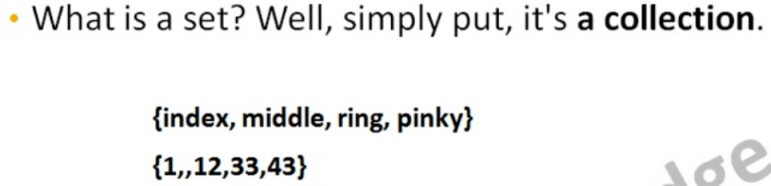
**Sets and Maps**

Sets : collection bag of any elements



- set don’t allow duplicate

- the set collection auto sort the elements in-order(First letter alphabetical/numerical)

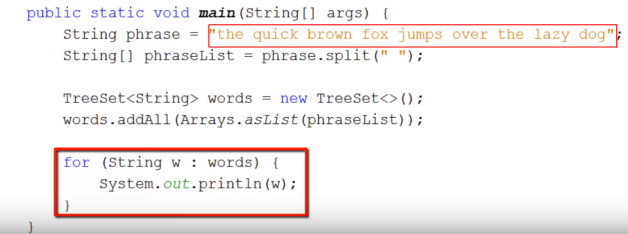
- not the same category not available for compare

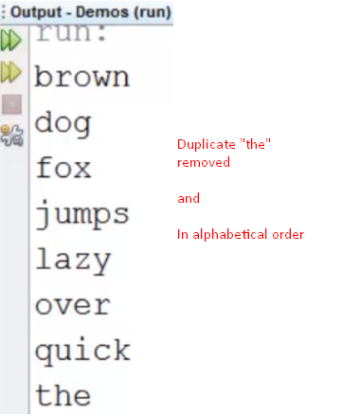
- Set and map collections store and retrieve data by value rather than by position.

- For Set and map insertion and retrieval time complexity = O(1)

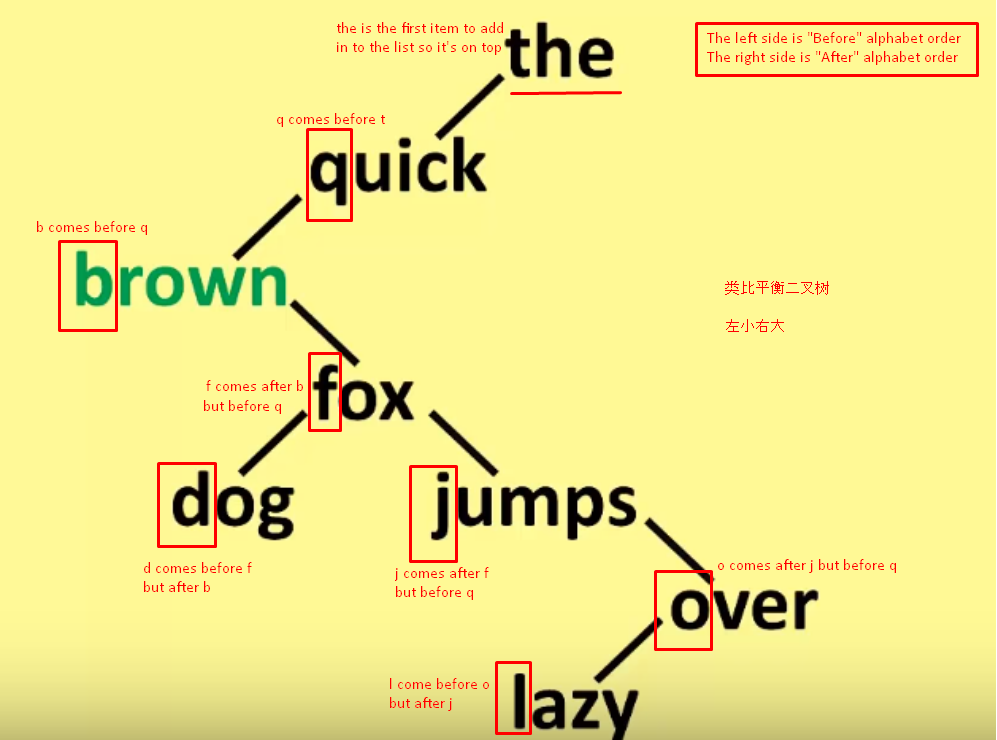
- For set and map , .put() for insert new element , .add() method is for arraylist.

