



链滴

Java 集合之 LinkedHashMap

作者: [hiccup234](#)

原文链接: <https://ld246.com/article/1546961836566>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

简介

LinkedHashMap是继承自HashMap的有序实现，和HashMap同样都是非线程安全的，多线程环境修改容易出现意想不到的问题。

LinkedHashMap使用相关疑问

疑问

是否允许存null

是否允许重复数据

存储是否有序

是否线程安全

结论

key和value都可以为空

key重复会覆盖，value可以重复

有序

非线程安全

既然LinkedHashMap继承自HashMap那他们内部有哪些区别呢？

LinkedHashMap中额外定义了三个属性

```
/**
 * The head (eldest) of the doubly linked list.
 * 双向链表的头部
 */
transient LinkedHashMap.Entry<K,V> head;

/**
 * The tail (youngest) of the doubly linked list.
 * 双向链表的尾部
 */
transient LinkedHashMap.Entry<K,V> tail;

/**
 * The iteration ordering method for this linked hash map: true
 * for access-order, false for insertion-order. * * @serial
 * true 表示最近最少使用次序规则，false表示插入顺序规则，默认初始化为false;
 */
final boolean accessOrder;
```

LinkedHashMap.Entry 继承自 HashMap.Node类并增加了两个属性

```
static class Entry<K,V> extends HashMap.Node<K,V> {
    //增加了两个属性均是为了维护元素的顺序，每个元素中都存储了前后元素的位置，注意不要和 next
    属性搞混了！！
    //next是用于维护HashMap指定table位置上连接的Entry的顺序的
    Entry<K,V> before, after;
    Entry(int hash, K key, V value, Node<K,V> next) {
        super(hash, key, value, next);
    }
}
```

LinkedHashMap的元素存储

翻阅 LinkedHashMap 源码并没有发现重写HashMap的 put 方法，而是重写了put方法里调用的几个子方法 如： 'newNode(...)' ， 'replacementNode(...)' ， 'newTreeNode(...)' 等等，基于多态的用，主要功能就是维护顺序，其它实现均和HashMap相同。

注：本篇文档是基于 [JDK1.8](#) 的源码所编写，原理和旧版JDK类似，只是实现稍有不同，主要区别还于JDK1.8后的HashMap实现。