



黑客派

细节见真功之 Javadoc

作者: [88250](#)

原文链接: <https://hacpai.com/article/1402537988442>

来源网站: [黑客派](#)

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

<p>作为一个 Java 程序员，Javadoc 大家都应该写过吧，是不是觉得写的时候特简单呢？相信看完文后你会若有所思。另外，本文非常适合处女座程序员阅读。</p>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
<script>
 (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<h3 id="toc_h3_0">句号</h3>
<p>为什么是句号而不是其他的标点符号呢？因为这涉及到一个 JDK 文档生成的规则：</p>
<blockquote>
<p>The first sentence of each doc comment should be a summary sentence, containing a concise but complete description of the API item. This means the first sentence of each member, class, interface or package description. (成员、类、接口或包注释的第一个句子将作为该注释项的总结，这个句子应该是言简意赅的)</p>
</blockquote>
<p>既然提到了句子，那就说明应该用标准的方式——句号来进行第一个句子的断句识别。也许是为简洁，Javadoc 工具并没有“多语言句号识别”这个特性，所以不管我们用什么语言撰写文档注释，句号都必须是在英文状态下的句号——.</p>
<p>比如在 NetBeans IDE 里（默认编辑器配置）可以很直观的看到用于总结的第一个句子和后面描述的区别：</p>
<p></p>
<p>在生成 Javadoc 后，类概要页面我们可以看到最终的效果：</p>
<p></p>
<p>在使用中文撰写文档注释时，为了保持整体风格一致，在所有需要使用句号的地方有两种策略可择：</p>

都使用英文句号：这样做可以让生成的文档句号统一，但缺点是看上去有点别扭
只有第一个句子使用英文句号，其余地方都是要中文句号：这样做后生成的文档看上去比较顺眼但别人可能会奇怪为什么第一个句号是.

<p>当然，最彻底的解决方案是不用中文写文档注释，这样就不存在要统一的问题了！</p>
<p> </p>
<p>下面我们重点介绍各种你熟悉的或是不熟悉的 Javadoc 文档标记（它们很有内涵）</p>
<h3 id="toc_h3_1">@author</h3>
<p>该标记使用频率是所有文档标记中最高的，我想这是因为：</p>


```
m2301.livefilestore.com/y2p0JF14tRuSfjarfg9QuYSOCdvpLC-LzY73CTKqLlrxpxZmO1SH9y5aq
v4WOdlhOdhAY2TZulhTswAocMoL5o8sveStzfFjxuZnWCTYutXYI/value-ref-code.png?psid=1"
</a></p>
<p><a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fplokra.dm2301.livefiles
ore.com%2Fy2pG-6eXeG1fRAO1O1AbGzMofB3uyMPxd-q1Zvu680JceRBttDjUhpeYqNhyMVu
B_ky6rVnck7Q9YwfK8pRdjFAmmZSIRWY9DOvrtQIVztX4%2Fvalue-ref-html.png%3Fpsid%3D1
class="fancybox" target="_blank" rel="nofollow ugc"></a>&nbsp;</p>
<h3 id="toc_h3_4">{@inheritDoc}</h3>
<p>&nbsp;<em>这个标签体现了 Java 面向对象的精辟所在：不但可以类可以集成，连文档都可以继承（
见 Java 在经典面向对象概念上的完备与圆润）。</em></p>
<p>比如有个计算面积的接口：</p>
<p><a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fplokra.dm2301.livefiles
ore.com%2Fy2pJKJ6CL0KWglqfNdV73U3dySu1SAeX0uMpMBe2GK3rRrX-AgsbdveU5nkUY4of
PvWt8K-3Fddkuxr9cQMmkrnOnexiBHVsvCskw9p0G0L58%2Finheritdoc-interface-code.png%
Fpsid%3D1" class="fancybox" target="_blank" rel="nofollow ugc"></a></p>
<p>它的实现方法标注了 {@inheritDoc}<em>（处女座阅读提示：无。）</em>：</p>
<p><a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fplokra.dm2301.livefiles
ore.com%2Fy2pEBtBhvFFNe_z3L-5JNzWcaa6_B-6NcwY24ljeAhkUpOU5Mx4NYZetayMfP-3Dw
rd9uL1mmqQw9q-pJU9K2nATNWL5fSHodgvCDrgAzxfqc%2Finheritdoc-impl-code.png%3Fps
d%3D1" class="fancybox" target="_blank" rel="nofollow ugc"></a></p>
<p>最后生成的文档：</p>
<p><a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fplokra.dm2301.livefiles
ore.com%2Fy2pAfyk_flGUC6Sh1GnOZREbXoB0tnBEIB79GxsaZNeLUkzmE0n-uobEFXlqVaclv4
ChzpqOOn2wapcsEq4vELKaA99sDb7obybpSjkykPSd4%2Finheritdoc-html.png%3Fpsid%3D1"
lass="fancybox" target="_blank" rel="nofollow ugc"></a></p>
<ul>
<li>基类的文档注释被继承到了子类</li>
<li>子类可以再加入自己的注释（特殊化扩展）</li>
<li>@return @param @throws 也会被继承</li>
</ul>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
```

