SET (I)

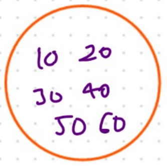
Implementation classes

(1) HashSet

(2) Linked HashSet

(3) Tree set

SET : If you want to represent group of individual objects in a single entity where duplicates are not allowed, then we can go with SET.



Hashtable

hashvalue	data
fa51b	10
fa2la	11
≡	≡

"10" $\longrightarrow$ hashvalue – fa51b

"11" $\longrightarrow$ hashvalue – fa62d

"10" $\longrightarrow$ "fa51b"

HashSet (C)

properties :

(1) It follows hashing alg (efficiently for searching) internally

(2) the internal data structure is hash table

(3) default capacity is 16

(4) load factor : 0.75 (when 75% of the set is filled it create a new set) // fill ratio

(5) new capacity = old capacity * 2

(6) duplicates are not allowed

(7) insertion order is not maintained

(8) Null insertion is allowed (only once)

(9) majorly used for searching

Constructor

new HashSet() //16

new HashSet (collection c)

new HashSet (int initial Capacity)

new HashSet (int initial Capacity, float load factor)

0.75 //default

0.95

0.91

$\uparrow$

Hash Set

(1) Insertion order is mainted

(2) Underlying data structure is hash table (retrieval)

Linked Hash Set

(1) Insertion order is maintained

(2) Underlying data structure is Linked List (insertion & deletion)

TreeSet

Constructor

new TreeSet()

new TreeSet(Collection C)

new TreeSet(SortedSet S)

new TreeSet(Comparator C)

methods() [present in SortedSet(I) i-e inherited from it]

object first() //10

object last() //120

79 end (e)

SortedSet headSet(Object O) //10,50

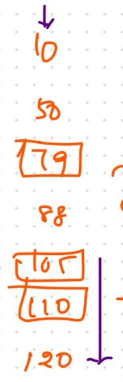
88

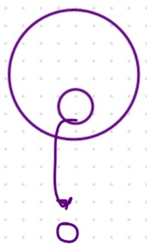
SortedSet tailSet(Object O) //105 110 120

SortedSet Subset(object O_1, Object O_2) // 79, 88, 105

inclusive exclusive

Subset(79, 106) // 79, 88, 105





Comparator comparator()

Properties ()

(1) duplicates are not allowed "abc", 10, True

(2) Insertion order is not present

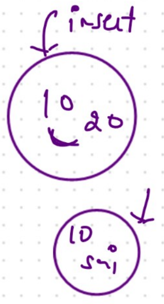
(3) Heterogenous object are not allowed

/ If we tries to add diff type of data we will get

[ClassCastException]

(4) Null insertion is not allowed, if we tries to add we get

[NullpointerException]



(5) elements are by default in Sorted order

* numbers on wrapper class follows ascending order
* strings it follows dictionary / alphabetical order

A, B . . .

aa
ab
ac

ba

(6) It accepts only homogeneous & comparable object

(7) Objects are said to be comparable if the class implements Comparable interface

i.e we can find integer class that implements Comparable Interface

i.e all the wrapper classes & String class have implemented Comparable interface