**Linked list**

* Thêm phần tử cho Linked list?

LinkedList<String> linkedList = new LinkedList<String>();

linkedList.add(“abc”);

* Lấy phần tử đầu tiên và phần tử cuối cùng của Linked list?

linkedList.getFirst();

linkedList.getLast();

* Thêm 1 phần tử vào vị trí đầu tiên và vị trí cuối cùng của Linked list?

linkedList.addFirst(“abc”);

linkedList.addLast(“abc”);

* Tìm kiếm 1 phần tử trong Linked list?

linkedList.contains(“abc”);

* Update 1 phần tử trong Linked list?

linkedList.set(1 , “abc”);

* Xoá nhiều phần tử trong Linked list?

linkedList.remove(“abc”);

**ArrayList**

* Convert Array sang Arraylist?

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

        // Array to ArrayList conversion

        for (String item : arr) {

            list.add(item);

       }

* Conver ArrayList sang Array?

 Integer[] arr = new Integer[list.size()];

        arr = list.toArray(arr);

* Reverse các phần tử trong ArrayList?

Để đảo ngược thứ tự của các phần tử trong List được chỉ định có thể được sử dụng hàm reverse()  trong class java.util.Collections

* Thay thế 1 phần tử trong ArrayList?

Để thay thế 1 phần tử trong ArrayList có thể sử dụng phương thức set(int index , value)

Ôn tập:

1, So sánh khác nhau giữa ArrayList và Linked list? Khi nào nên dung ArrayList, khi nào dùng LinkedList

2, Mối khối try có bắt buộc phải đi kèm 1 khối catch không?

3, Luồng thực thi chương trình của khối try-catch-finally là gì?

4, Khi nào khối finally không được thực thi?

5, Throw và throws dùng để làm gì? So sánh throw throws