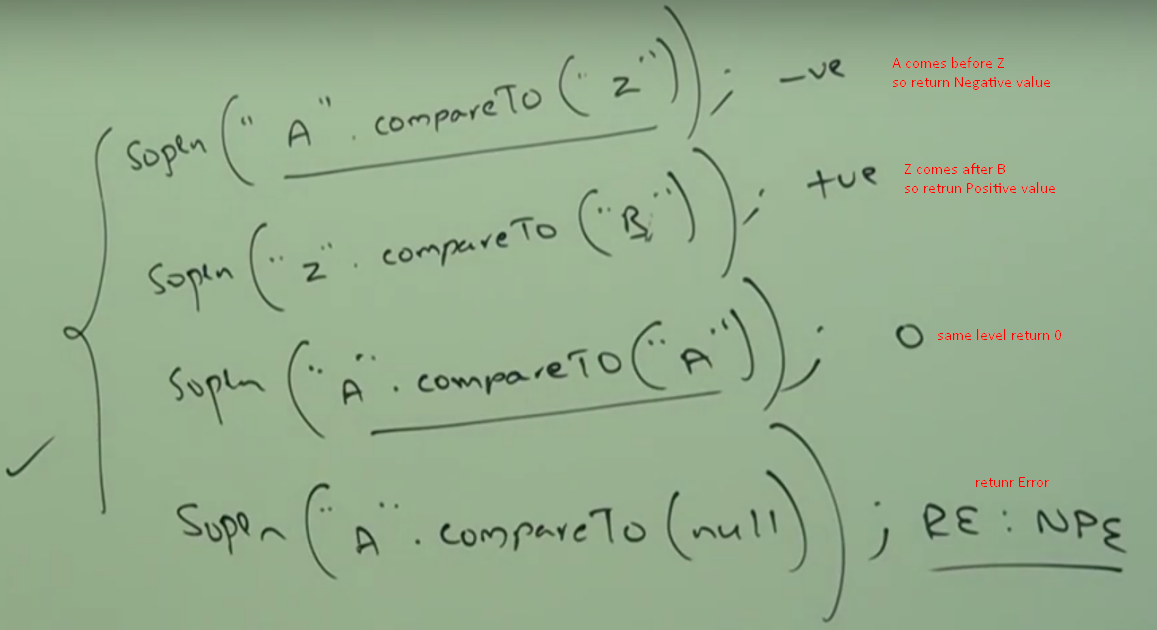
例:





How tree set order works ?

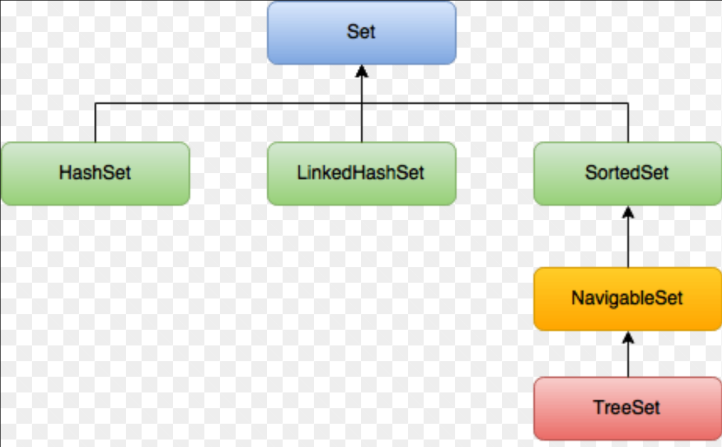




用函数instanceof(),add()加数学集合逻辑

可构造set operators (Union,Intersection,Difference,Subset..)

**HashSet 和 LinkedHashSet 和 TreeSet 的区别**



Input :

myset.add(31);

myset.add(21);

myset.add(43);

myset.add(4);

myset.add(45);

myset.add(6);

myset.add(57);

myset.add(86);

for(int i: myset)

{

System.out.println(i);

