

名字 1：JVM 从入门到精通

对 Java 程序员来说，Java 虚拟机（JVM）可以说是既熟悉又神秘，很少有 Java 程序员能够抑制自己探究它的冲动。可惜透彻讲解 JVM 的书籍（尤其是国内出版的）简直少之又少，这里就更不要说能系统讲解 JVM 的视频教程了。尚硅谷的这套 JVM 从入门到精通，绝对可谓是研究 JVM 的程序员及广大 JVM 爱好者的福音。

为什么要学 JVM？

- 面试的需要：入职 BATJ、TMD、PKQ 等一线大厂不光关注技术的广度，更关注技术的深度，JVM 技术是大厂面试的必备技能，掌握越深越好
- 中高级程序员、架构师必备技能：架构师每天都在思考如何让我的系统更快，如何避免系统出现性能瓶颈。单纯的依靠物理机不足以解决问题，分析系统性能、调优系统瓶颈离不开对 JVM 中内存、垃圾回收、字节码指令、性能监控工具、调优参数的熟练掌握。
- 精进技术、极客追求：JVM 是 Java 生态的核心价值的体现，垃圾回收算法、JIT、底层原理值得每个程序员去探索。同时，JVM 作为跨语言的平台，对于深入理解 Scala、Kotlin、JavaScript、Jython、Groovy 也很有帮助。

课程内容共分为三个篇章：

上篇：内存与垃圾回收篇

中篇：字节码与类的加载篇

下篇：性能监控与调优篇

课程适合人群：

- 拥有一定开发经验的 Java 平台开发人员
- 虚拟机爱好者，JVM 实践者
- 有一定的 Java 编程基础并希望进一步理解 Java 的程序员

尚硅谷宋红康老师亲自主刀，带你一刀一刀剖析 JVM 的内部细节，同时，一如既往的内容深入、通俗易懂、幽默、严谨。同时，由于 JVM 理论性较强，课程内容中也尽可能多的引入实践案例，并且全篇图示丰富（仅上篇就近 40 张经典内存图），具备非常强的指导意义。

