

CopyOnWriteArraySet 源码解析 | MrBird

“ CopyOnWriteArraySet 为线程安全的 Set 实现，本文记录 CopyOnWriteArraySet 源码解析，基于 JDK1.8。

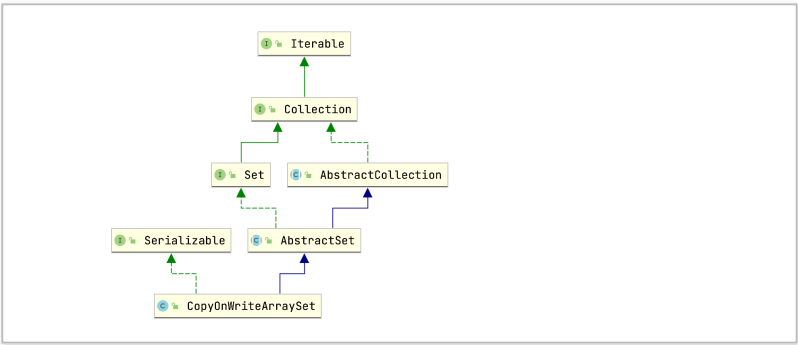
CopyOnWriteArraySet 源码解析

2020-08-26 | Visit count 1058056

CopyOnWriteArraySet 为线程安全的 Set 实现，本文记录 CopyOnWriteArraySet 源码解析，基于 JDK1.8。

类结构

先来看下 CopyOnWriteArraySet 的类层级关系图：



没什么好说的。再来看看内部属性：

```
// 就一个属性，CopyOnWriteArraySet内部采用CopyOnWriteArr
private final CopyOnWriteArrayList<E> al;
```

和 HashSet 不一样的是，CopyOnWriteArraySet 内部采用 CopyOnWriteArrayList 存储元素，这也是 CopyOnWriteArraySet 名字的由来，因为 CopyOnWriteArrayList 是线程安全的，CopyOnWriteArraySet 的方法都是基于 CopyOnWriteArrayList 实现的，所以 CopyOnWriteArraySet 自然而然也是线程安全的，同样的，在并发环境下获取数据是弱一致性的！

方法解析

构造函数

```
// 空参构造函数，实际就是初始化CopyOnWriteArrayList
public CopyOnWriteArraySet() {
    al = new CopyOnWriteArrayList<E>();
}

// 传入集合对象
public CopyOnWriteArraySet(Collection<? extends E> c) {
    // 分两种情况
    if (c.getClass() == CopyOnWriteArraySet.class) {
        // 如果集合就是CopyOnWriteArraySet类型，说明数据
        // 直接全部添加到CopyOnWriteArrayList中
        @SuppressWarnings("unchecked") CopyOnWriteArraySet<E> cc = (CopyOnWriteArraySet<E>)c;
        al = new CopyOnWriteArrayList<E>(cc.al);
    }
    else {
        // 否则调用addAllAbsent添加所有当前集合中不存在的
        al = new CopyOnWriteArrayList<E>();
        al.addAllAbsent(c);
    }
}
```

add(E e)

add(E e) 添加指定元素：

```
public boolean add(E e) {
    // 实际调用CopyOnWriteArrayList的addIfAbsent方法
    // 元素不存在，则添加，返回true；元素存在，则不添加
    return al.addIfAbsent(e);
}
```

可以看到，CopyOnWriteArraySet 的 add 方法通过调用 CopyOnWriteArrayList 的 addIfAbsent 来确保元素不重复，以满足 Set 的特性。

剩下方法略

剩下方法都比较简单，都是直接调用 CopyOnWriteArrayList 方法实现，感兴趣自己阅读源码。

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](#)

