<u>=</u>Q

下载APP



用户故事 | 与新特性开发者对话

2021-12-24 Jxin

《深入剖析Java新特性》 课程介绍>



讲述:梁正霖 时长 06:27 大小 5.92M



你好,我是Jxin。目前是一名供应链业务开发,坐标杭州。

其实,我算是范老师的"老学员"了。之前学习范老师的 ②《代码精进之路》,我就有不小的收获。这次看到开设了新课程,便第一时间参与了学习。关于 9-17 的 Java 新特性,我也是第一次了解,算是再次跟着范老师长见识了。学习过程中我有一些心得,不成熟分享给你,希望能给你一些参考。如果你也有一些见解或想法,欢迎讨论。

举个例子。 Java 里面我们遍历一个 List 可以用 For / Iterator / Stream 三种模式。这就是确定性的知识,也就是**学习的第一阶段**。但是,你问我遍历 List 该用哪种模式,我会说不一定。会说不一定就是知道要考虑不确定的场景,这就到了**学习的第二阶段**。为什么不一定?从性能角度看,如果 List 量级较小,For 和 Iterator 其实性能较优,加之常规编码习惯应当选择 For;但如果量级较大,因为 Stream 支持 parallelStream 转换,可以提供便捷的并行转换,所以会选择 Stream。另外,从可靠性、可读性角度来看,适用的模式又有所不同了。知道什么场景用什么遍历方式,这就算到了**学习的第三阶段**。

从这三个阶段出发,我又审视了一下自己学习《深入剖析 Java 新特性》的过程。

学习确定性知识

《深入剖析 Java 新特性》的行文有几分纪传体的味道。以每个特性为中心描述其发展史和应用,这样更贴合专栏零散时间学习的模式,有利于我们独立地学习每一种特性。行文提供的确定性知识主要是以下几点:

该特性在 JDK 版本的发展史,啥时候预览,啥时候改进,啥时候发布;

为什么要实现该特性;

该特性是什么样的;

该特性有哪些应用场景。

针对这四点,我个人的看法是:

首先,讨论新特性不加版本就是耍流氓。知道 JDK 版本的发展史,就相当于加了一条时间线上的认知。如此,你才能明确什么版本可以练手,什么版本还有哪些问题,什么版本可以正式使用。老师虽然提到了这一部分,可惜都一笔带过了。如果再展开讲讲,从什么时候开始考虑这个特性?为什么要加在这个版本?为了加这个特性放弃了什么?在每个版本选择哪些特性的主旨是什么?或许能让大家学习到语言开发者团队的一些决策原则,理解JDK 发展的风格/风向。

关于学习新特性的原因。因为程序员在学习过程中会遇到很多概念,这个时候如果缺少一些背景和上下文,就很容易曲解作者的原意。所以,由语言开发者自己来跟我们讲讲,为什么要有这个特效,可以说是最准确、最官方的解答了,是一次难得的机会。

至于特性本身的学习,我觉得反而是也应该是最简单的。我们在做设计时,如果一个解决方案非常复杂,或许应该停下来想想问题本身是否就有问题。学习新特性也是如此,如果你觉得一个特性很反人类,不好用或者有更好的解法,希望你能在留言区留下自己的观点,让我们一起讨论。你的建议对于 Java 社区很重要,说不定无心插柳就可以对 Java 社区产生有建设性的影响。

最后,说到特性的应用场景。我觉得场景是难以穷举的,专栏中的场景远不及我们真实会碰到的场景多。所以,除了知道这些场景,我们还需要去思考场景背后能够适用该特性的关键因素,只有看到关键因素,才能应对更多的场景。当然,如果有碰到或想到什么好场景,也非常希望大家都能分享出来,让我们一起思考。

总之,兼听则明,偏听则暗。在学习的第一阶段,就是要尽可能多地去收集各维度的信息,并将这些信息联系起来。这样才有机会透过表面的知识发起更深度的思考。从而更准确地应用自己的知识,为应万变打好基础。

发现不确定场景

如何更全面地考虑不确定的场景?事有两面,如果只讨论某个特性在哪些场景适用,却不思考它在哪些场景有弊端,我觉得是不健全的。所以后续的课程,希望老师能把会出现弊端的场景也讲一讲,通过利害两方面场景的枚举,让大家更全面地思考特性背后适用的逻辑。

关于这一点,其实我们可以和范老师一起来构思,就当来找茬,想想什么场景不适用,存在哪些弊端。让专栏"动起来",也让学习"深下去"。

因地制宜

学习的第三个阶段,就是我之前说的因地制宜了。如何做到因地制宜?因地制宜的本质绝非记住所有场景,因为人类的认知复杂度是有上限的。因地制宜的本质在于洞察场景背后的本质,有自己权衡的原则,能够抓住关键因素,基于原则,做出适合的决策。所以,希望范老师可以借着讲解新特性的机会,也谈谈自己的权衡思路和依据,让大家看看语言开发者心中的最佳编程实现是什么样的。

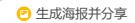
总结

如果要总结一下我的学习心得的话,我觉得,除了要扎实地学习确定性的知识,还要:

- 1. 考虑不同场景的弊端,警防误用,丰富思考;
- 2. 梳理知识点之间的关系,并借此发起更深度的思考。

另外,我还想表达一下我个人对于专栏的期待:希望范老师针对确定性的知识可以多加点维度,也多分享一些自己的权衡思路,带我们看看语言开发者的设计视角。

分享给需要的人, Ta订阅后你可得 20 元现金奖励



△ 赞 2 **△** 提建议

⑥ 版权归极客邦科技所有,未经许可不得传播售卖。页面已增加防盗追踪,如有侵权极客邦将依法追究其法律责任。

上一篇 用户故事 | 保持好奇心, 积极拥抱变化

精选留言(1)





范学雷

2021-12-24

谢谢分享这么好的观点!如果以对话的方式讨论Java的新技术,可能是一种更好的方式。 也许,每一个技术都可以讨论两个小时。