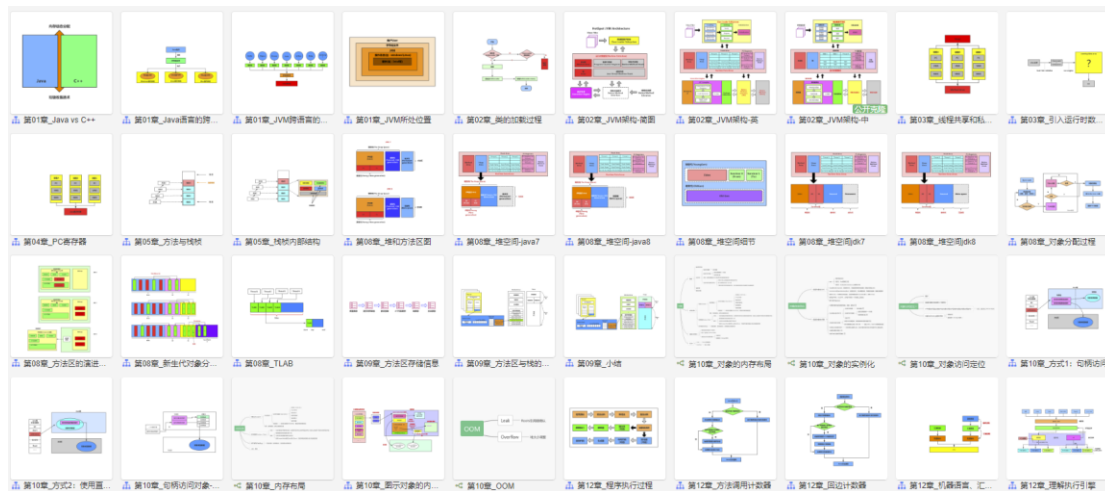
我的课件里，有这样一页内容，需要可以用：

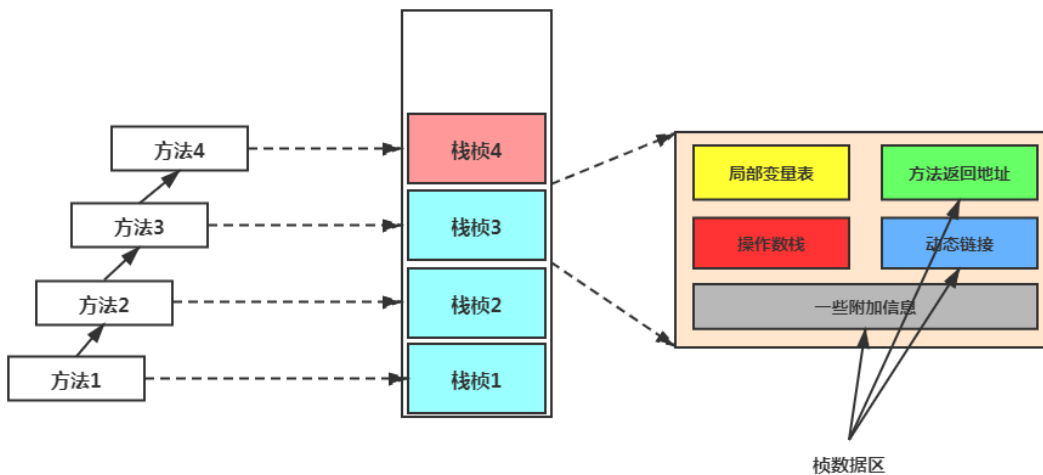
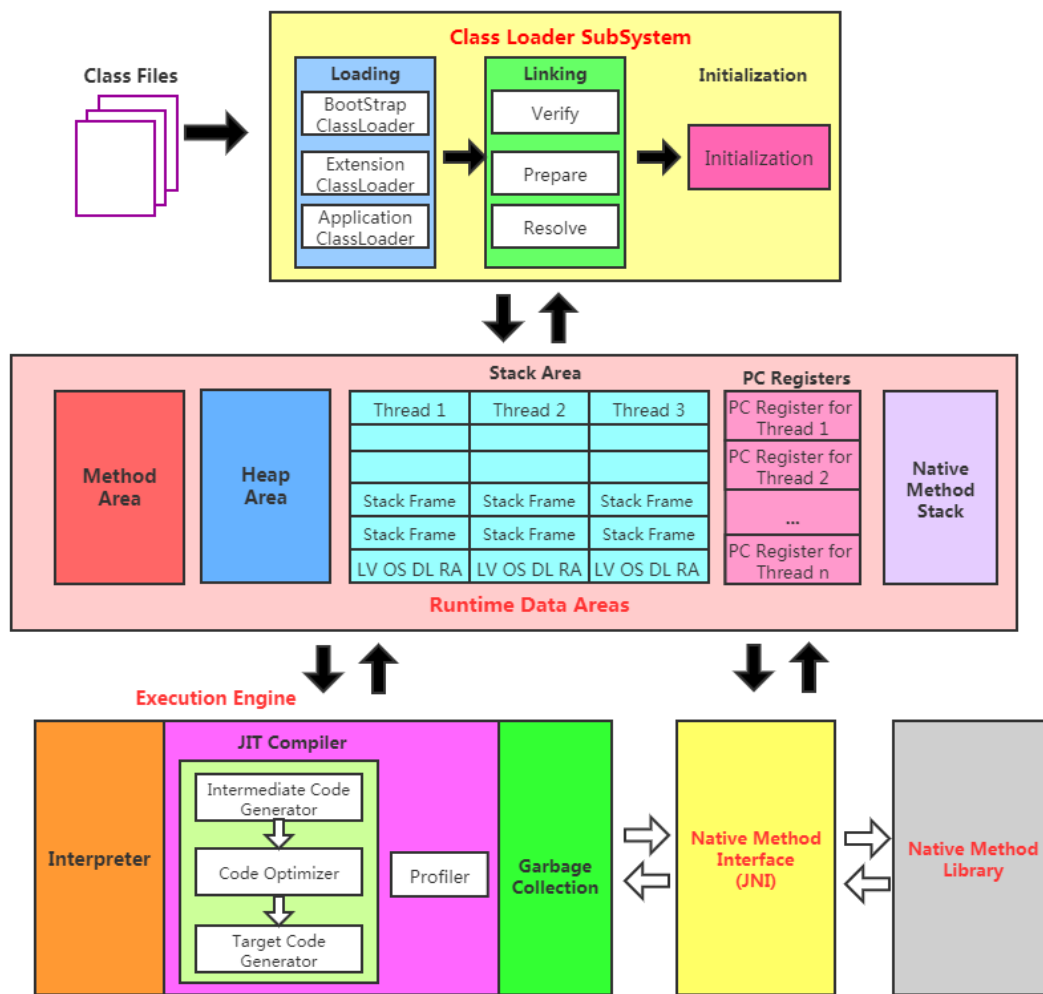
作为 Java 工程师的你曾被 JVM 伤害过吗？

