet, TreeSet, LinkedHashSet.

\* HashSet: kế thừa từ AbstractSet và triển khai Set Interface. HashSet cung cấp 4 constructor:

- HashSet(): tạo một set mặc định rỗng.

- HashSet(Collection c): tạo 1 HashSet từ Collection c cho trước.

- HashSet(int count): tạo 1 HashSet với số phần từ cho trước . số này sẽ tự tang nên khi kích cỡ không đủ để add.

- HashSet(int capacity, float fillRadio): bổ xung thêm biến dung lượng tải từ (0 -> 1) .

- Ngoài những phương thức triển khai từ SetInterface HashSet còn bổ xung thêm những phương thức riêng cho nó như: clone(), iterator(), clear().

\* LinkedHashSet: là một cải tiến của HashSet hay nói cách khác nó kế thừa từ HashSet. LinkedHashSet duy trì 1 linked list của các entry trong Set theo thứ tự chúng như đã đc chèn(Linked không chỉ số). Khi chèn thứ tự không bị đảo lộn.

- LinkedHashSet cũng cung cấp 4 constructor giống như HashSet.

\* TreeSet: cũng như HashSet, kế thừa từ lớp AbstractSet và triển khai Set Interface, nhứng điểm khác biệt là TreeSet sử dụng tree cho lưu giữ các đối tượng được sắp xếp tang dần.

- TreeSet cung cấp 4 constructor cho người dùng là:

+ TreeSet(): constructor mặc định.

+ TreeSet(Collection c): tạo TreeSet từ c.

+ TreeSet(Comparator comp): tạo tree trống và các phần từ sẽ được sắp xếp theo bộ so sánh được xác định bởi comp.

+ TreeSet(SortedSet ss): xây dụng set từ các phần tử của ss

- Ngoài những phương thức kế thừa sẵn có , TreeSet cũng bổ sung thêm những phương thức của riêng nó: (những phương thức gần giống nhau sử dụng cấu trúc dữ liệu tree head,sub,tail):

|  |  |
| --- | --- |
| **STT** | **Phương thức và Miêu tả** |
| 1 | **void add(Object o)**  Thêm phần tử đã cho tới Set này nếu nó chưa có mặt |
| 2 | **boolean addAll(Collection c)**  Thêm tất cả phần tử trong collection đã cho tới Set này |
| 3 | **void clear()**  Gỡ bỏ tất cả phần tử từ Set này |
| 4 | **Object clone()**  Trả về một shallow copy của đối tượng TreeSet này |
| 5 | **Comparator comparator()**  Trả về bộ so sánh được sử dụng để xếp thứ tự Sorted set này, hoặc null nếu tree set sử dụng cách xếp thứ tự phần tử tự nhiên |
| 6 | **boolean contains(Object o)**  Trả về true nếu Set này chứa phần tử đã cho |
| 7 | **Object first()**  Trả về phần tử đầu tiên (thấp nhất) hiện tại trong Sorted Set này |
| 8 | **SortedSet headSet(Object toElement)**  Trả về một phần của Set này mà có các phần tử nhỏ hơn toElement |
| 9 | **boolean isEmpty()**  Trả về true nếu Set này không chứa phần tử nào |
| 10 | **Iterator iterator()**  Trả về một Iterator qua các phần tử trong Set này |
| 11 | **Object last()**  Trả về phần tử cuối cùng (cao nhất) hiện tại Sorted Set này |
| 12 | **boolean remove(Object o)**  Gỡ bỏ phần tử đã cho từ Set này nếu nó có mặt |
| 13 | **int size()**  Trả về số phần tử trong Set này |
| 14 | **SortedSet subSet(Object fromElement, Object toElement)**  Trả về một phần Set này mà có các phần tử từ fromElement toElement |
| 15 | **SortedSet tailSet(Object fromElement)**  Trả về một phần Set này mà có phần tử lớn hơn hoặc bằng fromElement |