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、调优系统瓶颈离不了对 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。

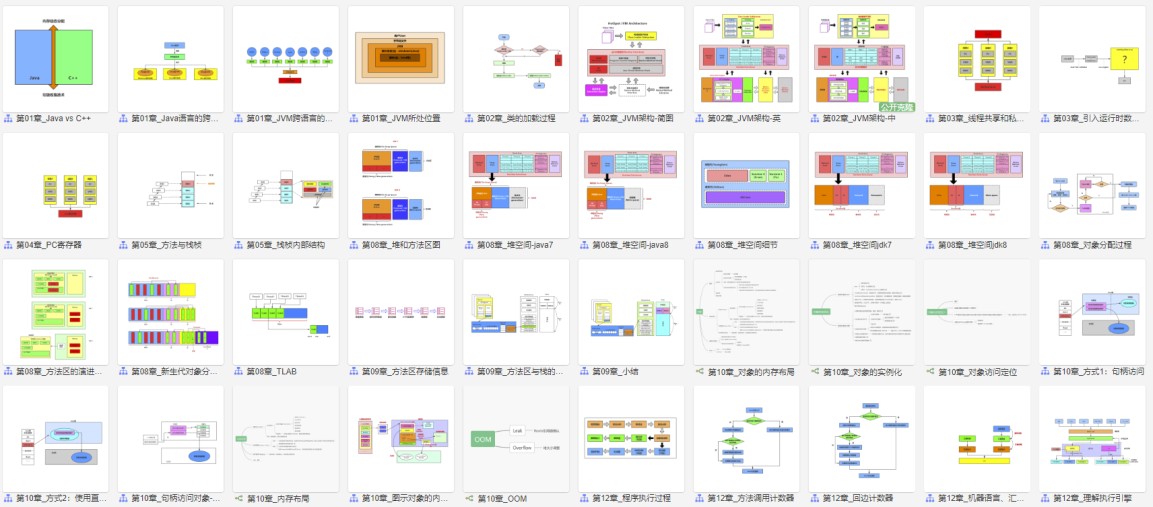
**我的课件里，有这样一页内容，需要可以用：** 作为 Java 工程师的你曾被 JVM 伤害过吗？