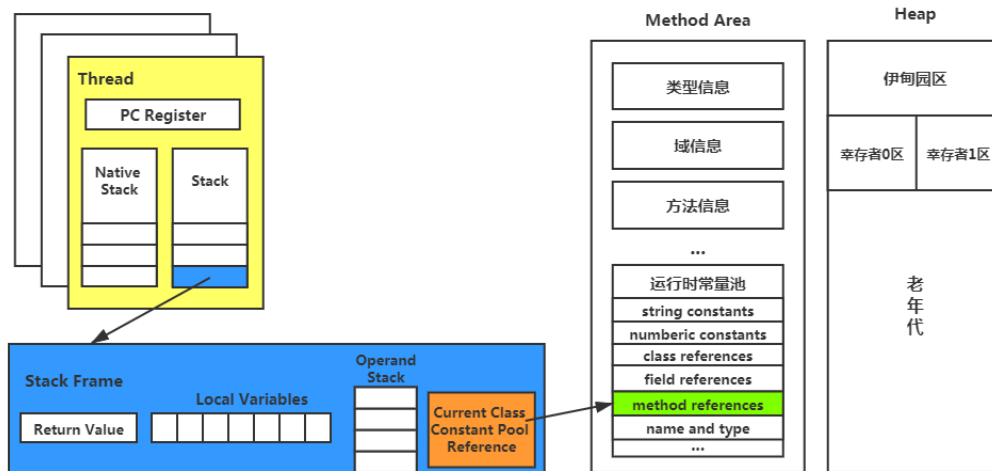
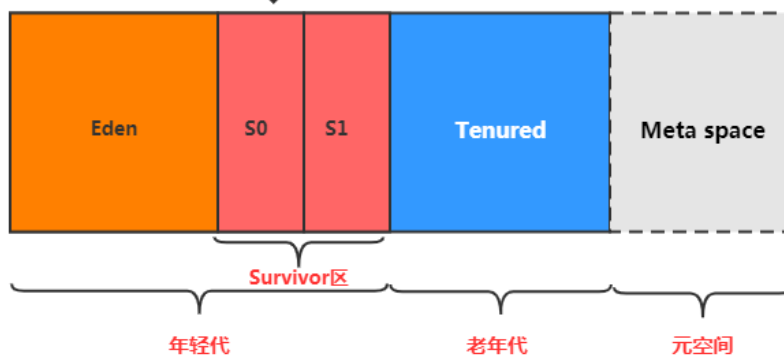
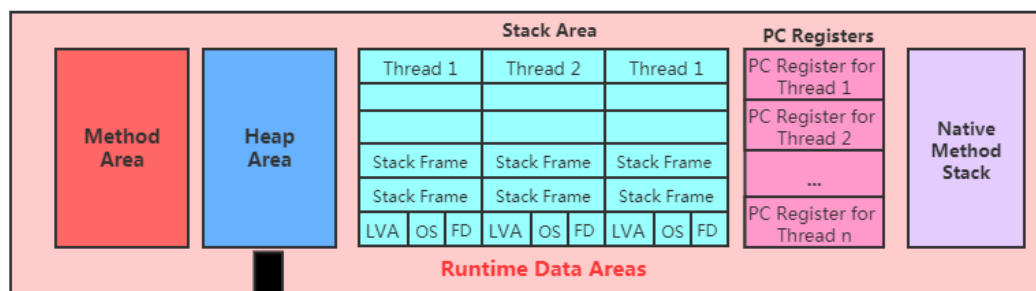
你是否也遇到过这些问题？

