Collection

- Why collection ?
- Why Array ?
- Limitations of an Array ?
- What is collection ?

Why collection:

to store on a single value we go for int x = 10. for two values, int x=10, y=20; if you want to store 10,000 values!

So, To over come this problem Array came into Picture.

Why Array ?

An array is indexed collection of fixed number of homogenous data elements.

Limitations of Array

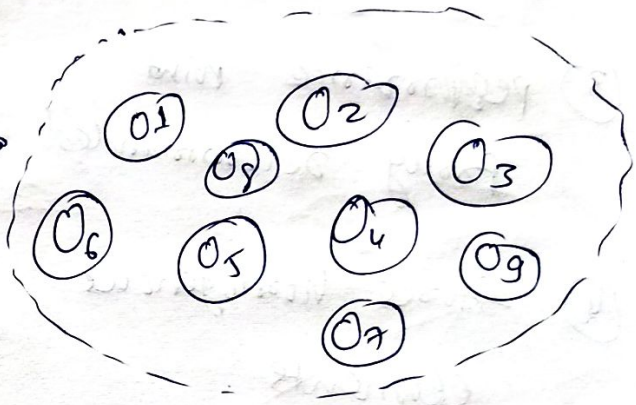
- fixed in size
- Array can store homogenous data elements
- In array ready-made method support is not available.

So, To overcome this limitation collection concept is introduced.

* Collection is one type of containes which stores the group of objects into single variable.
- Group of objects that are stored and manipulated in well manner
- introduced in JDK 1.2

* Collection: group of indusival objects represented as a single Entity

Collection



* Collection framework:

To represent a group of indusival objects as a single entity we need several classes and Interfaces, these are nothing But framework.

* Advantages of collection over the Array

(1) Collection is growable in nature
(2) collection can store heterogenous objects
(3) In collection ready made method support is avalible.

Difference Between Array and Collection

Array	Collection
(1) Fixed size	Growable in size
(2) memory point not good	memory point collection recommended
(3) performance point Array recommanded	performance point collection not recommanded.
(4) Stores homogenous elements	Stores both homogeneous and heterogenous.
(5) NIO, ready made method support	yes, ready made method support avilable.
(6) Array holds Both Primitive and objects	holds objects not Primitives.