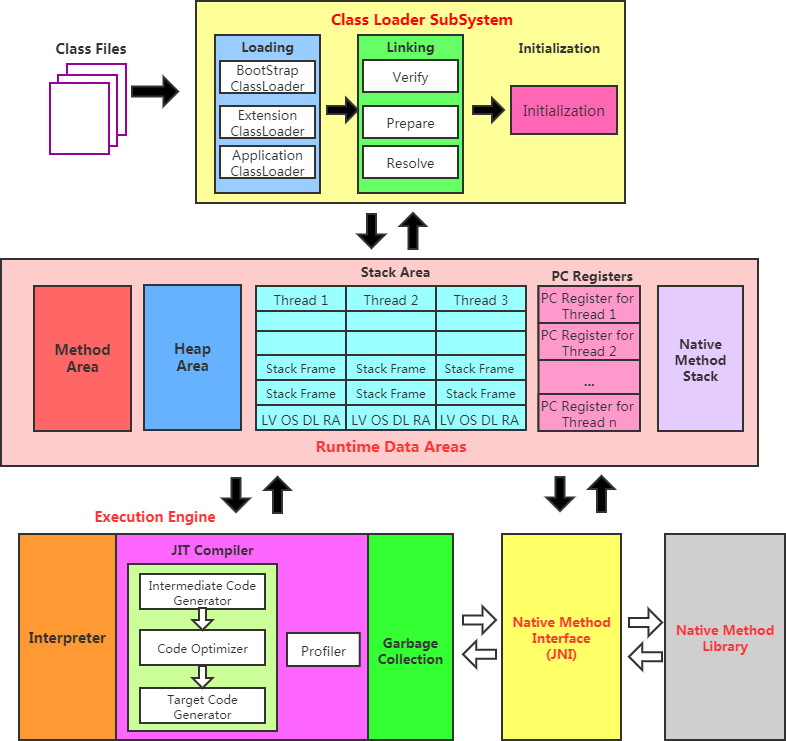
你是否也遇到过这些问题？

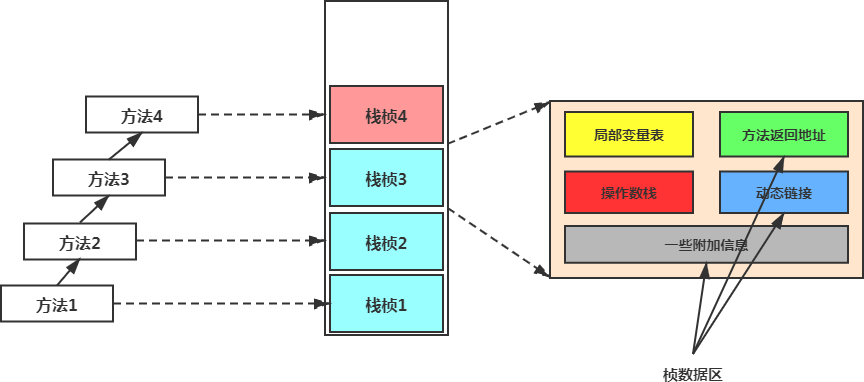
* 运行着的线上系统突然卡死，系统无法访问，甚至直接 OOM！
* 想解决线上 JVM GC 问题，但却无从下手。
* 新项目上线，对各种 JVM 参数设置一脸茫然，直接默认吧，然后就 JJ

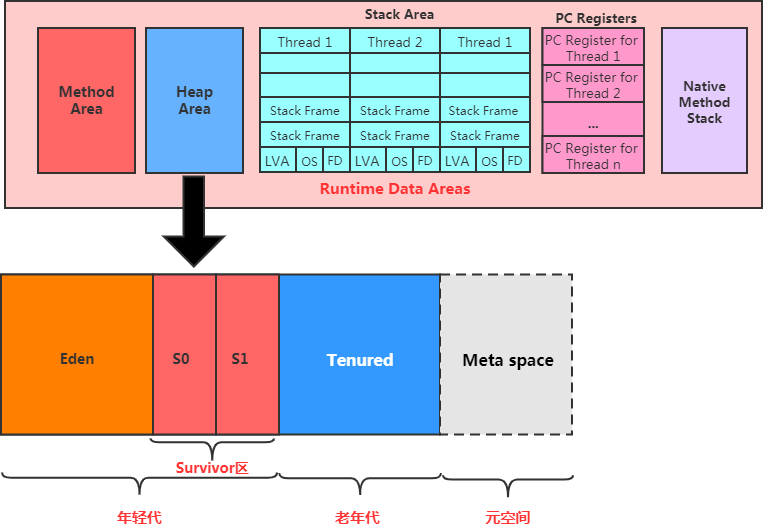
了

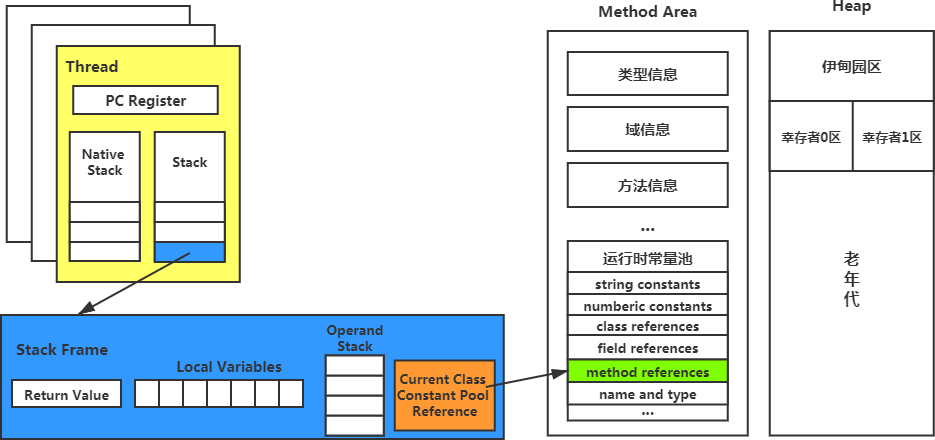
* 每次面试之前都要重新背一遍 JVM 的一些原理概念性的东西，然而面试官却经常问你在实际项目中如何调优 JVM 参数，如何解决 GC、OOM 等问题，一脸懵逼。