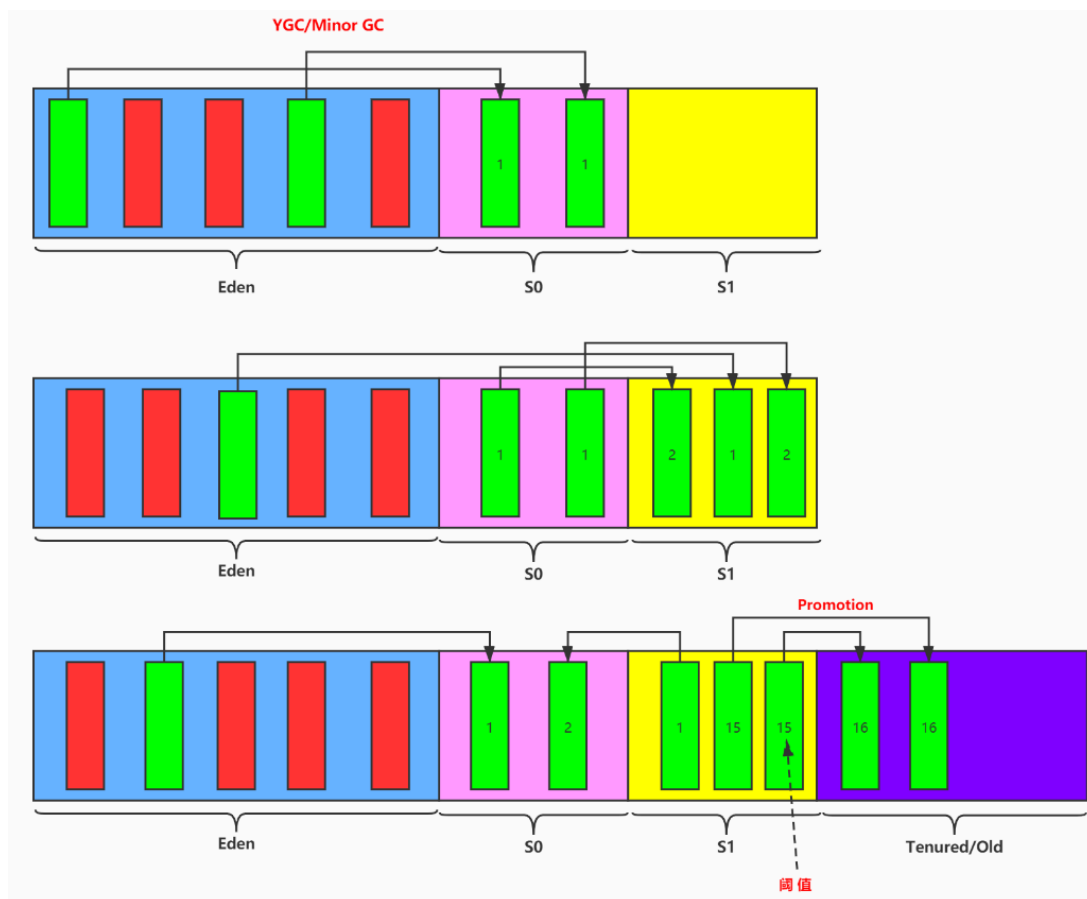
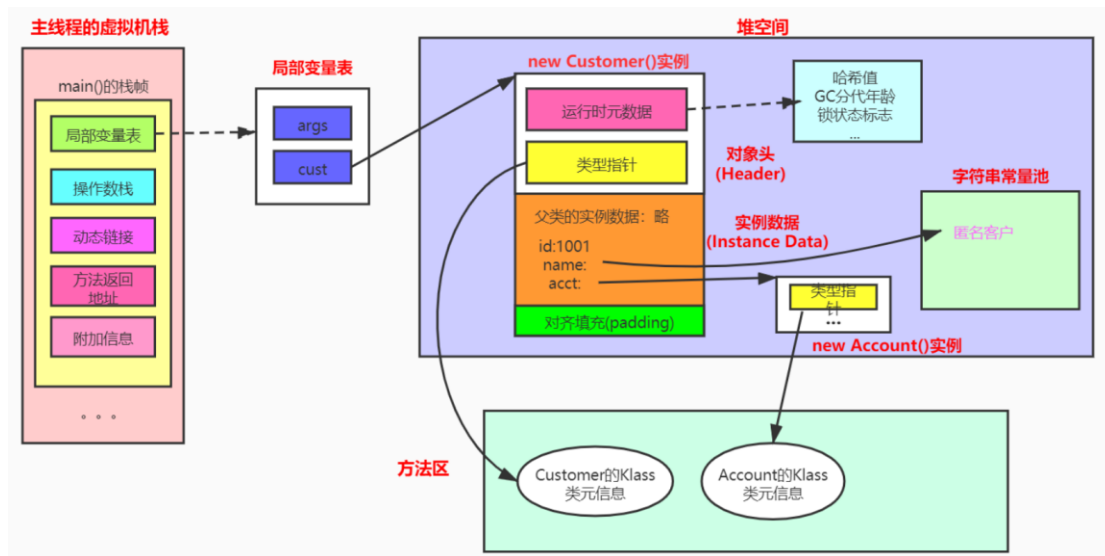
- 运行着的线上系统突然卡死，系统无法访问，甚至直接 OOM！