Collection vs collections

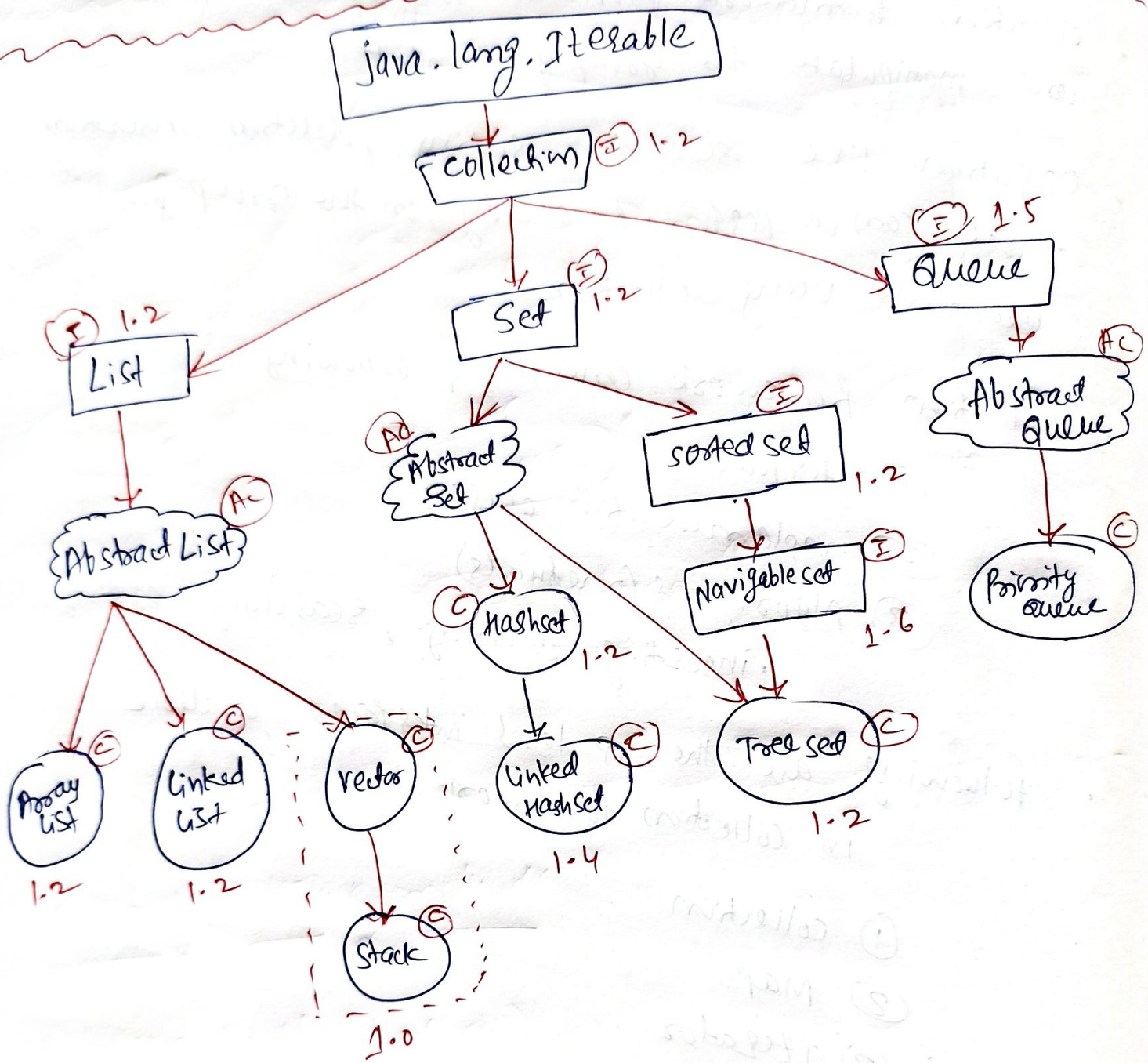
→ Collection is an interface, can be used to represent a group of individual object as a single entity.

→ Collections is an utility class, present in java.util package, to define several utility methods for collections.

Collection framework

- Collection framework provides mechanism to store and manipulate the group of elements
- operations like searching, sorting, deletion, insertion ...etc can be performed easily on the group of elements using collection.
- Collection framework consists of following
  1. Interfaces
  2. implementation classes
  3. Algorithms (methods)
     - inserting, deleting, searching, sorting....
- following are the top level interfaces avalible in collection framework.
  1. collection
  2. Map
  3. Iterator
- Subclasses of collection are used to manage the collection of elements without key-value format
- subclasses of map interface are used to manage the collection of elements with key and value format.
- Iterator is used to access data from the collection.

* Collection Hierarchy *



- collection Interface is root Interface of collection Hierarchy
- There are 9 Interfaces and are follows.

