LinkedHashSet

LinkedHashSet in Java is a concrete class that implements set interface and extends HashSet class with a doubly linked list implementation.

It internally uses a linked list to store the elements in the set. It was added in Java 1.4 version.

Java LinkedHashSet class is the same as HashSet class, except that it maintains the ordering of elements in the set in which they are inserted.

That is, LinkedHashSet not only uses a hash table for storing elements but also maintains a double-linked list of the elements in the order during iteration.

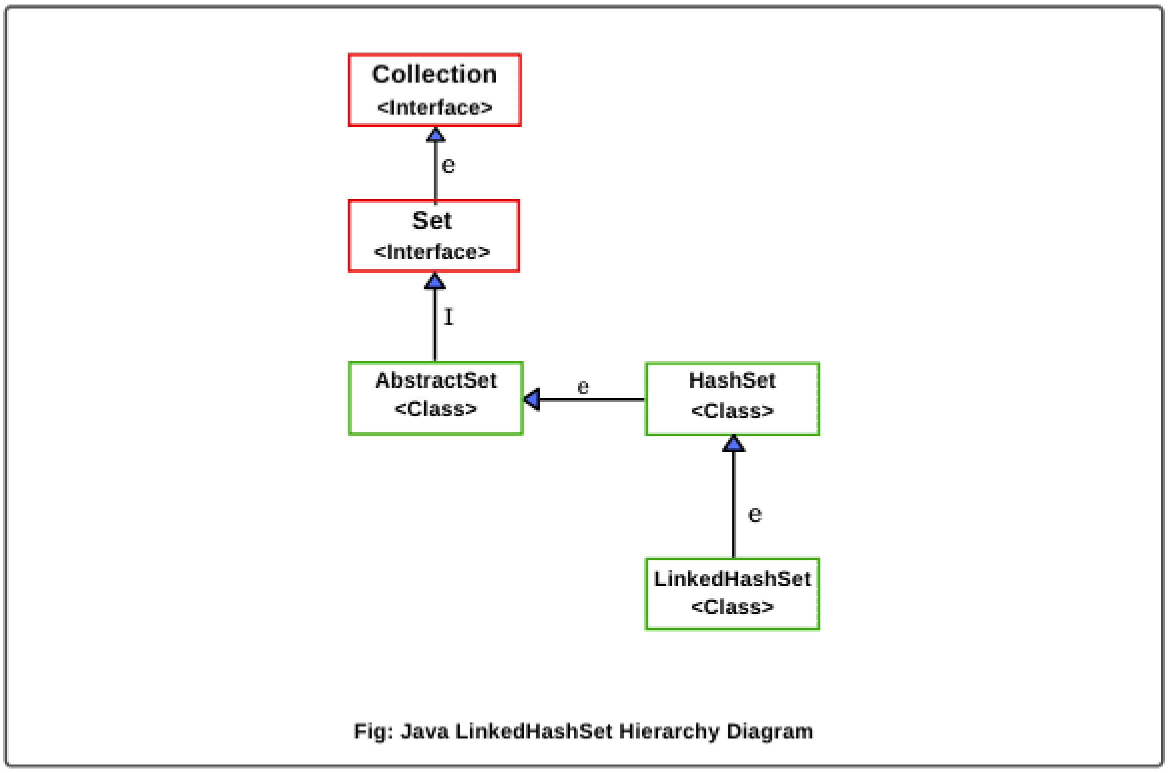

Fig: Java LinkedHashSet Hierarchy Diagram

```
import java.util.LinkedHashSet;
public class AddTest {
public static void main(String[] args)
{
// Create a Linked hash set of generic type.
   LinkedHashSet<String> lhset= new LinkedHashSet<String>();

// Checking the size of LinkedHashSet before adding elements.
   int size = lhset.size();
   System.out.println("Size of LinkedHashSet before adding elements: " +size);
// Adding elements in the linked hash set.
   lhset.add("Red"); // lhset.size() is 1.
   lhset.add("Green"); // lhset.size() is 2.
   lhset.add("Yellow"); // lhset.size() is 3.
   lhset.add("Blue"); // lhset.size() is 4.
   lhset.add("Orange"); // lhset.size() is 5.

   System.out.println("Elements in Set: " +lhset);
   int size2 = lhset.size();
   System.out.println("Size of LinkedHashSet after adding elements: " +size2);

// Adding duplicate elements that already exist in set.
   lhset.add("Red"); // lhset.size() is still 5.
   lhset.add("Yellow"); // lhset.size() is still 5.

// Create another set of String type.
   LinkedHashSet<String> lhset2 = new LinkedHashSet<String>();
   lhset2.add("Brown");
   lhset2.add(null);

// Adding elements of set2 into set.
   lhset.addAll(lhset2);
   System.out.println("Elements in Set after adding: " +lhset);
 }
}
```

```
Output:
      Size of LinkedHashSet before adding elements: 0
      Elements in Set: [Red, Green, Yellow, Blue, Orange]
      Size of LinkedHashSet after adding elements: 5
      Elements in Set after adding: [Red, Green, Yellow, Blue, Orange, Brown, null]
```

```
import java.util.LinkedHashSet;
public class RemoveDemo
{
public static void main(String[] args)
{
// Create a Linked hash set of generic type.
   LinkedHashSet<String> set= new LinkedHashSet<String>();

// Adding elements in the linked hash set.
   set.add("A");
   set.add("G");
   set.add("Y");
   set.add("B");
   set.add("O");
   set.add(null);

   System.out.println("Elements in set: " +set);

// Remove a string element from linked hash set.
   set.remove(null);
   System.out.println("Elements in set after removing: " +set);
   System.out.println(set.size()+ " elements in set");

// Create another linked hash set of String type.
   LinkedHashSet<String> set2 = new LinkedHashSet<String>();
   set2.add("S");
   set2.add(null);
   System.out.println("Elements in set2: " +set2);
   System.out.println(set2.size()+ " elements in set2");

   System.out.println("Is S in set2? " +set2.contains("S"));

   set.addAll(set2);
   System.out.println("Elements in set after adding: " +set);

   set.removeAll(set2);
