Collection



1 The core collection interfaces.

<http://docs.oracle.com/javase/tutorial/collections/interfaces/index.html>

Data Structures consist of the following interface and classes:

1. Enumeration
2. BitSet
3. Vector
4. Stack
5. Dictionary
6. Hashtable
7. Properties (subclass of hashtable)

***Collection***is a data structure in which Objects are stored.

The Collection interface is the foundation upon which the collections framework is built.

Objects can be Added, Deleted and can traversed in Collection.

There are 4 type of basic Collection:

1. **List:** **Ordered**, **Duplicates** are **allowed**, Indexed (inserted or accessed by index)

No need to specify size.

1. **Sets:**May or may not Ordered. **Duplicates** are **not** allowed.
2. **Maps** (keys - values)**: Duplicate** keys are **not** allowed.

Map - an object that maps keys to values. A Map cannot contain duplicate keys; each key can map to at most one value. If you've used Hashtable, you're already familiar with the basics of Map.

1. **Queue: Ordered**by**FIFO**or**priority.**
2. **ArrayList :**Fast Iteration & Fast Random Access.

Dynamic array.

1. **Vector:**Synchronized Method.
2. **LinkedList :**Good for implementing Stack and Queue.

The LinkedList class extends AbstractSequentialList and implements the List interface. LinkedList is a kind of List, and it is also a kind of Queue.

1. **HashSet :**Fast Access, No Duplicates, No Ordering.

HashSet extends AbstractSet and implements the Set interface.

1. **LinkedHashSet :**No Duplicates, Iterates by insertion order.
2. **TreeSet :**No Duplicates, Iterates in sorted order.

TreeSet provides an implementation of the Set interface that uses a tree for storage. Objects are stored in sorted, ascending order.

*Prior to Java 2, Java provided ad hoc classes such as****Dictionary, Vector, Stack****, and****Properties****to store and manipulate groups of objects.*

Code Examples:

**List vs ArrayList vs LinkedList:**

LinkedList best for fast insert and remove.

ArrayList is best for fast search.

***ArrayList****:* [*http://www.tutorialspoint.com/java/java\_arraylist\_class.htm*](http://www.tutorialspoint.com/java/java_arraylist_class.htm)

*ArrayList al = new ArrayList();*

*al.add("C");*

*al.add(1, "A2");*

*al.remove("F");*

*al.size();*

*//We can add and remove by value or index.*

***List****: http://www.tutorialspoint.com/java/java\_list\_interface.htm*

*List a1 = new ArrayList();*

*a1.add("Zara");*

*a1.add("Mahnaz");*

*a1.add("Ayan");*

*System.out.println(" ArrayList Elements");*

*System.out.print("\t" + a1)*

**LinkedList: http://www.tutorialspoint.com/java/java\_linkedlist\_class.htm**

*// create a linked list*

*LinkedList ll = new LinkedList();*

*// add elements to the linked list*

*ll.add("F");*

*ll.add("B");*

*ll.add("D");*

*ll.add("E");*

*ll.add("C");*

*ll.addLast("Z");*

*ll.addFirst("A");*

*ll.add(1, "A2");*

*System.out.println("Original contents of ll: " + ll);*

*// remove elements from the linked list*

*ll.remove("F");*

*ll.remove(2);*

*System.out.println("Contents of ll after deletion: "*

*+ ll);*

*// remove first and last elements*

*ll.removeFirst();*

*ll.removeLast();*

*System.out.println("ll after deleting first and last: "*

*+ ll);*

*// get and set a value*

*Object val = ll.get(2);*

*ll.set(2, (String) val + " Changed");*

*System.out.println("ll after change: " + ll);*

*Hashset:*

*HashSet hs = new HashSet();*

*// add elements to the hash set*

*hs.add("B");*

*hs.add("A");*

*hs.add("C");*

*System.out.println(hs);*