</script>

<p>其实在不写 {@inheritDoc} 的情况下也存在文档注释的继承，具体规则请看这里。</p>

<h3 id="toc_h3_5">{@link} {@linkplain}</h3>

<p>这两个链接标记大家用/见的应该比较多，但它们有什么区别、在什么场景下该怎么使用很少有能够区分开（我猜你要用的时候一般也都是用 link 吧）。</p>

<p>看看官网的标准解释：</p>

<p></p>

<p>
link 和 linkplain 的实参都是 package.class#member label。唯一的不同就是因为字体不同，如果 label 是个纯文本，那就使用 linkplain 吧。（根据这点我严重怀疑 Javadoc 文档标记的设计者是处女座，~~）</p>

<h3 id="toc_h3_6">pre</h3>

<p>没错，这就是那个 HTML 标签，用于显示“原始样子”的。这个标签在写 Javadoc 的时候非常用，用或者不使用在打码的时候看上去差别不大：</p>

<p> <a></p>

<p>但最终生成 apidocs 之后差别一目了然（处女座阅读提示：在源码文档注释中特别需要意 pre 后 { 的位置，紧跟 *，无空格）：</p>

<p></p>

<h3 id="toc_h3_7">@since</h3>

<p>这个从字面的意思上很好理解，所以使用的比较多（如同 @author、@version 一样）。但问是大家写的时候表达的意思五花八门，常见的有：</p>

想表达日期/时间
@since 2014-01-01
@since 2014-01-01 14:00:00

想表达可运行的 JDK 版本
@since JDK1.5

想表达加入这个元素的版本
@since 1.0.0

<p>根据官方文档解释，@since 表达的是被标记元素是哪个发布版本引入的（3）。比如别人在我的文档注释中看到</p>

<p></p>

<p>那他可以（应该）认为这个类是在该程序对外发布 1.0.0 版本时已经引入的。如果他要做二次开，那他就可以很清晰的向后兼容了（我们在用 JDK 的时候就是这个场景）。</p>

<h3 id="toc_h3_8">@version</h3>

<p>提到了 @since 就自然会联想到 @version，因为它们的实参都是版本相关的。@version 要表的是被标记元素自己的版本（注意对比 @since），也就是说这个版本只要代码改过就应该发生变化而 @since 是不会变的。</p>

<p>官方文档也解释了怎么用好这个文档标记：通过 SCCS 字符 "% % , %G%"，例如 1.39, 02/28/97（文件版本号, 日期）生成。但实际上很少有项目这么做（至少目前racle JDK 没这么做，甚至都没有使用 @version，或者是使用了但最后由于特殊原因总体移除了）大家一般都是 @version 1.0.0 然后就再也不修改了，不管被标记的元素改了多少次（这样的做法还如不写）。</p>

<p>当然，通过版本控制系统 hook 来做是比较经典的做法，不过这样总感觉没有把这个标记的能力全发挥出来。在我们的项目里是这样使用的：@version 1.2.3 4, Jun 9, 2014</p>

<p>重点是版本号部分，在这个例子中从左到右（1.2.3.4）分别表示：</p>

- 兼容位 1，表示兼容性，如果 +1 了说明这个修改是不兼容的
- 特性位 2，表示已引入了两个特性，每次 +1 说明引入一个新特性
- 缺陷修复位 3，表示已经修复了 3 个缺陷，每次 +1 说明修复了一个缺陷
- 重构位 4，表示已经进行了 4 次重构，每次 +1 说明重构了一次

<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>

<!-- 黑客派PC帖子内嵌-展示 -->

<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>

<script>

(adsbygoogle = window.adsbygoogle || []).push({});