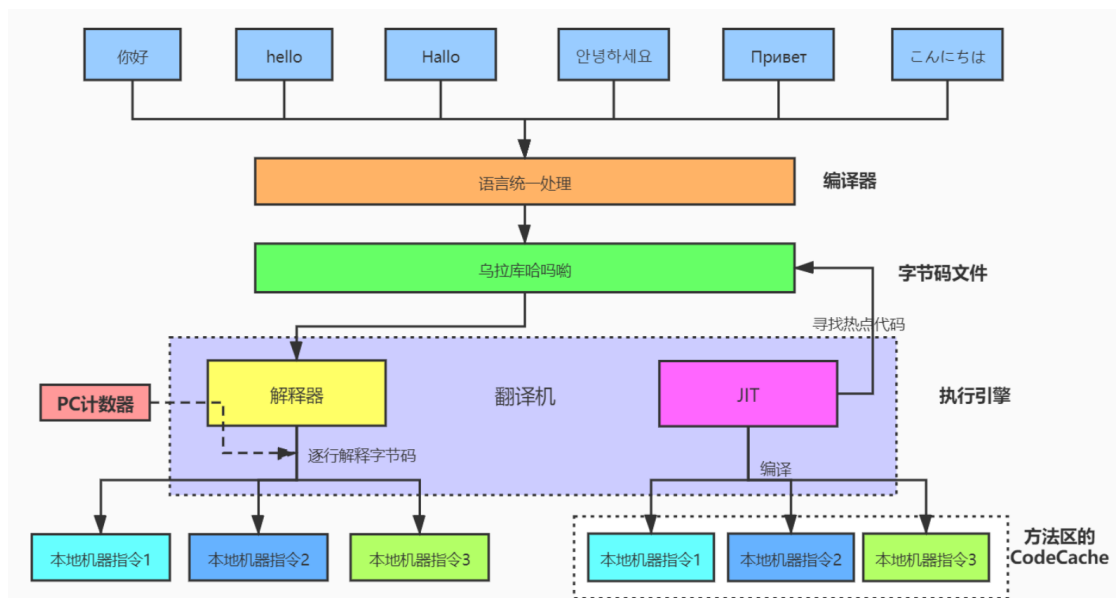
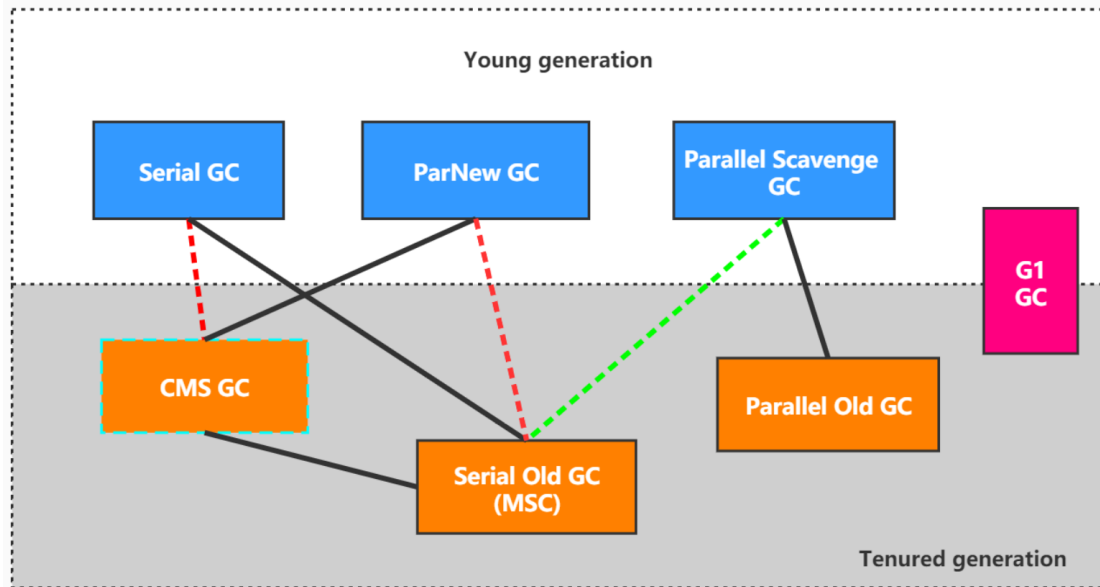
- 想解决线上 JVM GC 问题，但却无从下手。
- 新项目上线，对各种 JVM 参数设置一脸茫然，直接默认吧，然后就 JJ 了
- 每次面试之前都要重新背一遍 JVM 的一些原理概念性的东西，然而面试官却经常问你在实际项目中如何调优 JVM 参数，如何解决 GC、OOM 等问题，一脸懵逼。

