Friday, November 19, 2021 5:35 PM

Lists --- add(index,value) , value = get(index) } Index based API

Set interface ---- **No** index based APIs , No duplicates !!!

Set interface **extends** from Collection interface

Class TreeSet **implemets** Set
Class HashSet implements Set

TreeSet ---------------- internally it will use the binary tree data structure.

When we add any element to the TreeSet object then that goes in the binary tree
34, 23,78,1,65

```
            34
      23          78
1              65
```

When we traverse the TreeSet using Iterator, Enumeration or for(:) ----
The elements are read in INORDER (inorder traversal L Root Right)
1,23,34,65,78 } yielding us sorted list

We want to store our data in sorted order always
The elements that we want to store in the TreeSet must be Comparable OR they must have a Comparator

---

HashSet --- NO index based access , no duplicates

The elements are stored by HASHING !!

Collection ===== red , blue, green , turquoise

```
hashcode   hashcode_function(element)
 {
        //Return length(element )   // weak criteria

        Return 1  //HORRIBLE
 }
```

Hashcode	Bucket
1	
2	
3	red
4	blue

5	Green
6	
7	
8	
9	Turquoise
10	
11	

Advantage of Hashing --- **FASTEST search** technique
Search if **grey** is in the list
For array worst case number of comparisons O(n)
For Hash worst case number of comparisons O(1)
For Binary tree worst case number of comparison O(log n)

Java. Lang. Object class ---------
Equals, toString , hashcode

hashcode returns a unique value

Equals HASHCODE Contract !!!

if equals is true for obj1 and obj2
then hash code for obj1 and obj2 must be SAME !!!

By default hashcode method in the Object class returns the hashcode
based on the address of the object!!!

Whenever we OVERRIDE equals WE should override hashcode !!!

List , Set }}} One element is one Value

Map Interface =
Each element is a PAIR !! (Key,Value)

List has 5 elements
1 add (45)
10
23
55
9

Map has 5 elements
1,"nikhil"
10,"shubham"
23, "siddhesh"
55, "vedant"
9,"atharva"

Map interface DOES NOT extend from COLLECTION interface

API

Put method is used for adding element to MAP

put (key , value)

Get method is used to GET the value if the key is given

value = get (key)

Value = get(55)

Sysout (value) ---- vedant

value = get (100)

Sysout (value) ------- null

Map interface ---- put , get , keySet , values

TreeMap , HashMap , HashTable , Dictionary , Properties } Subclasses of Map interface

TreeMap ----- Binary tree is created

One Pair is stored in ONE node

Key must be Comparable !!!

HashMap ------- pair is stored in the bucket

Key must be having equals and hashcode overriden

Hashcode is created on Key

Find out difference between HashMap , HashTable ?

HW ----- run the codes done in class !!