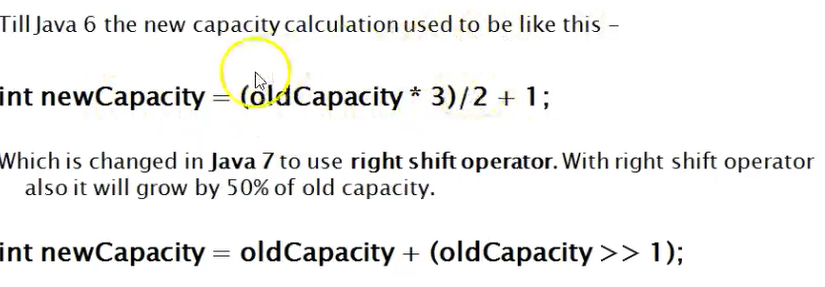


Array List : size increase by current size\*3/2+1

Next one will be 16 (10\*3/2+1),25(16\*3/2+1) if 11th element will come, all the old element will be copy in the new created array and 6 are empty now. old array will go for garbage collector.



Default size is 10.

Null value can be inserted. // Duplicate are allowed.

# Insertion operation will take more time because of shifting the element +1 position. Arraylist is bed choice .

# for deletion operation Arraylist is bed choice .

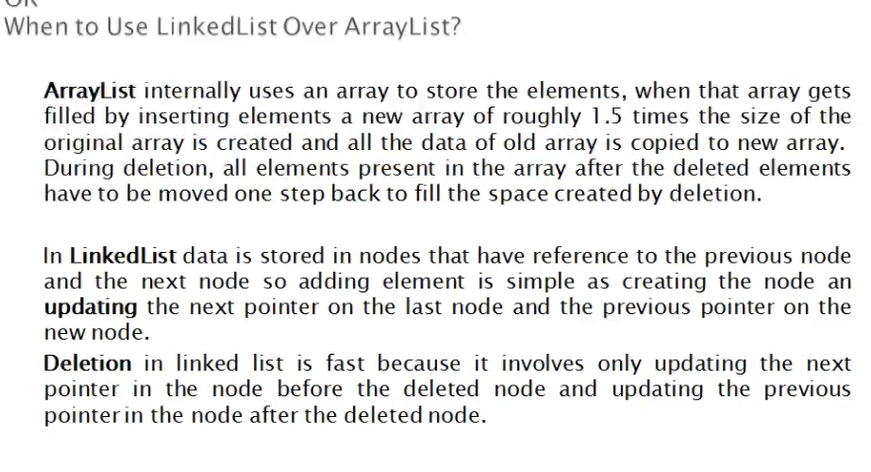
# list.get(0) // For retrival operation Arraylist are good choice, it implemets random interface. Vector class is also implement random interface.

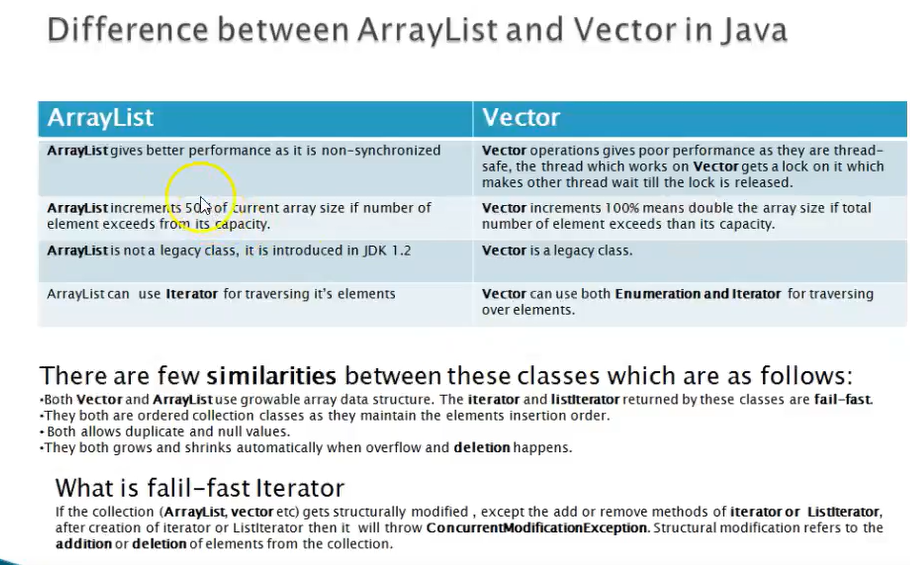
Vector : increment size by doubled.