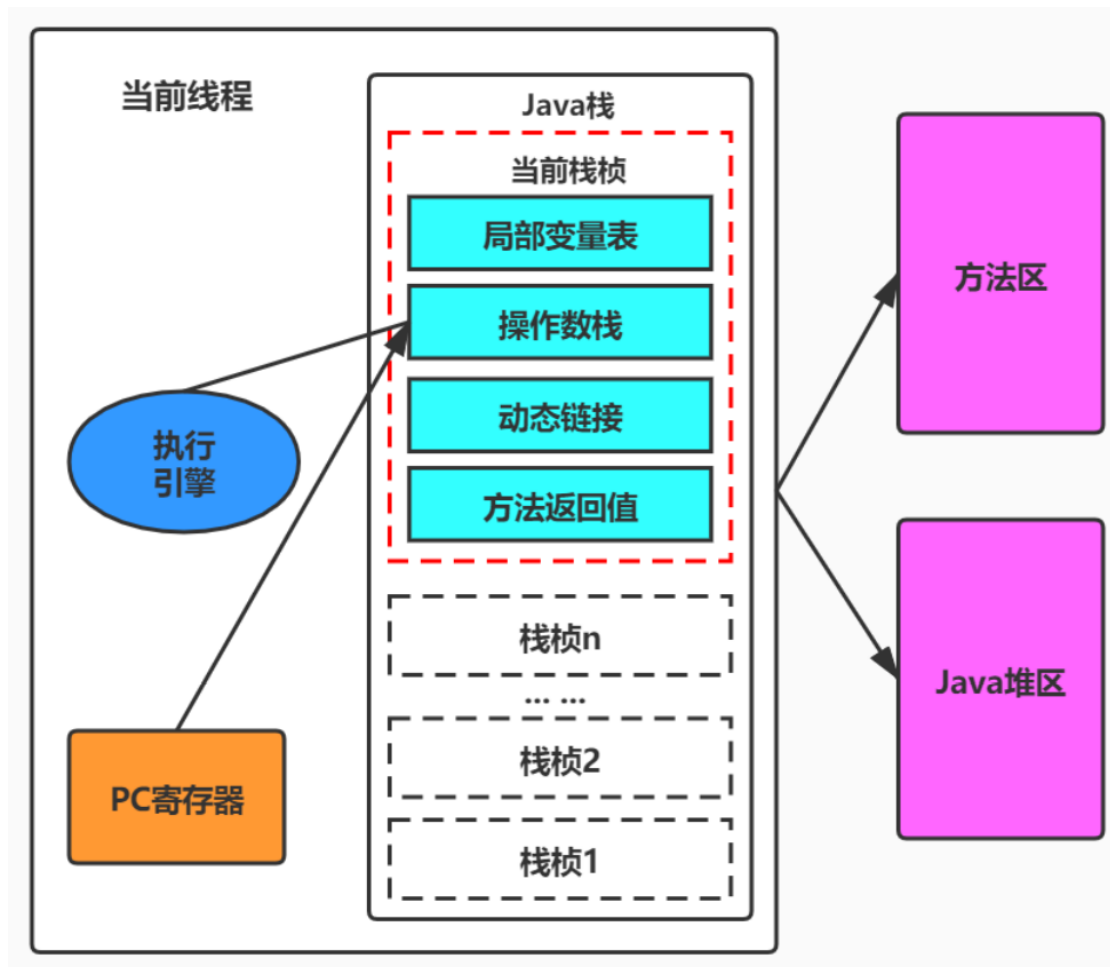












下面是之前已经发布的上篇内容：

- 01-JVM 内存与垃圾回收篇概述
- 02-如何看待 Java 上层技术与 JVM
- 03-为什么要学习 JVM
- 04-面向人群和课程特点
- 05-官方规范下载与参考书目
- 06-跨平台的语言 Java 和跨语言的平台 JVM
- 07-字节码与多语言混合编程
- 08-Java 及 JVM 历史上的重大事件
- 09-虚拟机与 Java 虚拟机介绍
- 10-JVM 的位置
- 11-JVM 的整体结构
- 12-Java 代码执行流程
- 13-区分栈的指令集架构和寄存器的指令集架构
- 14-JVM 的生命周期
- 15-SUN Classic VM 的介绍
- 16-Exact VM 的介绍
- 17-HotSpot VM 的介绍
- 18-JRockit VM 的介绍
- 19-IBM J9 VM 的介绍