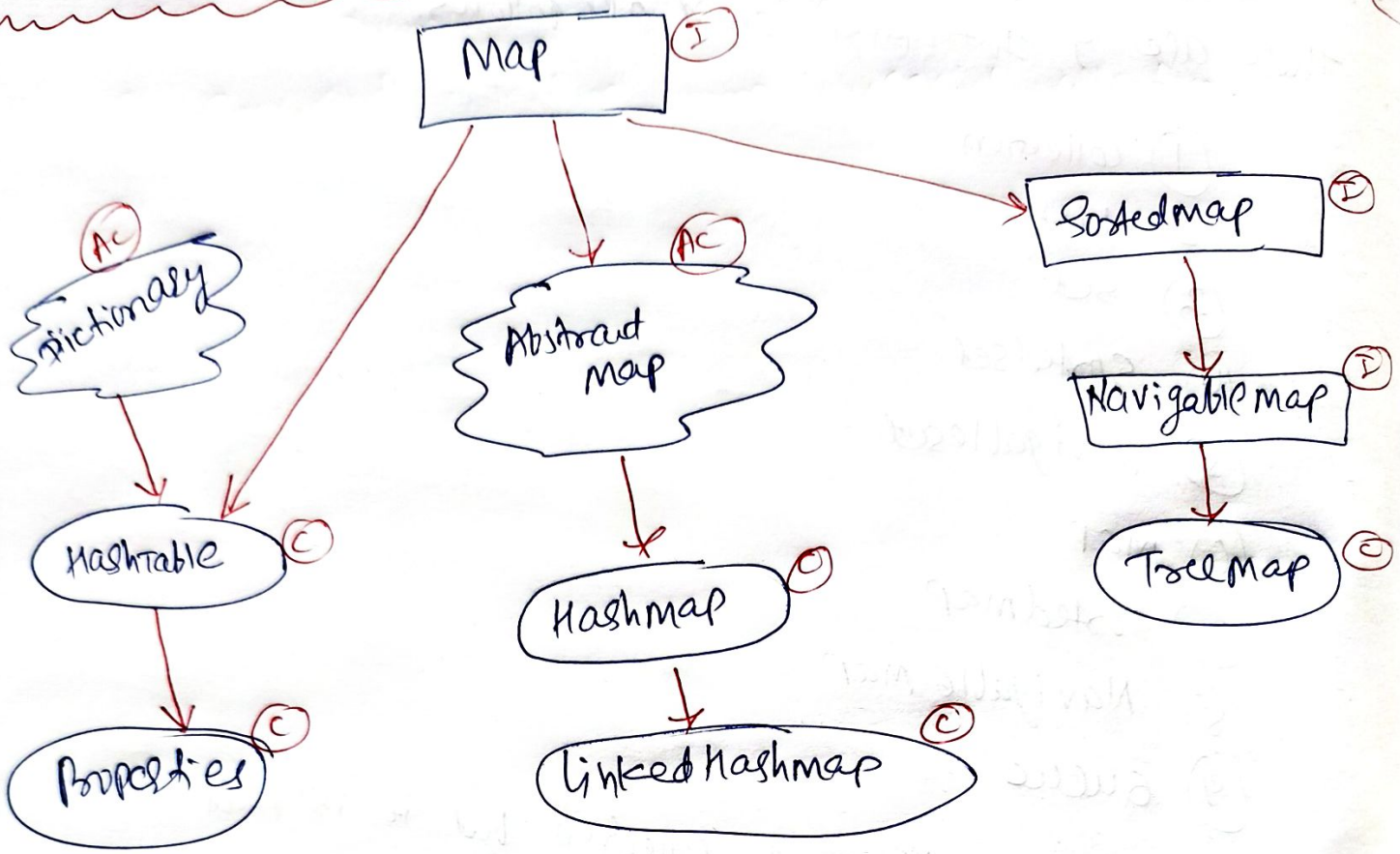
1. collection
2. List
3. Set
4. Sortedset
5. Navigableset
6. Map
7. Sortedmap
8. Navigable map
9. Queue

- map is not child interface of collection but it is part of collection framework.
- collection interface has following 3 sub interfaces

1. List
2. Set
3. Queue

- vector, stack, Dictionary, Hashtable and properties classes are legacy classes and which came in Jdk 1.0 version.
- Queue concept came in 1.5
- Navigableset and Navigable Map Interface came in 1.6 version of JDk.
- LinkedHashset, LinkedHashmap and IdentityHashmap came in Jdk 1.4

Map Interface



Collection Interface methods

1. boolean add (Object obj)

   Adds objects to collection

2. int size ()

   returns no of objects available in collection.

3. boolean isEmpty ()

   checks whether collection is empty or not

(4) boolean contains (object obj)

checks whether collection has specified object or not.

(5) Iterator iterator()

Returns Iterator subclass object

(6) Object[] toArray()

returns an array containing all of the elements of collection.

(7) boolean remove (object obj)

removes first occurance of specified object from collection.

(8) void clear()

removes all of the elements from collection.

(9) boolean addAll (Collection c)

Adds All the data of specified collection in the current collection.

(10) boolean containsAll (Collection c)

returns true if current collection all the elements of the specified collection.

(11) boolean removeAll (Collection c)

Removes all the matching elements of specified collection from current collection.

(12) boolean retainAll(Collection c)

removes all the non matching elements of specified collection from current collection.