Most of the methods are synchronized.

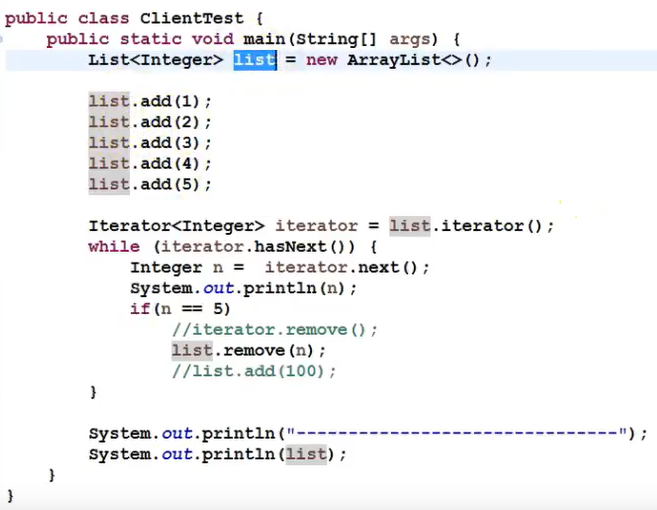
**Linked List :**





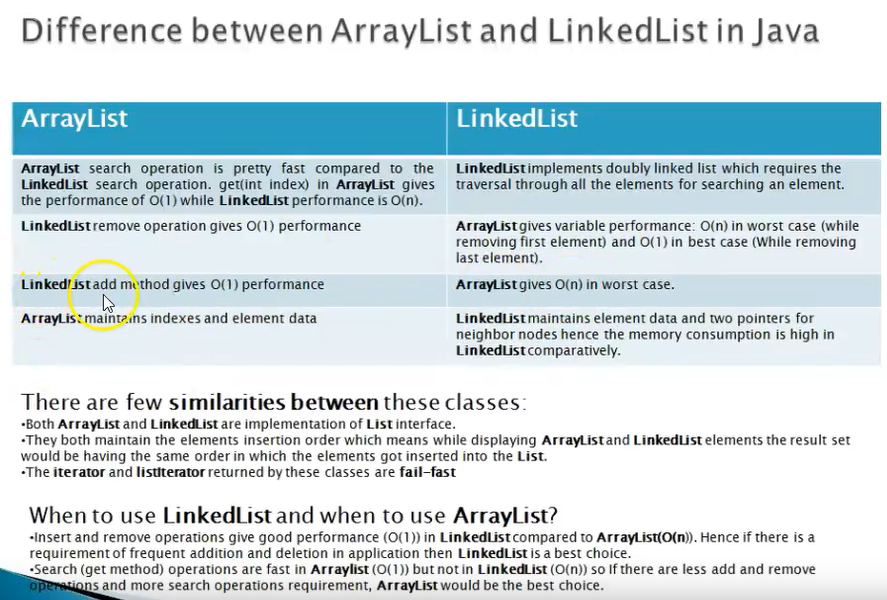


Example of fail-fast iterator :