20-KVM、CDC、CLDC 的介绍

21-Azul VM 和 BEA Liquid VM 的介绍

22-Apache Harmony 的介绍

23-Microsoft JVM 和 TaobaoJVM

24-Dalvik VM 及其他虚拟机的介绍

25-Graal VM 的介绍

26-内存结构概述

27-概述类的加载器及类加载过程

28-类的加载过程一：Loading

29-类的加载过程二：Linking

30-类的加载过程三：Initialization

31-几种类加载器的使用体会

32-引导类、扩展类、系统类加载器的使用及演示

33-为什么需要用户自定义类加载器及具体实现

34-ClassLoader 的常用方法及获取方法

35-双亲委派机制的工作原理及演示

36-双亲委派机制的优势

37-沙箱安全机制

38-类的主动使用与被动使用等

39-运行时数据区内部结构

40-JVM 中的线程说明

41-PC 寄存器概述

42-PC 寄存器的使用举例

43-解决 PC 寄存器两个面试问题

44-虚拟机栈的主要特点

45-虚拟机栈的常见异常与如何设置栈大小

46-栈的存储结构和运行原理

47-栈帧的内部结构

48-局部变量表结构的认识

49-字节码中方法内部结构的剖析

50-变量槽 slot 的理解与演示

51-静态变量与局部变量的对比及小结

52-操作数栈的特点

53-涉及操作数栈的字节码指令执行分析

54-栈顶缓存技术

55-动态链接的理解与常量池的作用

56-方法的绑定机制：静态绑定与动态绑定

57-4 种方法调用指令区分非虚方法与虚方法

58-invokedynamic 指令的使用

59-方法重写的本质与虚方法表的使用

60-方法返回地址的说明

61-栈帧中的一些附加信息

62-虚拟机栈的 5 道面试题

63-本地方法接口的理解

64-本地方法栈的理解

本次发布的内容：

- 65-JVM 学习路线与内容回顾
- 66-堆空间的概述_进程中堆的唯一性
- 67-堆空间关于对象创建和 GC 的概述
- 68-堆的细分内存结构
- 69-堆空间大小的设置和查看
- 70-OOM 的说明与举例
- 71-新生代与老年代中相关参数的设置
- 72-图解对象分配的一般过程
- 73-对象分配的特殊情况
- 74-代码举例与 JVisualVM 演示对象的分配过程
- 75-常用优化工具概述与 Jprofiler 的演示
- 76-MinorGC、MajorGC 和 FullGC 的对比
- 77-GC 举例与日志分析
- 78-体会堆空间分代的思想
- 79-总结内存分配策略
- 80-堆空间为每个线程分配的 TLAB
- 81-小结堆空间的常用参数设置
- 82-通过逃逸分析看堆空间的对象分配策略
- 83-代码优化之栈上分配
- 84-代码优化之同步省略
- 85-代码优化之标量替换
- 86-代码优化及堆的小结
- 100-方法区的垃圾回收行为
- 101-运行时数据区的总结与常见大厂面试题说明
- 87-方法区概述_栈堆方法区间的交互关系
- 88-方法区的基本理解
- 89-Hotspot 中方法区的演进
- 90-设置方法区大小的参数
- 91-OOM：PermGen 和 OOM：Metaspace 举例
- 92-方法区的内部结构 1
- 93-方法区的内部结构 2
- 94-class 文件中常量池的理解
- 95-运行时常量池的理解
- 96-图示举例方法区的使用
- 97-方法区在 jdk6、jdk7、jdk8 中的演进细节
- 98-StringTable 为什么要调整位置
- 99-如何证明静态变量存在哪
- 102-对象实例化的几种方式
- 103-字节码角度看对象的创建过程
- 104-对象创建的六个步骤