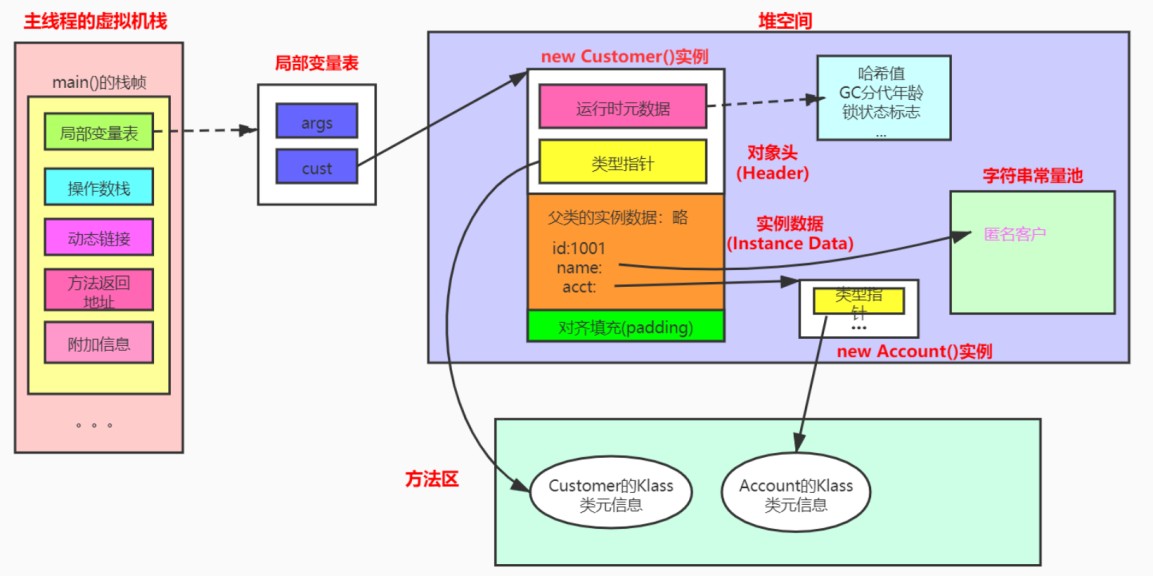


