

深入Java集合学习系列：LinkedHashSet的实现原理

博客分类：Core Java

原文地址：<http://zhangshixi.iteye.com/blog/673319>

Java

C

C++

C#

1. LinkedHashSet概述：

LinkedHashSet是具有可预知迭代顺序的Set接口的哈希表和链接列表实现。此实现与HashSet的不同之处在于，后者维护着一个运行于所有条目的双重链接列表。此链接列表定义了迭代顺序，该迭代顺序可为插入顺序或是访问顺序。

注意，此实现不是同步的。如果多个线程同时访问链接的哈希Set，而其中至少一个线程修改了该Set，则它必须保持外部同步。

2. LinkedHashSet的实现：

对于LinkedHashSet而言，它继承与HashSet、又基于LinkedHashMap来实现的。

LinkedHashSet底层使用LinkedHashMap来保存所有元素，它继承与HashSet，其所有的方法操作上又与HashSet相同，因此LinkedHashSet的实现上非常简单，只提供了四个构造方法，并通过传递一个标识参数，调用父类的构造器，底层构造一个LinkedHashMap来实现，在相关操作上与父类HashSet的操作相同，直接调用父类HashSet的方法即可。LinkedHashSet的源代码如下：

Java代码



```
1. public class LinkedHashSet<E>
2.     extends HashSet<E>
3.     implements Set<E>, Cloneable, java.io.Serializable {
4.
5.     private static final long serialVersionUID = -2851667679971038690L;
6.
7.     /**
8.      * 构造一个带有指定初始容量和加载因子的新空链接哈希set。
9.      *
10.     * 底层会调用父类的构造方法，构造一个有指定初始容量和加载因子的LinkedHashMap实例。
11.     * @param initialCapacity 初始容量。
12.     * @param loadFactor 加载因子。
13.     */
14.     public LinkedHashSet(int initialCapacity, float loadFactor) {
15.         super(initialCapacity, loadFactor, true);
```

```

16.     }
17.
18.     /**
19.      * 构造一个带指定初始容量和默认加载因子0.75的新空链接哈希set。
20.      *
21.      * 底层会调用父类的构造方法，构造一个带指定初始容量和默认加载因子0.75的LinkedHashMap实例。
22.      * @param initialCapacity 初始容量。
23.      */
24.     public LinkedHashSet(int initialCapacity) {
25.         super(initialCapacity, .75f, true);
26.     }
27.
28.     /**
29.      * 构造一个带默认初始容量16和加载因子0.75的新空链接哈希set。
30.      *
31.      * 底层会调用父类的构造方法，构造一个带默认初始容量16和加载因子0.75的LinkedHashMap实例。
32.      */
33.     public LinkedHashSet() {
34.         super(16, .75f, true);
35.     }
36.
37.     /**
38.      * 构造一个与指定collection中的元素相同的新链接哈希set。
39.      *
40.      * 底层会调用父类的构造方法，构造一个足以包含指定collection
41.      * 中所有元素的初始容量和加载因子为0.75的LinkedHashMap实例。
42.      * @param c 其中的元素将存放在此set中的collection。
43.      */
44.     public LinkedHashSet(Collection<? extends E> c) {
45.         super(Math.max(2*c.size(), 11), .75f, true);
46.         addAll(c);
47.     }
48. }

```

在父类HashSet中，专为LinkedHashSet提供的构造方法如下，该方法为包访问权限，并未对外公开。

Java代码  

```

1.     /**
2.      * 以指定的initialCapacity和loadFactor构造一个新的空链接哈希集合。
3.      * 此构造函数为包访问权限，不对外公开，实际只是对LinkedHashSet的支持。
4.      *
5.      * 实际底层会以指定的参数构造一个空LinkedHashMap实例来实现。

```

```
6.      * @param initialCapacity 初始容量。  
7.      * @param loadFactor 加载因子。  
8.      * @param dummy 标记。  
9.      */  
10.     HashSet(int initialCapacity, float loadFactor, boolean dummy) {  
11.         map = new LinkedHashMap<E, Object>(initialCapacity, loadFactor);  
12.     }
```

由上述源代码可见，LinkedHashSet通过继承HashSet，底层使用LinkedHashMap，以很简单明了的方式来实现了其自身的所有功能。

3. 相关说明：

- 1) 相关HashSet的实现原理，请参考我的上一遍总结：[深入Java集合学习系列：HashSet的实现原理](#)。
- 2) 相关HashMap的实现原理，请参考我的上一遍总结：[深入Java集合学习系列：HashMap的实现原理](#)。