## 常用集合类源码解析

源码解析JDK1.8：ArrayList、LinkedList、HashSet、TreeSet、HashMap、TreeMap、ConcurrentHashMap

#### 一、ArrayList源码解析

List有索引、有序、可重复。

查询快，新增删除慢。

**1. 实现接口：**

List、RandomAccess、Cloneable、Serializable

实现RandomAccess便于集合元素的快速访问即查询。

实现Cloneable便于集合的拷贝，本质是数组的复制，属于浅拷贝。

实现Serializable便于集合的序列化

**深拷贝、浅拷贝的概念：**

Java中对象的复制通常通过赋值来实现，但是这种方式通过任何一方的引用操作都会改变对象。

Clone方法，可以实现对象的复制，得到一个新的对象。对于一个对象中的属性既包含基本类型又包含引用类型，clone方法完成的拷贝是浅拷贝，因为引用类型的对象仍然是共用的。

深拷贝就是让对象中的任何一个属性，包括基本数据类型和应用数据类型都是不相同的，这个拷贝得到的对象就是一个深拷贝的对象。这种方法需要各个引用类型都实现Cloneable接口，重写clone方法。

以List<Student>为例，直接调用ArrayList的clone方法属于浅拷贝，即ArrayList对象是新对象，但是里面的Student元素是相同的，新老集合指向相同的Student引用。

**2. 成员变量：**

初始容量10、最大容量Integer.MAX\_VALUE – 8(和jvm相关)，底层数据结构Object[]、集合大小size。

**3. 构造方法：**

可指定初始容量、或者直接传递一个集合实现类来构造ArrayList

**4. 常用方法：**

**5. 扩容：**

size+1 > elementData.length()进行扩容;

扩容1.5倍，即1.5(elementData.length());

进行数组的复制拷贝System.arraycopy()。

**6.线程安全问题：**

add方法中elementData[size++] = e;不是原子操作且没有加锁，多线程情况下回出现同一索引处的元素值被覆盖。

#### 二、LinkedList源码解析

查询慢，新增删除快。

**1.实现接口：**

List、Deque、Cloneable、Serializable

实现Deque表示这个一个支持元素在两端进行插入和删除操作的线性集合。

实现Cloneable，集合的浅拷贝。

**2.成员变量**

集合容量size,第一个节点 first、最后一个节点last，底层数据结构是双向链表。

first的特性(first == null && last == null) ||(first.prev == null && first.item != null)。

last的特性(first == null && last == null) ||(last.next == null && last.item != null)。

当集合初始化时,fist和last均为null;当集合中只有一个元素时，first和last是同一对象。first和last节点会携带集合元素的。

**3.构造方法**

支持传递集合Collection的构造方式

**4.常见问题**

无扩容问题，但是元素的查找效率不高，都是从头结点或者尾结点开始依次查找。

#### 三、HashSet源码解析

Set无索引、无序、不可重复。

HashSet<E>的底层是HashMap<E,Object>。

#### 四、TreeSet源码解析

TreeSet<E>的底层是NavigableMap<E,Object>。本质是TreeMap<E,Object>。

#### 五、HashMap源码解析

#### 六、TreeMap源码解析

#### 七、ConcurrentHashMap 源码解析