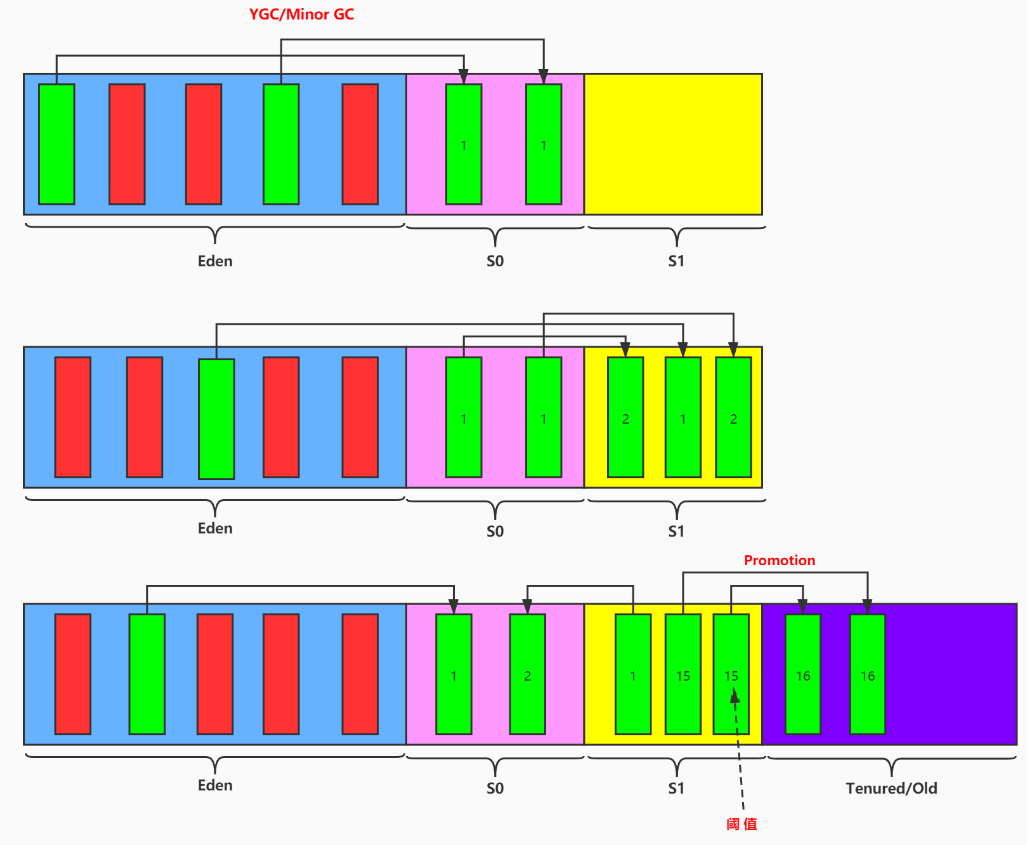


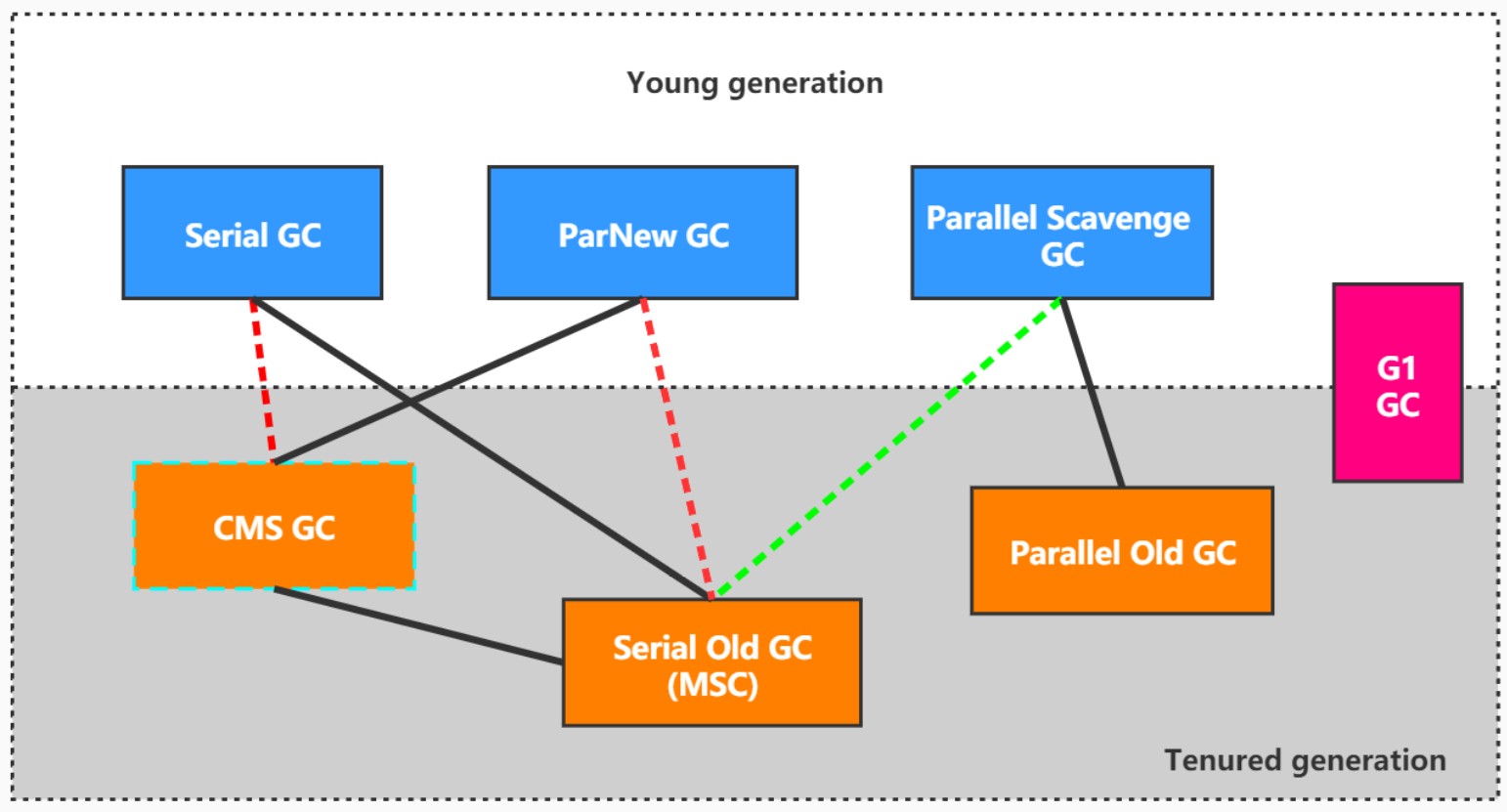