</script>

<p>前面 3 位表达的意义和 Semantic Versioning 建议一致，重构我觉得非常重要，所以也加了进来。</p>

<h3 id="toc_h3_9">@exception @throws</h3>

<p>这两兄弟的情况比 @link @linkplain 更纠结（人家 link 兄弟最起码可以区分出使用场景）。按照官方文档解释：它们完全是同义词，没有任何区分。那当年 Sun 在 JDK1.2 的时候什么要加入 @throws 呢——答案是起名失误了，词性没弄匹配：@throws Exception 比 @exception Exception 更符合语法，代入感更好！（细节：In javadoc, what is the difference between the tags @throws and @exception?）</p>

<h3 id="toc_h3_10">标记总表</h3>

<p>来张 Javadoc 文档标记总表：</p>

<div>

```

<table summary="list of current javadoc tags" width="40%">
  <tbody>
    <tr>
      <th><span>Tag</span></th>
      <th><span>Introduced in JDK/SDK</span></th>
    </tr>
    <tr>
      <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23author" target="_
blank" rel="nofollow ugc"><code>@author</code></a></td>
      <td>1.0</td>
    </tr>
    <tr>
      <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23code" target="_
lank" rel="nofollow ugc"><code>{@code}</code></a></td>
      <td>1.5</td>
    </tr>
    <tr>
      <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23docRoot" targe
="_blank" rel="nofollow ugc"><code>{@docRoot}</code></a></td>
      <td>1.3</td>
    </tr>
    <tr>
      <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23deprecated" tar
et="_blank" rel="nofollow ugc"><code>@deprecated</code></a></td>
      <td>1.0</td>
    </tr>
    <tr>
      <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23exception" targe
="_blank" rel="nofollow ugc"><code>@exception</code></a></td>
      <td>1.0</td>
    </tr>
    <tr>
      <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23inheritDoc" targ
t="_blank" rel="nofollow ugc"><code>{@inheritDoc}</code></a></td>
      <td>1.4</td>
    </tr>
    <tr>
      <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23link" target="_bl
nk" rel="nofollow ugc"><code>{@link}</code></a></td>
      <td>1.2</td>
    </tr>
    <tr>
      <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23linkplain" target
="_blank" rel="nofollow ugc"><code>{@linkplain}</code></a></td>
      <td>1.4</td>
    </tr>
  </tbody>
</table>

```

```

<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23literal" target="
blank" rel="nofollow ugc"><code>{@literal}</code></a></td>
  <td>1.5</td>
</tr>
<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23param" target="
blank" rel="nofollow ugc"><code>@param</code></a></td>
  <td>1.0</td>
</tr>
<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23return" target="
blank" rel="nofollow ugc"><code>@return</code></a></td>
  <td>1.0</td>
</tr>
<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23see" target="_bl
nk" rel="nofollow ugc"><code>@see</code></a></td>
  <td>1.0</td>
</tr>
<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23serial" target="_
lank" rel="nofollow ugc"><code>@serial</code></a></td>
  <td>1.2</td>
</tr>
<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23serialData" targe
="_blank" rel="nofollow ugc"><code>@serialData</code></a></td>
  <td>1.2</td>
</tr>
<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23serialField" targe
="_blank" rel="nofollow ugc"><code>@serialField</code></a></td>
  <td>1.2</td>
</tr>
<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23since" target="_
lank" rel="nofollow ugc"><code>@since</code></a></td>
  <td>1.1</td>
</tr>
<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23throws" target=
_blank" rel="nofollow ugc"><code>@throws</code></a></td>
  <td>1.2</td>
</tr>

```



```

<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23value" target="_
lank" rel="nofollow ugc"><code>{@value}</code></a></td>
  <td>1.4</td>
</tr>
<tr>
  <td><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2F
avase%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html%23version" target=
_blank" rel="nofollow ugc"><code>@version</code></a></td>
  <td>1.0</td>
</tr>
</tbody>
</table>
</div>

```

<p>这个表是 JDK7 技术手册里的，从中我们可以看出，自 JDK1.5 以后就没有加过新的文档标记了
目测有两个原因：</p>

- Oracle：“这些已经足够开发人员使用了，没必要加新的了”
- *Sun：“看吧，Oracle 严重缺乏折腾精神，当初不应该卖给它的”