105-对象的内存布局
106-对象访问定位
107-直接内存的简单体验
108-使用本地内存读写数据的测试
109-直接内存的 OOM 与内存大小的设置
110-执行引擎的作用及工作过程概述
111-Java 程序的编译和解释运行的理解
112-机器码_指令_汇编_高级语言理解与执行过程
113-解释器的使用
114-HotspotVM 为何解释器与 JIT 编译器并存
115-热点代码探测确定何时 JIT
116-Hotspot 设置模式_C1 与 C2 编译器
117-Graal 编译器与 AOT 编译器
118-String 的不可变性
119-String 底层 Hashtable 结构的说明
120-String 内存结构的分配位置
121-两个案例熟悉 String 的基本操作
122-字符串拼接操作的面试题讲解
123-字符串变量拼接操作的底层原理
124-拼接操作与 append 操作的效率对比
125-intern()的理解
126-new String()到底创建了几个对象
127-关于 intern()的面试难题
128-面试的拓展问题
129-intern()的课后练习 1
130-intern()的课后练习 2
131-intern()的空间效率测试
132-StringTable 的垃圾回收测试
133-G1 垃圾收集器的 String 去重操作
134-垃圾回收相关章节的说明
135-什么是 GC, 为什么需要 GC
136-了解早期垃圾回收行为
137-Java 自动内存管理介绍
138-垃圾回收相关算法概述
139-引用计数算法的原理及优缺点
140-Java 代码举例_Python 的引用计数实施方案
141-可达性分析算法与 GC Roots
142-对象的 finalization 机制
143-代码演示可复活的对象
144-使用 MAT 查看 GC Roots
145-使用 JProfiler 进行 GC Roots 溯源
146-使用 JProfiler 分析 OOM
147-标记-清除算法原理及优缺点
148-复制算法原理及优缺点

149-标记-压缩算法原理及优缺点
150-不同指标上对比三种算法
151-分代收集算法的说明
152-增量收集算法原理及优缺点
153-分区算法的说明
154-垃圾回收相关概念的概述
155-System.gc()的理解
156-手动 gc 理解不可达对象的回收行为
157-内存溢出的分析
158-内存泄漏的分析
159-StopTheWorld 事件的理解
160-程序的并行与并发
161-垃圾回收的并行与并发
162-安全点与安全区域的说明
163-Java 中几种不同引用的概述
164-强引用：不回收
165-软引用：内存不足即回收
166-弱引用：发现即回收
167-虚引用：对象回收跟踪
168-终结器引用的介绍
169-垃圾回收器章节概览
170-垃圾回收器的分类
171-GC 性能指标的整体说明
172-吞吐量与暂停时间的对比说明
173-垃圾回收器的发展迭代史
174-垃圾回收器的组合关系
175-如何查看默认的垃圾回收器
176-Serial 与 Serial Old 垃圾回收器的介绍
177-如何设置使用 Serial 垃圾回收器
178-ParNew 垃圾回收器的介绍
179-如何设置使用 ParNew 垃圾回收器
180-Parallel 与 Parallel Old 垃圾回收器的介绍
181-Parallel 垃圾回收器的相关参数设置
182-CMS 垃圾回收器概述与工作原理
183-CMS 的特点与弊端分析
184-CMS 垃圾回收器的参数设置
185-CMS 的小结及后续 JDK 版本中的变化
186-认识 G1 垃圾回收器
187-G1 垃圾回收器的优势和不足
188-G1 的参数设置
189-G1 在生产环境的适用场景
190-region 的使用介绍
191-G1 垃圾回收器的主要回收环节
192-记忆集与写屏障

193-G1 垃圾回收过程的详细说明
194-G1 垃圾回收的优化建议
195-7 种经典的垃圾回收器总结与调优建议
196-常用的显示 GC 日志的参数
197-GC 日志中垃圾回收数据的分析
198-举例说明日志中堆空间数据如何解读
199-日志分析工具的使用
200-新时期的 Epsilon 和 Shenandoah 垃圾回收器
201-革命性的 ZGC 的性能介绍
202-其他的厂商的垃圾回收器
203-最后寄语

注意 !!! 这仅是上篇内容 !

中篇：字节码与类的加载篇

Class 文件结构

字节码指令集与解析

自定义类加载器

类加载的详细过程

下篇：性能监控与调优篇

基于 JDK 的命令行工具的监控及诊断

JConsole、VisualVM、MAT、JMC、JProfiler 等 GUI 工具的监控及诊断

内存分配与垃圾回收过程的参数调优

Java 代码层调优