❖ Java HashSet

- Java HashSet class is used to create a collection that uses a hash table for storage. It inherits the AbstractSet class and implements Set interface.

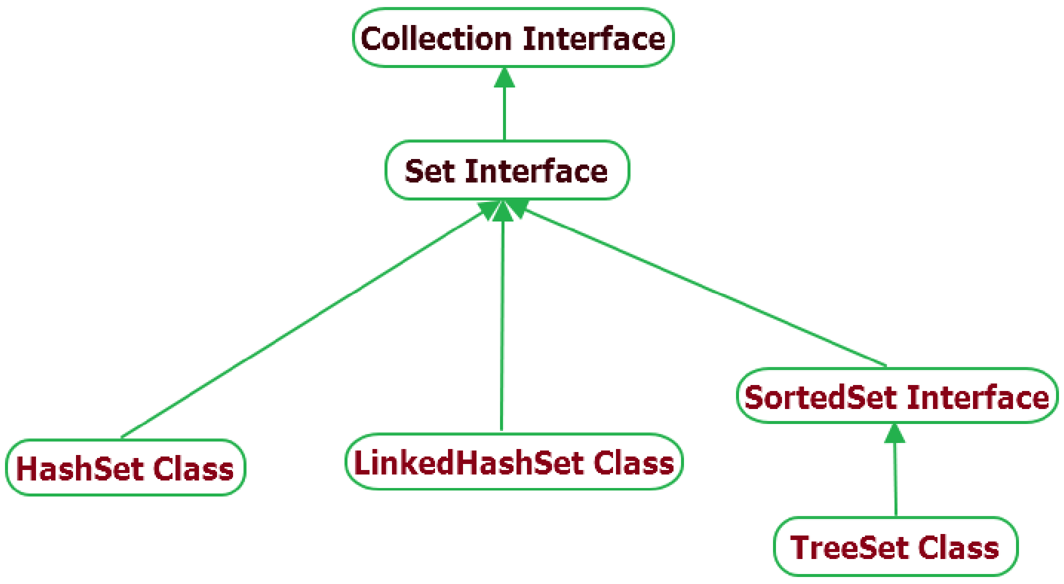
❖ The important points about Java HashSet class are:

- HashSet stores the elements by using a mechanism called hashing.
- HashSet doesn't maintain the insertion order. Here, elements are inserted on the basis of their hashcode.
- HashSet contains unique elements only.
- HashSet allows null value.
- HashSet class is non synchronized.
- HashSet is the best approach for search operations.
- The initial default capacity of HashSet is 16.

❖ Difference between List and Set

- A list can contain duplicate elements whereas Set contains unique elements only.

❖ Hierarchy of HashSet class



❖ Constructors of Java HashSet class

HashSet()

- It is used to construct a default HashSet.

HashSet(int capacity)

- It is used to initialize the capacity of the hash set to the given integer value capacity. The capacity grows automatically as elements are added to the HashSet.

❖ Methods of HasSet

```
import java.util.HashSet;
import java.util.Iterator;

public class Hash1
{
   public static void main(String[] args)
   {
      HashSet<Integer> hs1 = new HashSet<>();

   //*1*//public int size()

   /*Returns number of elements in the set */

      System.out.println(hs1.size());//0
/*______________________________________________________________*/

   //*2*//public boolean add(E e)

   /*Adds the specified element to this set if it is not already present  and returns true.If this
     set already contains the element, the call leaves the set unchanged and returns  false.*/

      System.out.println(hs1.add(8));//true
      hs1.add(1);
      hs1.add(2);
      hs1.add(3);
      hs1.add(7);
      hs1.add(4);
      hs1.add(5);
      hs1.add(6);
      hs1.add(7);
      System.out.println(hs1);//[1, 2, 3, 4, 5, 6, 7, 8]

      System.out.println(hs1.size());//8


   /*Duplicates are not allowed */

      System.out.println(hs1.add(5));//false
      System.out.println(hs1);//[1, 2, 3, 4, 5, 6, 7, 8]

/*______________________________________________________________*/
```

//*3*//public boolean remove(Object o)

/*Removes the specified element from set if it is present and returns true If this set does't contains the element, the call leaves the set unchangedand returns false.*/

```
hs1.remove(5);
System.out.println(hs1);//[1, 2, 3, 4, 6, 7, 8]

System.out.println(hs1.remove(4));//true
System.out.println(hs1);//[1, 2, 3, 6, 7, 8]

System.out.println(hs1.remove(9));//false
System.out.println(hs1);//[1, 2, 3, 6, 7, 8]
```

/*____*/

//*4*//public boolean isEmpty()

/*Returns true if the set is empty and false if set contains elements */

```
System.out.println(hs1.isEmpty());//false

HashSet<Integer> hs2 = new HashSet<>();

System.out.println(hs2.isEmpty());//true
```

/*____*/

//*5*//public boolean addAll(Collection<? extends E> c)

```
System.out.println(hs2.addAll(hs1));//true
System.out.println(hs2);//[1, 2, 3, 6, 7, 8]

HashSet<Integer> hs3 = new HashSet<>();
hs3.add(2);
hs3.add(9);
System.out.println(hs3.addAll(hs2));//true
System.out.println(hs3);//[1, 2, 3, 6, 7, 8, 9]
```

/*____*/

//*6*//public boolean removeAll(Collection<?> c)

/*removes all the elements from a set that are contained in a collection*/

```
System.out.println(hs3.removeAll(hs2));//true
System.out.println(hs3);//[9]
```

/*____*/

```
    //*7*// public void clear()

    /*Removes all of the elements from this set.*/

        hs3.clear();
        System.out.println(hs3);//[]
/*____________________________________________________________________*/

    //*8*// public boolean equals(Object o)

    /* returns true if the specified object is equal to this set else false */

        System.out.println("set hs1:"+hs1);//set hs1:[1, 2, 3, 6, 7, 8]
        System.out.println("set hs2:"+hs2);//set hs2:[1, 2, 3, 6, 7, 8]
        System.out.println("set hs3:"+hs3);//set hs3:[]

        System.out.println(hs1.equals(hs2));//true
        System.out.println(hs1.equals(hs3));//false


        System.out.println(hs1.hashCode());//27
        System.out.println(hs2.hashCode());//27
        System.out.println(hs3.hashCode());//0
/*____________________________________________________________________*/

   //*printing elements*//
        Iterator itr = hs1.iterator();

        while (itr.hasNext())
        {
            System.out.println(itr.next());

        }
    }
}
```