Java Collections

* Collection Interface
  + A collection in java is a framework that provides architecture to store and manipulate a group of objects. We can perform all types of operation on collections that we perform on data. The collection framework consists of Interfaces, classes & algorithms
  + It declares the methods that every collection will have.
  + Collection interface builds the foundation on which collection framework depends.
* Iterable Interface
  + It is the root interface for all collections classes. Since all subclasses of the collection interface implement collection, they also implement iterable interface.
  + It provides of iterating all the elements in forward direction only.
  + It contains only one abstract method
* List Interface
  + This is the child interface of collection interface. It inhibits a list type data structure which can have duplicate values. This collection is implemented by classes such as-
    - ArrayList: This class implements List Interface. It uses dynamic array to store duplicate elements of different data types.
    - LinkedList: It implements the collection interface. It uses doubly linked list to store the elements and can also store duplicate elements. It has a quick manipulation since no shifting is required.
    - Vector: It implements List Interface and uses dynamic array to store elements, just like ArrayList. However it is synchronized and contains many methods which do not belong to the collection framework.
      * Stack: It is a subclass of vector class; it implements LAST IN FIRST OUT data structure. It contains all of the methods which belong to the vector class.
* Queue Interface
  + This maintains FIRST IN FIRST OUT order. It can be defined as a ordered list and is used to hold elements which are about to be processed.
    - Priority Queue: This class Implements the Queue Interface. It holds elements in an order which are to be processed by their priorities.
    - Deque Interface: It extends the Queue Interface, in this elements can be added from the beginning or the end making it a double ended queue which enables us to perform operations on both sides.
      * ArrayDeque: This is a class which implements Deque Interface. It facilitates Deque and has no capacity restrictions. It is faster than ArrayList and Stack.
* Set Interface
  + This extends the collection interface, It is a set of unordered elements which cannot store duplicate elements. It can only store upto 1 null values.
    - Hash Set: It implements set interface. It represents a collection which uses hash table for storage. Hashing is used to store elements in hash set.
    - Linked Hash Set: This represents Linked List implementation of set interface. It extends Hash Set and implements set interface. Like hash set, it also contains unique elements. It maintains inserting order and permits null elements.
    - Sorted Set: It is a alternative of set interface. It arranges elements in ascending (increasing) order. It provides additional methods that inhibit natural ordering of elements.
      * Tree Set: Tree set implements the set interface. Like hash set, it also contains unique elements but its access and retrieval time is fast. It contains elements in ascending order.