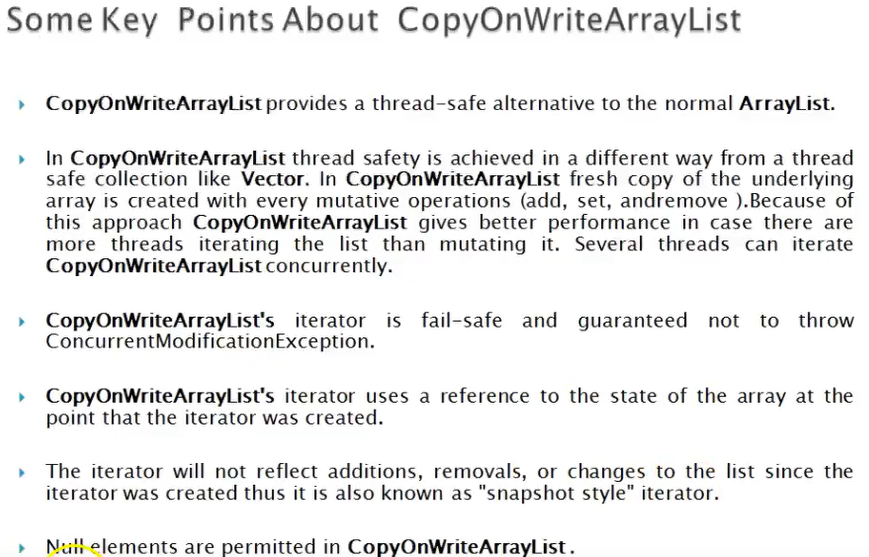


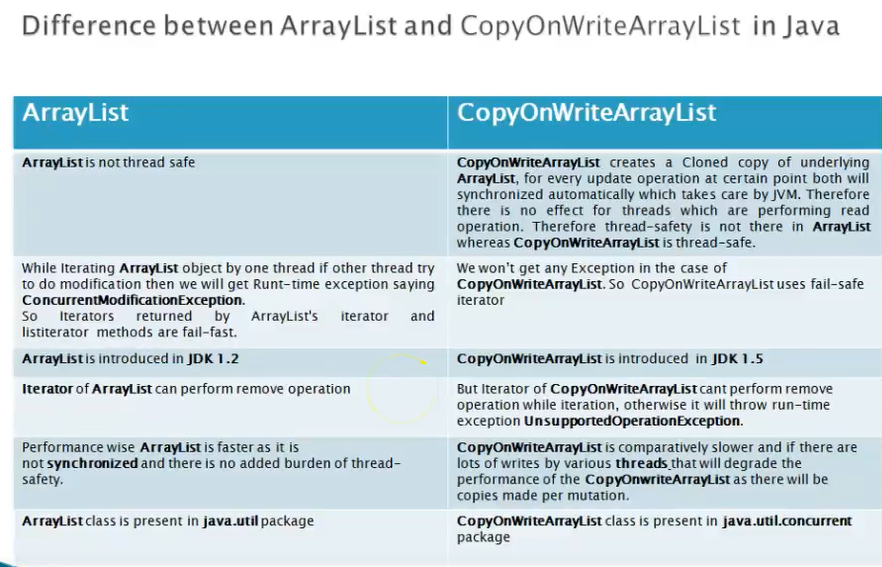
// if we do iterator.remove then we don’t have any exception. But

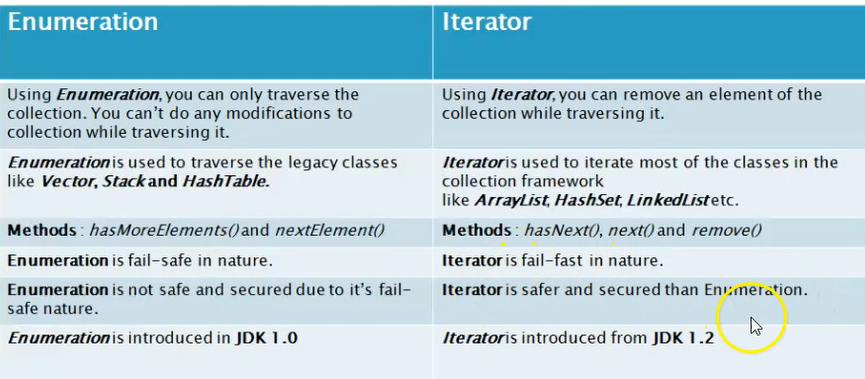
We use list.remove(), here we are changing the structure of the list. here it will throw concurrent modification exception.

