Java Hashtable class

- Java Hashtable class implements a hashtable, which maps keys to values. It inherits Dictionary class and implements the Map interface.

Points to remember

- A Hashtable is an array of a list. Each list is known as a bucket. The position of the bucket is identified by calling the hashcode() method. A Hashtable contains values based on the key.
- Java Hashtable class contains unique elements.
- Java Hashtable class doesn't allow null key or value.
- Java Hashtable class is synchronized.
- The initial default capacity of Hashtable class is 11.

Methods of HashTable

```
import java.util.HashMap;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;

public class Hash3
{
    public static void main(String[] args)
    {
        Hashtable<Integer, String> ht1 = new Hashtable<>();
        Hashtable<Integer, String> ht2 = new Hashtable<>();

        //(1)//
        /*public synchronized V put(K key, V value)*/

        ht1.put(10, "JJS");
        ht1.put(15, "HDS");
        ht1.put(20, "CVM");
        ht1.put(30, "PAT");
        ht1.put(40, "DJU");

        System.out.println("hm1 is: " + ht1);
        //hm1 is: {10=JJS, 20=CVM, 30=PAT, 40=DJU, 15=HDS}

        //(2)//
        /* public Collection<V> values() */

        System.out.println(ht1.values()); // Provides Collections
        //[JJS, CVM, PAT, DJU, HDS]
        //(3)//
        /*public Set<K> keySet()*/

        System.out.println(ht1.keySet()); // Provides Set
        //[10, 20, 30, 40, 15]

        //(4)//
        /*  public synchronized void putAll(Map<? extends K, ? extends V> t) */

        ht2.putAll(ht1);

        System.out.println("hm2 is: " + ht2);
        //hm2 is: {10=JJS, 20=CVM, 30=PAT, 40=DJU, 15=HDS}
/*_________________________________________________________________________*/

        //(5)//
        /* public synchronized V remove(Object key) */

         // remove(key) method returns its value and removes available key and
            value from the hashmap.
        // remove(key) method returns null if key is not found

        System.out.println("Removed element is: " + ht1.remove(100));
        //Removed element is: null
        System.out.println("Removed element is: " + ht1.remove(10));
        //Removed element is: JJS
        System.out.println(ht1);
        // {20=CVM, 30=PAT, 40=DJU, 15=HDS}

        //(6)//
        /* public synchronized boolean remove(Object key, Object value) */

        // remove(key,value) method returns boolean (false) and removes pair if
        // available.

        System.out.println("Removed element is: " + ht1.remove(10, "CVM"));
        //Removed element is: false
        System.out.println("Removed element is: " + ht1.remove(30, "PAT"));
        //Removed element is: true

        System.out.println(ht1);
        //{20=CVM, 40=DJU, 15=HDS}

        //(6)//
        /*public synchronized void clear() */
        //Clears this hashtable so that it contains no keys.
        ht2.clear();

        System.out.println("Updated hm2 is: " + ht2);
        //Updated hm2 is: {}

/*_________________________________________________________________ */
        //(7)//
        /*public synchronized boolean containsKey(Object key) */

        //return true if and only if the specified object is a key in this
        //hashtable false otherwise.


        System.out.println(ht1.containsKey(20));//true
        System.out.println(ht1.containsKey(30)); //false



        //(8)//
        /*public boolean containsValue(Object value) */
        //Returns true if this hashtable maps one or more keys to this value.

        System.out.println(ht1.containsValue("HDS"));//true
        System.out.println(ht1.containsValue("ARP"));//false

/*_________________________________________________________________________ */
        // How to print HashTable
        //(1)//

        Set set=ht1.entrySet();
        System.out.println(set);
        //[20=CVM, 40=DJU, 15=HDS]
        Iterator itr=set.iterator();
        while(itr.hasNext())
        {
            Map.Entry entry=(Map.Entry) itr.next();
            System.out.println(entry.getKey()+"="+entry.getValue());
        }
         //(2)//

        for (Map.Entry m : ht1.entrySet())
        {
            System.out.println(m.getKey()+"="+m.getValue());
        }
        /* 20=CVM
           40=DJU
           15=HDS*/
    }
}
```