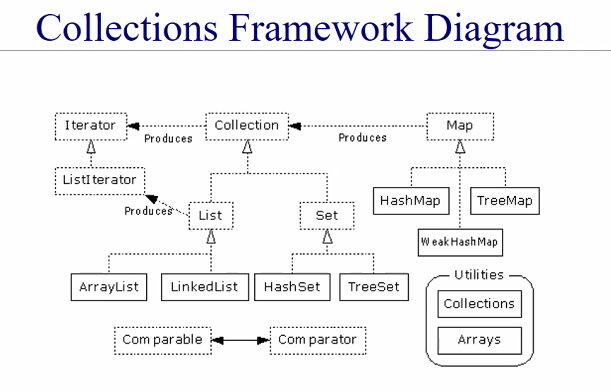
Sơ đô Collections

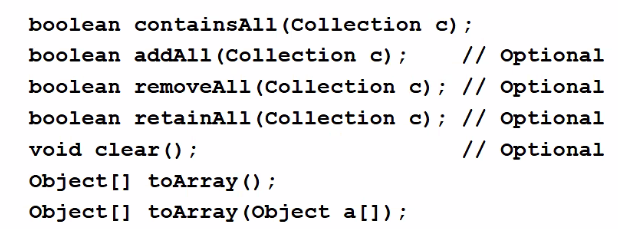


Nét đứt: interface

Nét liền: class.

Iterator: duyệt các phần tử trong Collection. Cơ chế hasNext().

|  |  |
| --- | --- |
| SET:  + Không trùng nhau. | |
| HASH\_SET  +Là cơ chế hàm băm. Truy xuất phần tử nhanh. Tra theo chuỗi băm chứ ko phải tra/ tìm kiếm theo tham chiếu  + Hướng đến mục tiêu truy xuất và tìm kiếm  + Equals như nào là do mình viết. | TREE SET  + Tree\_set thì nó sẽ ưu tiên sắp xếp.  + Cài đặt trên cây đỏ đen.  + khi lấy ra các phần tử thì sẽ đưa ra thứ tự tăng dần. Sử dụng cơ chế so sánh compareTo(object o) để đưa vào cây.  + tìm kiếm và thêm chậm hơn so với hashset.  + Nếu dùng hashSet mà sắp xếp thì nên dùng treeSet thì sẽ nhanh hơn, dù cho tìm kiếm và thêm có thể chậm hơn xíu so với Hashset. |
|  |  |
| MAP  + mục tiêu lưu trữ cặp dữ liệu key/value. Key duy nhất (set)  + thay vì làm trên 1 mảng đối tượng thì làm việc trên key 🡪 value  + Dành cho tìm kiếm  >> put(key, value)  >>get(key)  >>remove(key)  >>containsKey(key) 🡪 boolearn  >>containsValue(value) 🡪 boolearn  >>int size()  >> boolean isEmpty()  >> Set keySet():  >> Collection values  >> Set entrySet() | |
| Hash\_map  + tìm kiếm dữ liệu rất nhanh, nhanh nhất có thế. (chìa khóa về BigData)  + | Tree\_Map  + nhấn mạnh tính sắp xếp(cây đỏ đen)  + compareTo (object). Sắp xếp key |



🡪 basic của Collection

Các lớp tiện ích hỗ trợ collection: (cài đặt các thuật toán cơ bản)

|  |  |  |
| --- | --- | --- |
|  | Ngăn xếp | Hàng đợi |
| Khai báo | >>push  >>pop  >>peek: giá trị đầu tiên  >>empty | - là 1 interface, không phải một class    >>add: đưa vào  >>poll: lôi ra  >>offer: đưa vào  >>isEmpty |
| Ý nghĩa | - kiểm soát được cặp mở đóng  - chuyển đổi biểu diễn phép toán: tiền tố trung hậu tố  - kiểm soát các giá trị nhỏ hơn hoặc lớn hơn bên phải  - |  |