<p>文档标记介绍完了，下面我们来聊聊 Javadoc 相关的其他侃点。</p>

<p>对于 POJO 来说，这几个方法的注释格式非常固定，一般我们都是用 IDE 自动生成：这样的话 人一看到这样固定格式的注释（或者索性不要添加任何注释）就知道这部分相对于其他部分并不重要 而一旦有的 getter/setter/isTrue 注释不是这样约定的，那就说明了实现上面不只是简单的 get/set/i , 还加入了额外的逻辑处理。</p> <p>Javadoc 文档注释也有对齐（不是前面 pre 例子那种），这里说的对齐主要指的是以源码视图 到的，最典型的场景就是在给方法添加文档注释的时候，我们经常看到两种风格：</p> ``` <p></p> ``` <p>第一眼看上去是不是风格 2 要顺眼得多？但最好还是使用风格 1，因为：</p> - 这和编辑器配置的字体系有关，如果不是（适合的）等宽字体，那会非常的参差不齐 - 浪费空间，特别是当注释内容多了需要换行的时候会很别扭 - 最后生成的 apidocs 效果是一模一样的（无对齐） <p>（一些 IDE 默认格式化文档注释的时候也是使用风格 1 进行格式化的，强烈建议使用风格 1）</p> <p>和前面几点打码风格相关的细节比起来，包注释是具有一定的实用性的。虽然大家可能用得很少 但看得应该比较多，就是这部分：</p> ``` <script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr ipt> ``` <!-- 黑客派PC帖子内嵌-展示 --> ``` <ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in ``` 原文链接：[细节见真功之 Javadoc](#)

```
>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p><a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fplokra.dm2301.livefiles
ore.com%2Fy2pxjXlPHiuouivoM_l4eK9QzBSFj8bhvXX0Q5RWPYt-XYRKNi0Wy3iFFHe14jr9NkT
XjSlpXHEPKZ1ZurUX6dRbyz-jZBtqyzEaNFAKCZDvo%2Fpackage.png%3Fpsid%3D1" class="fa
cybox" target="_blank" rel="nofollow ugc"></a></p>
<p>这里我们使用了两种方式来生成包文档：</p>
<ul>
<li>package.html：这是 JDK1.5 以前的方式，现在已经不推荐使用</li>
<li>package-info.java：目前推荐方式，因为这样可以<strong><em>使用注解</em></strong>
</li>
</ul>
<p>在包上面使用注解？这个用法和在其他地方使用注解一样，只是被标注的元素变成了包，在运行
可以获取到包的注解，然后做你想做的事情吧！</p>
<h3 id="toc_h3_14">中文</h3>
<p>一开始我们提到了句号的问题（那的确是一个问题），最后我们来看看中文在写文档注释的时候
非常值得注意的一点（其实不只是 Javadoc 文档注释，该建议也适用于其他一些情况）：在中文和
文、数字中间插入一个空格（本文就是这样排版的）。</p>
<p>比如说：</p>
<ul>
<li>我觉得Java非常cool，特别是JDK8中的lambda，真希望9能带来更多实用特性</li>
<li></li>
<li>我觉得 Java 非常 cool，特别是 JDK8 中的 lambda，真希望 9 能带来更多实用特性</li>
</ul>
<p>后者看上去就比前者更舒服一些，这样的排版方式适合纯文本编辑器，如果使用的是 Office 之
的工具就不需要手动空格了，因为它们默认已经处理的很好了。</p>
<h3 id="toc_h3_15">总结</h3>
<p>本文介绍了一些 Javadoc 文档注释相关的细节，从这冰山一角相信你对 Java 也有了另一番体验
Java 的进化、工业化)。</p>
<p>总结一下本文内容：</p>
<ul>
<li>对于文档标记，大家可以尽量尝试使用：把自己的思想通过适合的方式表达给他人是一种好习惯
</li>
<li>对于风格相关，大家也可以适当尝试（处女座/强迫症就算了）：某大厂在某次改句号问题后出
过生产故障</li>
</ul>
<p>不过，大家也千万不要太较真，毕竟对于一个好的程序来说，代码应该就是它的文档（之一）。<
p>
<h3 id="toc_h3_16">参考</h3>
<ul>
<li><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fwww.oracle.com%2Fte
hnetwork%2Fjava%2Fjavase%2Fdocumentation%2Findex-137868.html" target="_blank" rel="
ofollow ugc">http://www.oracle.com/technetwork/java/javase/documentation/index-137868.
tml</a></li>
<li><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fdocs.oracle.com%2Fjav
se%2F7%2Fdocs%2Ftechnotes%2Ftools%2Fwindows%2Fjavadoc.html" target="_blank" rel="
ofollow ugc">http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html<
a></li>
</ul>
```