   System.out.println("Elements in set after removing set2: " +set);

   set.retainAll(set2);
   System.out.println("After removing common elements in set2 " + "from set, set is "
 + set);
  }
}
```

```
Output:
      Elements in set after removing: [A, G, Y, B, O]
      5 elements in set
      Elements in set2: [S, null]
      2 elements in set2
      Is S in set2? true
      Elements in set after adding: [A, G, Y, B, O, S, null]
      Elements in set after removing set2: [A, G, Y, B, O]
      After removing common elements in set2 from set, set is []
```

```
import java.util.ArrayList;
import java.util.LinkedHashSet;
public class RemovingDuplicate {
public static void main(String[] args)
{
  int[] num = {20, 30, 50, 30, 40, 80, 10, 10};
  ArrayList<Integer> ar = new ArrayList<Integer>();

// Adding numbers to the array list.
   for(int i = 0; i < num.length; i++)
   {
      ar.add(num[i]);
   }
  System.out.println("Original list: " +ar);
  LinkedHashSet<Integer> lhset = new LinkedHashSet<>(ar);
  System.out.println("New list after removing duplicate numbers: " +lhset);
 }
}
```

```
Output:
       Original list: [20, 30, 50, 30, 40, 80, 10, 10]
       New list after removing duplicate numbers: [20, 30, 50, 40, 80, 10]
```

```
import java.util.Iterator;
import java.util.LinkedHashSet;
public class IteratingLinkedHashSet {
public static void main(String[] args)
{
  LinkedHashSet<String> lhset = new LinkedHashSet<>();
  lhset.add("New York");
  lhset.add("Dhanbad");
  lhset.add("Sydney");
  lhset.add("Cape Town");
  lhset.add("London");

// Iterating elements of LinkedHashSet using iterator() method.
   System.out.println("Iteration using iterator");
   Iterator<String> itr = lhset.iterator();
   while(itr.hasNext())
   {
      System.out.println(itr.next());
   }
   System.out.println();

// Iterating elements of LinkedHashSet using enhanced for loop
   System.out.println("Iteration using enhanced for loop");
   for (String s : lhset)
     System.out.print(s + " ");
     System.out.println();
  }
}
```

```
Output:
      Iteration using iterator
      New York
      Dhanbad
      Sydney
      Cape Town
      London

      Iteration using enhanced for loop
      New York Dhanbad Sydney Cape Town London
```

```
public class Student {
String name;
int id;
String city;
Student(String name, int id, String city){
 this.name = name;
 this.id = id;
 this.city = city;
  }
}
import java.util.LinkedHashSet;
public class StudentInfo {
public static void main(String[] args)
{
LinkedHashSet<Student> lhset = new LinkedHashSet<Student>();

// Creating objects of Students.
   Student st1 = new Student("John", 2345, "New York");
   Student st2 = new Student("Deep", 1234, "Dhanbad");
   Student st3 = new Student("Ricky", 7583, "Cape Town");

// Adding elements (object references) into LinkedHashSet.
   lhset.add(st1);
   lhset.add(st2);
   lhset.add(st3);

// Traversing linked hash set.
  for(Student s:lhset){
      System.out.println("Name: " +s.name+" "+ "Id: " +s.id+" "+"City: "+s.city);
   }
  }
}
```

```
Output:
      Name: John Id: 2345 City: New York
      Name: Deep Id: 1234 City: Dhanbad
      Name: Ricky Id: 7583 City: Cape Town
```