}

- HashSet Output : //Not in order

4 , 21 , 6 , 86 , 57 , 43 , 45 , 31

- LinkedHashSet Output : //Same as input order

31 , 21 , 43 , 4 , 45 , 6 , 57 , 86

- TreeSet Output : //Sorted order

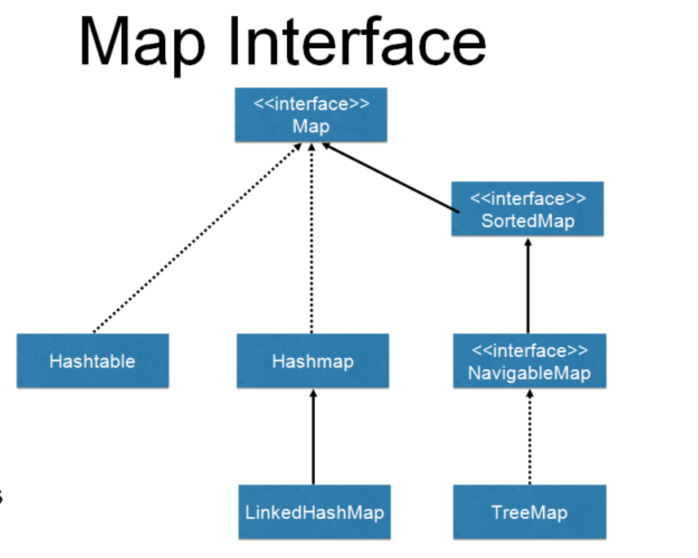
4 , 6 , 21 , 31 , 43 , 45 , 57 , 86

**Maps : comes with key value pairs**



**Hashtable(注意t小写) , HashMap 和TreeMap**

hashtable is thread safe , hashmap is not thread safe



Input :

phonebook.put("y", "1");

phonebook.put("x", "2");

phonebook.put("q", "3");

phonebook.put("p", "4");

phonebook.put("o", "5"); //.put( ) for set/map , .add( ) for arraylist

Set<Map.Entry<String, String>> values= phonebook.entrySet();

for(Map.Entry<String,String>e:values) //Entry is a In

{

System.out.println(e.getKey()+":"+e.getValue());

}

- Hashtable Output: //Unodered

x:2 , q:3 , p:4 , o:5 , y:1

- HashMap Output: //Unordered

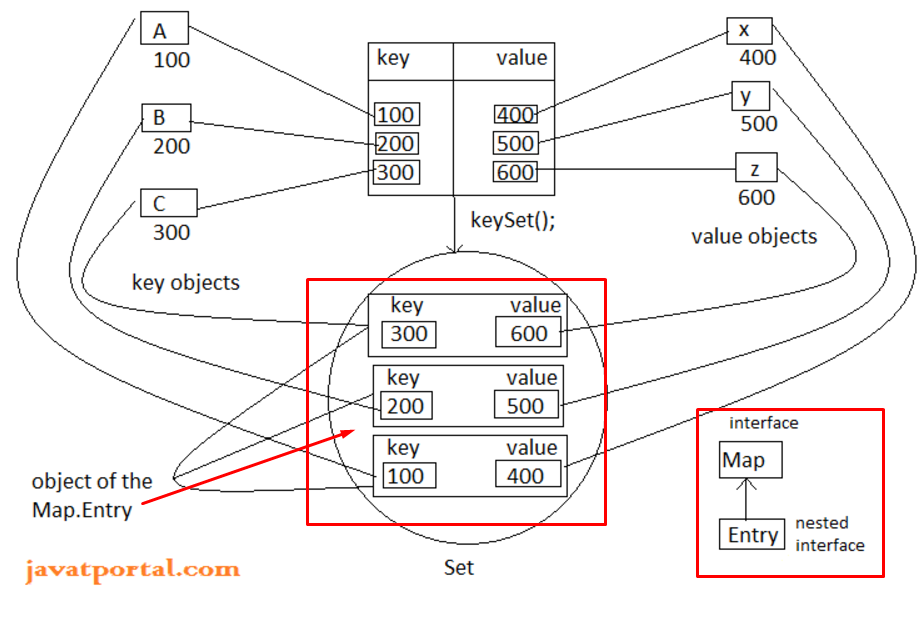
p:4 , q:3 , x:2 , y:1 , o:5

-TreeMap Output: //Ordered

o:5 , p:4 , q:3 , x:2 , y:1

**Map.Entry Interface**

Entry interface of map interface helps to provide getKey() getValue() setValue() method.



P591

**Entry Set Iterator**

跟for loop差不多，输出结果排序也遵循上述规律

Set setofKeys=phonebook.keySet();

java.util.Iterator iterator=setofKeys.iterator();

while(iterator.hasNext()) {

String key=(String) iterator.next();

String value=phonebook.get(key);

System.out.println("Key:"+key+",Value:"+value);

**modCount()方法**

modCount = modified count 更改次数计算

modCount的作用是迭代器在遍历时做线程安全检查的

**Anonymous Inner class**

- class with no name

- Used when implemented interfaces (because interface include abstract stuff which cannot be instantiated)

- Implemented inside a method

