

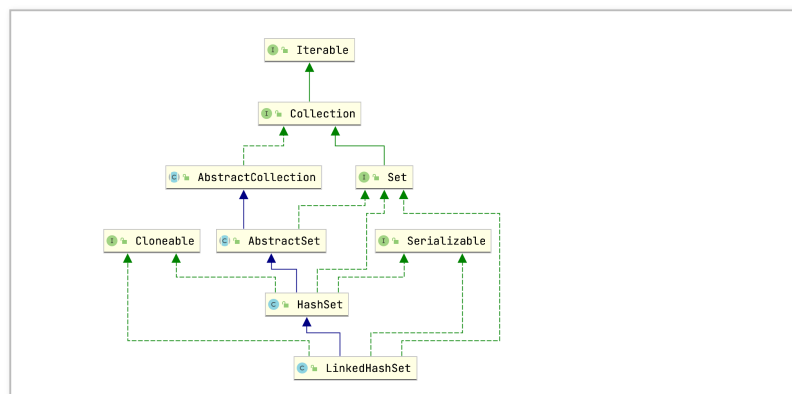
# LinkedHashSet 源码解析 | MrBird

“ 我们知道，HashSet 内部使用 HashMap 存储元素，所以 HashSet 遍历数据时是无序的，要保证插入的元素有序，我们可以使用 LinkedHashSet。

2020-09-03 | Visit count 1058059

我们知道，HashSet 内部使用 HashMap 存储元素，所以 HashSet 遍历数据时是无序的，要保证插入的元素有序，我们可以使用 LinkedHashSet。本节记录 LinkedHashSet 源码解析，基于 JDK1.8。

LinkedHashSet 类层级关系如下所示：



LinkedHashSet 继承自 HashSet。查看 LinkedHashSet 的构造方法源码会发现内部都是调用父类的 `HashSet(int initialCapacity, float loadFactor, boolean dummy)` 方法：

```
HashSet(int initialCapacity, float loadFactor, boolean dummy) {
    map = new LinkedHashMap<>(initialCapacity, loadFactor, true);
}
```

dummy 参数没有任何意义，仅用于和别的入参为 (int,float) 的构造器区分开。方法内部创建的是 LinkedHashMap，所以 LinkedHashSet 就是用 LinkedHashMap 来保证插入元素有序的，对 LinkedHashMap 不熟悉的请参考 [LinkedHashMap 源码解析](#)。

从上面的代码我们还可以发现，LinkedHashSet 无法改变 linkedHashMap 的 accessOrder 属性值，所以在 LinkedHashSet 中，元素的顺序只能和插入顺序一致：

```
LinkedHashSet<String> set = new LinkedHashSet<>();
set.add("apple");
set.add("orange");
set.add("watermelon");
set.add("strawberry");
System.out.println(set);
```

输出顺序和插入顺序一致：

```
[apple, orange, watermelon, strawberry]
```

---

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 <sup>beta</sup>，[点击查看详细说明](#)

