

HashSet 源码解析 | MrBird

“ 本文记录 HashSet 源码解析，基于 JDK1.8。

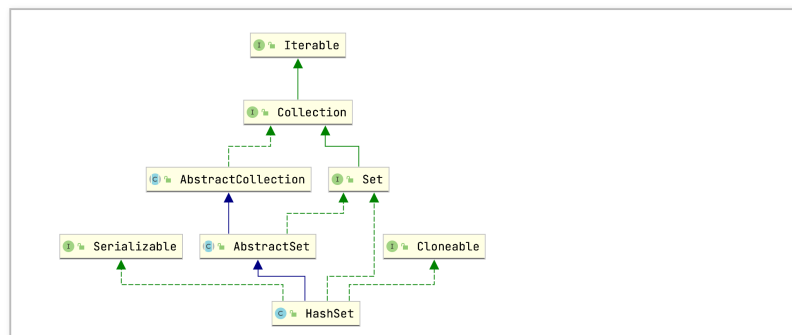
HashSet 源码解析

2020-08-25 | Visit count 1058055

本文记录 HashSet 源码解析，基于 JDK1.8。

类结构

HashSet 类层级关系图：



HashSet 实现了 Set 接口，为什么叫 HashSet？因为 HashSet 内部采用哈希表（实际就是 HashMap）来存储不重复的数据，查看 HashSet 内部属性：

```
// 使用HashMap存储数据，HashSet的数据实际为HashMap的key
private transient HashMap<E,Object> map;
// HashMap value占位符
private static final Object PRESENT = new Object();
```

HashMap 的 key 是不允许重复的，这也正好符合 Set 的特性。因为 HashSet 内部采用 HashMap 存储数据，所以 HashSet 可以存储 null 值，支持快速失败，非线程安全。

map 属性通过 transient 修饰，原因在介绍 [HashMap 源码](#) 的时候分析过。

方法解析

构造函数

```
// 空参构造函数，内部初始化map属性
public HashSet() {
    map = new HashMap<>();
}

// 传入集合对象
public HashSet(Collection<? extends E> c) {
    // 初始化map，计算map的容量
    // 计算公式为 c.size/0.75f + 1，如果值小于16，则取值16
    map = new HashMap<>(Math.max((int) (c.size()/.75f) + 1, 16));
    // 将集合中的所有元素添加进去
    addAll(c);
}

// 手动指定容量和加载因子
public HashSet(int initialCapacity, float loadFactor) {
    map = new HashMap<>(initialCapacity, loadFactor);
}

// 手动指定容量
public HashSet(int initialCapacity) {
    map = new HashMap<>(initialCapacity);
}
```

可以看到，创建 HashSet 的本质就是初始化 HashMap。

add(E e)

add(E e) 添加指定元素：

```
public boolean add(E e) {
    // 往map里添加元素，如果key已经存在则返回false，否则返回true
    return map.put(e, PRESENT) != null;
}
```

contains(Object o)

contains(Object o) 判断是否包含指定元素：

```
public boolean contains(Object o) {  
    // 本质就是判断map中是否包含该key  
    return map.containsKey(o);  
}
```

size()

size() 获取元素个数:

```
public int size() {  
    // 本质就是获取map的元素个数  
    return map.size();  
}
```

isEmpty()

isEmpty() 判断集合是否为空:

```
public boolean isEmpty() {  
    // 本质就是判断map是否为空  
    return map.isEmpty();  
}
```

remove(Object o)

remove(Object o) 删除指定元素:

```
public boolean remove(Object o) {  
    // 本质就是通过key删除map中的元素，如果该key存在，则返  
    return map.remove(o)==PRESENT;  
}
```

clear()

clear() 清空集合:

```
public void clear() {  
    // 本质就是清空map  
    map.clear();  
}
```

iterator()

`iterator()` 获取迭代器:

```
public Iterator<E> iterator() {  
    // 本质就是获取map key的迭代器  
    return map.keySet().iterator();  
}
```

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^{beta}, 点击查看详细说明

