Collection API

Java1.5

Vector class

Architecture

Interface → Set of sub -interfaces

Abstract Classes

Implementation class

Generic Based

Collection<E>
Common Functionalities

Iterable
(iterator())

specialized

List

Resizable linear
Array model
indexed access
can have duplicate
values

Set

Set of Unique
Values

Queue

Collection based on
rule (FIFO)

Map

Key-Value
unique key

interface Iterable{

Iterator iterator();

}

interface Collection extends Iterable
{

}

Editable : while traversing through collection (editing is possible)

interface Iterator{

 <method to help

traverse a collection>

next()

hasMoreElements();

remove()

}

Collection → AbstractCollection

List → AbstractList

ArrayList
(Fixed Size)
Array implementation
traversing
(traversal/access)

LinkedList
Linked Implentation
(insertion/deletion)

Vector
(sync)
(Legacy)

Stack (LIFO)

Collection → AbstractCollection

Queue → AbstractQueue

Deque → ArrayDeque

PriorityQueue

Array model of queue/deque

(Fixed Size)

LinkedList

maintain linked impl of queue

Collection → AbstractCollection

Set → AbstractSet

SortedSet   TreeSet   EnumSet   HashSet   LinkedHashSet

NavigableSet

Set : Unique Values

Enums : uniques

Hashing :

HashSet : Table Hashing

LinkedHashSet : Linked Hashing

SortedSet : maintain data in sorted

NavigableSet : Quick Navigation functionalities

Red-Black Tree

Keys must be unique

Map → AbstractMap

SortedMap

TreeMap   EnumMap   HashTable   HashMap

NavigableMap   Key would be enum

LinkedHashMap

HashTable : synchronized
              : HashTable
HashMap

sync : Thread Safe

=> Map interface : Set<k> keySet()

              V get(<K>)

returns the list of all keys as Set<>

iterate through key

Algorithm Classes

    1. Arrays.

    2. Collections.

@ Provide common functionalities can be applied on Array int [] and Collection

 Collections.concurrentList(); // Thread safe equivalent