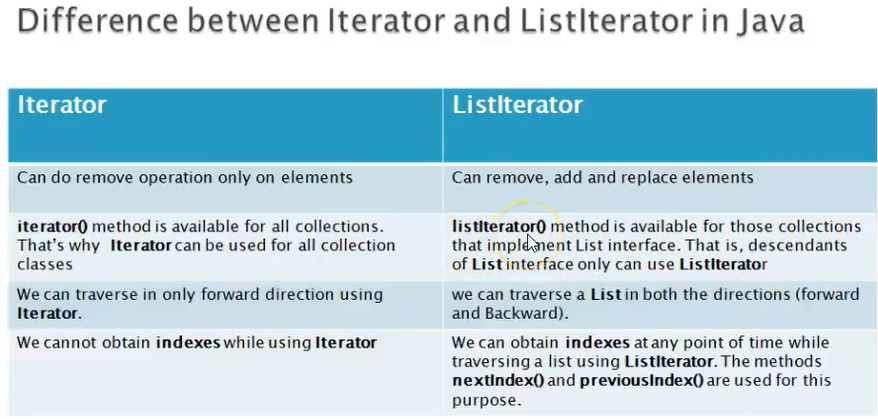


**CopyOnWriteArrayList** :









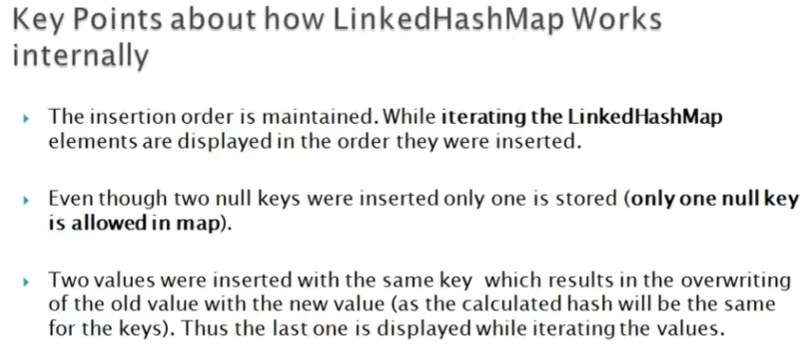


Load factor of the HashMap is 0.75 means if size of the map is 75 % then it will be increased with another 16 bucket HashMap.

Threshold is 12 it means if 12 buckets is full then the map size will be doubled.

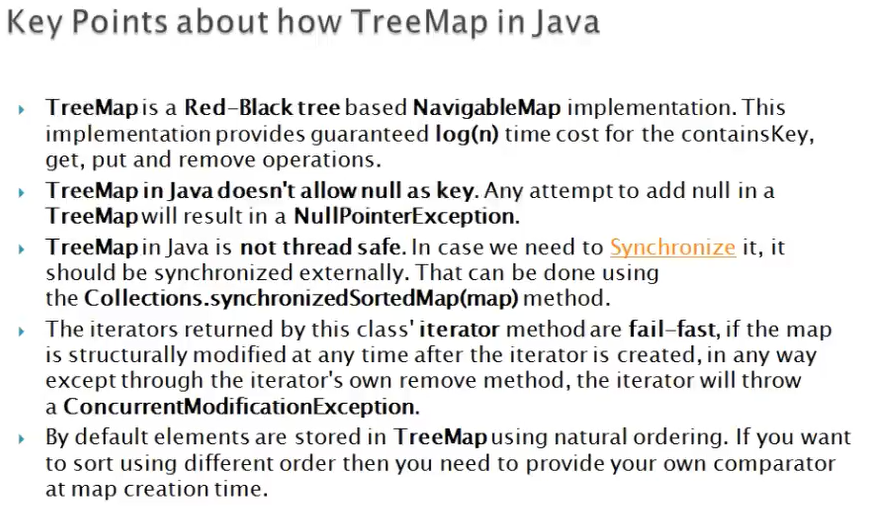
HashSet: if we add any element s.add(“one”), it calls the equals method to check element is exist or not.

