黑客派

UnsupportedOperationException 的研究

作者：zjhch123

原文链接：https://hacpai.com/article/1487164177985

来源网站：黑客派

<h2 id="1--场景还原">1. 场景还原</h2>
最近在设计的一个类中，有成员属性
<pre><code class="highlight-chroma">Map<String, List<String>> headers = new HashMap<>();
</code></pre>
在实际使用过程中，发现对该属性的 <code>value</code> 进行 <code>add()</code> 操会抛出异常 <code>UnsupportedOperationException</code>。感觉略有奇怪，于是便开始研究
<h2 id="2--什么时候会抛出异常">2. 什么时候会抛出异常</h2>
使用这段代码
<pre><code class="highlight-chroma">Map<String, List<String>> headers = new HashMap<>();
headers.put("zjh", Arrays.asList("123"));
headers.get("zjh").add("456");
</code></pre>
会抛出异常。
 而使用这段代码
<pre><code class="highlight-chroma">Map<String, List<String>> headers = new HashMap<>();
String[] str = {"123"};
headers.put("zjh", new ArrayList<>(Arrays.asList(str)));
headers.get("zjh").add("000");
</code></pre>
不会抛出异常。
而在前文所述中，我设计的类内就是使用第一段代码的形式才抛出了异常。
 以此，我便猜是 <code>Arrays.asList()</code> 有问题。
<h2 id="3--进一步研究">3. 进一步研究</h2>
使用这段代码
<pre><code class="highlight-chroma">List<String> lists = Arrays.asList("123","456","78","000");
lists.add("999");
</code></pre>
会抛出异常。
查看 <code>Arrays.asList</code> 的源码，发现其返回的是对象属性是 <code>ArrayList</code>
之后再往下看
查看这个父类 <code>AbstractList</code> 的情况
 其内部的 <code>add</code> 实为
<pre><code class="highlight-chroma">public boolean add(E e) {
    add(size(), e);
    return true;
}

public void add(int index, E element) {
throw new UnsupportedOperationException();
}
</code></pre>

一切真相大白，<code>Arrays.asList</code> 方法返回的 <code>List</code> 对象内部没实现 add 方法！

<h2 id="4--更多的实验">4. 更多的实验</h2>
其实，<code>Arrays.asList</code> 方法的返回就是对传入参数数组的引用。对数组进行修
也会导致 <code>Arrays.asList</code> 的返回值发生改变。
 如下代码：
<pre><code class="highlight-chroma">String[] str = {"123","456","789","000"};
List<String> lists = Arrays.asList(str);
str[1] = "9999";
System.out.println(str[1]); // 9999
System.out.println(lists); // [123, 9999, 789, 000]
</code></pre>
这也可以解释为什么 <code>Arrays.asList</code> 的返回值无法 <code>add</code> 或者
code>remove</code>。如果允许其可以变长，那么原始的数组内容应该如何修改呢？所以最后 Jav
的开发人员便使其返回一个不可修改长度的 <code>List</code>。
 以上所述在源码里也能体
出来，更具体的大家可以参阅 <code>Arrays</code> 这个类的源码！
<h2 id="5--解决方案">5. 解决方案</h2>
我的建议是如果对 List 内部元素会有增删的操作的话，慎用 <code>Arrays.asList</code> 方
，如果非要用，可以绕一绕。
 例如：
<pre><code class="highlight-chroma">List<String> lists = new ArrayList<>(Arrays
asList("111", "222", "333"));
</code></pre>
如此生成的 list，可变长！
 代码稍微长一点，可以使用循环来给 <code>List</code> 内
加元素。
 当然，我只是提供了最简单的解决方案，具体方案要放在具体项目中选择使用。