**map(I):**

It contains unique elements.

map contains value based on the key.

K<Integer> v<String>

101 ganesh

102 ramesh

103 suresh

**1: HashTable(IC)**

Hashtable doesn't allow any null key or value.

HashTable does not maintain order while iterating.

It works on the principle of hashing.

It is legacy class, synchronized, low performance

**2: HashMap(IC)**

It may have a single null key and multiple null values.

HashMap does not maintain order while iterating.

It works on the principle of hashing.

It is not legacy class, non-Synchronized, high performance

**3: LinkedHashMap(IC)**

It may have a single null key and multiple null values.

\*HashMap maintain order while iterating.

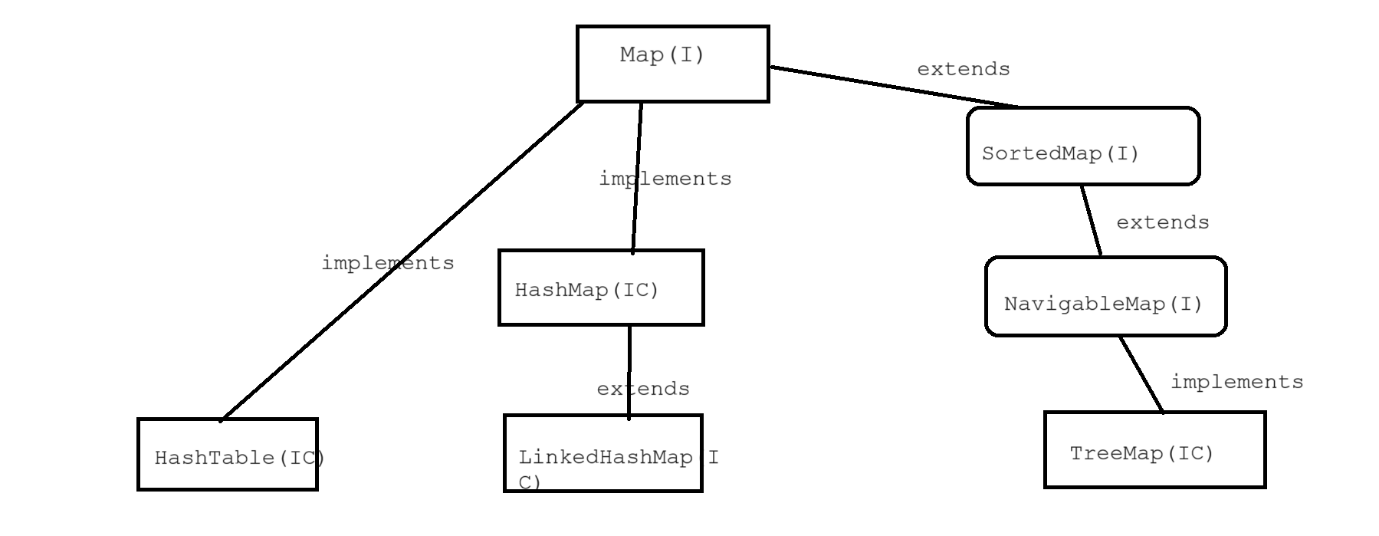
\*It works on the principle of hashing.

**4: TreeMap(IC)**

It cannot have a null key but have multiple null values.

\*TreeMap is sorted by keys in ascending order.

\*It stores the object in the tree structure.



package Map;

import java.util.HashMap;

import java.util.Set;

public class Map\_ex1\_HashMap

{

public static void main(String[] args)

{

HashMap<Integer , String> hm=new HashMap<>();

hm.put(104, "ganesh");

hm.put(55, "suresh");

hm.put(101, "mahesh");

hm.put(71, "rahul");

//update value of spacefic key

hm.put(71, "RAHUL");

System.out.println(hm);

System.out.println(hm.size());

System.out.println(hm.containsKey(104)); //true

System.out.println(hm.get(101));

System.out.println("-----------");

//get all keys

Set<Integer> allKeys = hm.keySet(); //{104, 55, 101, 71}

for(Integer key:allKeys)

{

System.out.println(key);

}

System.out.println("-----");

//get all keys & Values

for(Integer key:allKeys)

{

System.out.println(key+" : "+hm.get(key));

}

}

}

package Map;

import java.util.HashMap;

import java.util.Hashtable;

public class Map\_ex2\_DiffBetween\_HashMap\_And\_HashTable

{

public static void main(String[] args)

{

// Hashtable ht=new Hashtable<>();

// ht.put("ramesh", 1);

// ht.put("mahesh", null);

// ht.put("ganesh", null);

// ht.put(null, 1);

//

// System.out.println(ht);

HashMap mp=new HashMap<>();

mp.put("ramesh", 1);

mp.put("mahesh", null);

mp.put("ganesh", null);

mp.put(null, 1);

System.out.println(mp);

}

}

package Map;

import java.util.HashMap;

import java.util.Hashtable;

public class Map\_ex2\_DiffBetween\_HashMap\_And\_HashTable

{

public static void main(String[] args)

{

// Hashtable ht=new Hashtable<>();

// ht.put("ramesh", 1);

// ht.put("mahesh", null);

// ht.put("ganesh", null);

// ht.put(null, 1);

//

// System.out.println(ht);

HashMap mp=new HashMap<>();

mp.put("ramesh", 1);

mp.put("mahesh", null);

mp.put("ganesh", null);

mp.put(null, 1);

System.out.println(mp);

}

}

**package** Map;

**import** java.util.TreeMap;

**public** **class** Map\_ex4\_TreeMap

{

**public** **static** **void** main(String[] args)

{

TreeMap<Integer, String> hm=**new** TreeMap<>();

hm.put(101, "ganesh");

hm.put(200, "Ramesh");

hm.put(300, "suresh");

hm.put(40, "ramesh");

System.***out***.println(hm);

}

}