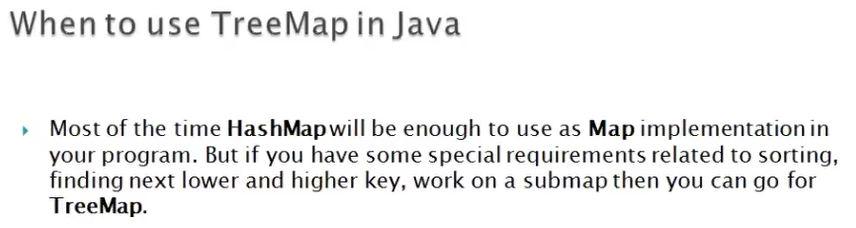
‘One’ is stored as key and object is stored as value.

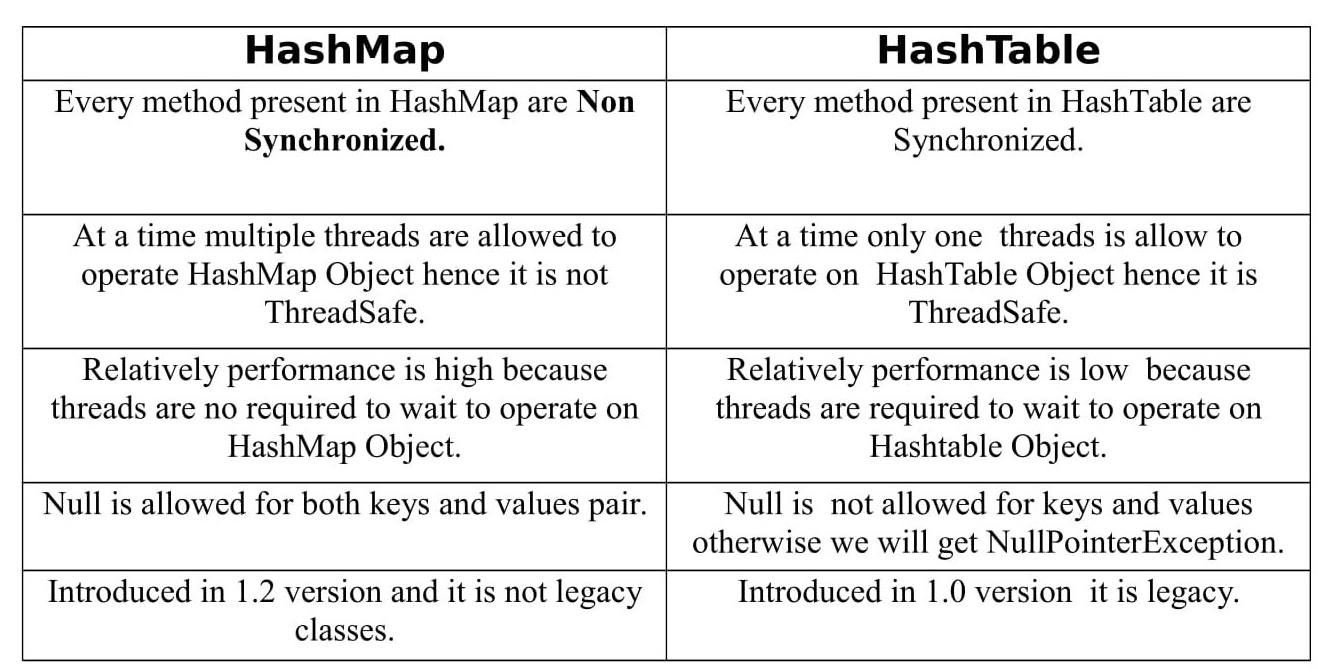


If we have two null keys in the linked list, it will accept one null key with the overridden value.

For example: lhm1.put (null, deparment1); lhm1.put (null, department2) it will display the 2nd department.







**Thread:**

