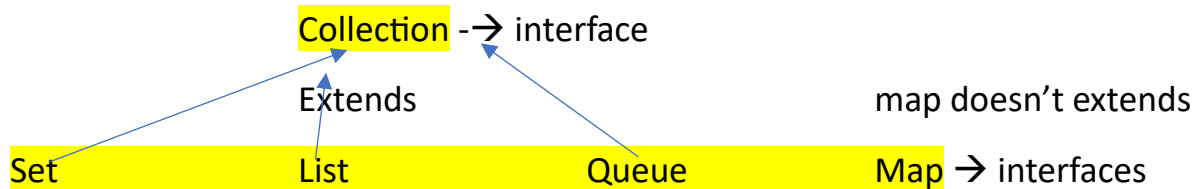


Util package.

Collection Framework hierarchy



Set : it doesn't allow duplicate data. Under set implementation classes may be unordered, ordered or sorted. Set doesn't provide index.

Set implementation classes

1. **HashSet** : HashSet is a type of set class which help to store any type of values. HashSet stored in the element in unordered manner.
2. **LinkedHashSet**: LinkedHashSet is a type of set class which internally extends HashSet class. LinkedHashSet maintain the order.
3. **TreeSet**: TreeSet is a type of Set api. Which internally implements SortedSet interface and that interface extends Set interface. SortedSet interface provided logic or algorithms to do sorting by default in Ascending order. In TreeSet we need to store same type of data types values. TreeSet provided few extra method like headset, tailset, subset etc.

List : it allow duplicate. It maintains the order and provide index features to get the value.

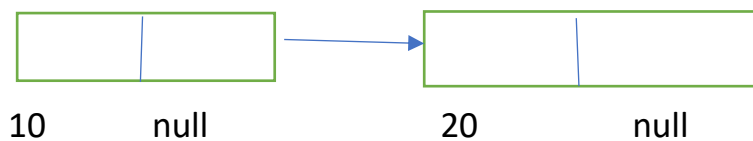
List implementation classes

1. **ArrayList**
ArrayList Vs Normal Array
 - a. ArrayList dynamic memory. Normal array fixed memory size.
 - b. ArrayList provided lot of pre defined methods help to add, remove, search etc method But normal array doesn't provide any predefined method.

c. ArrayList allow to store any type of data types values by nature. But normal array same data types values

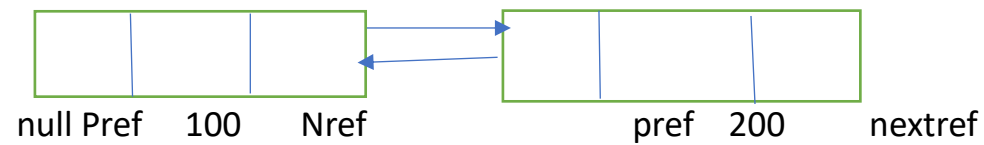
2. LinkedList : LinkedList is a type of List API in Java. LinkedList internally use Node concept to store the data. The node mainly divided into two or three parts depending upon type of LinkedList

- a. Singular linked list
- b. Double linked list
- c. Circular linked (singular or double)



Value ref of another node

->



null

Null-Prev value nexRef -Null

<-----

->

In Java by default LinkedList consider as double linked list.

Arraylist is good option when we retrieve the element from array list again and again.

LinkedList is good option when we do more insertion and deletion operation.

3. Vector

Vector : Vector is known as legacy class. In Vector all methods by default synchronized. It is a thread safe but slow performance.

4. Stack

Stack type of data structure which provide features as first in last out or last in first out.

Stack class internally extends Vector.

Push → to add the data

Pop → to remove the data

Peek → to check top most data

Search → to search data present in stack or not.

Queue : Queue is a type of data structure. Queue provide First In First Out. FIFO.

Queue implementation classes

1. PriorityQueue

Base upon priority ie lower value by default

First in First out

2. LinkedList : First in first out no priority.

Set, List and Queue we can store one-one information of any type.

Map : it allow to store the information in key-value pairs.

Key can be unique and value may be duplicate.

Map implementation classes

1. **HashMap** : unordered . it can allow null key as well as null value.
But only one null key and more than one null value.
2. **LinkedHashMap** : order. . it can allow null key as well as null value.
But only one null key and more than one null value.
3. **TreeMap** : asc order as key so key must be same data types
TreeMap can't allow null key. But value can.
4. **Hashtable** : by default synchronized, legacy class.
It doesn't allow null key as well as null value.

Retrieve the value from collection framework one by one

1. For each loop
2. **Iterator** : iterator is an interface which provide set of methods

Which help to retrieve the element one by one.

Only forward direction

3. **ListIterator** : it is a type of interface which internally extends Iterator.
It provide features to move forward as well as backward direction.
4. **Enumeration** : it is a type of legacy iterator. It use with vector.

1 one is type of loop and 2,3 and 4 type of interfaces.

Collection framework with generic

If we store any type data types values in collection storing become easily

Retrieve become complex.

```
Collection<Type> objectName = new CollectionName<Type>();
```

Type can be Integer, Float, String, Double, or User Defined Object.