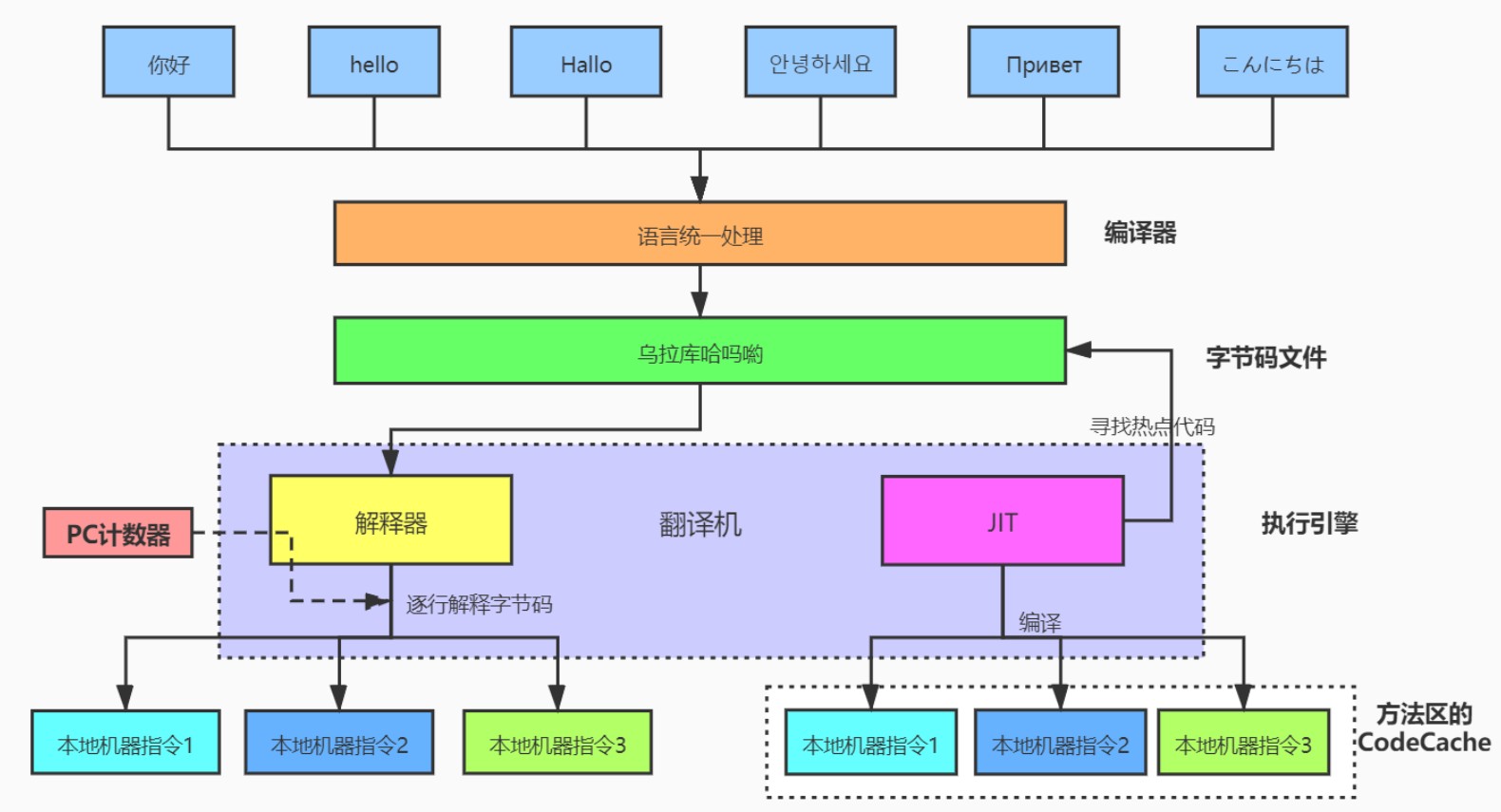


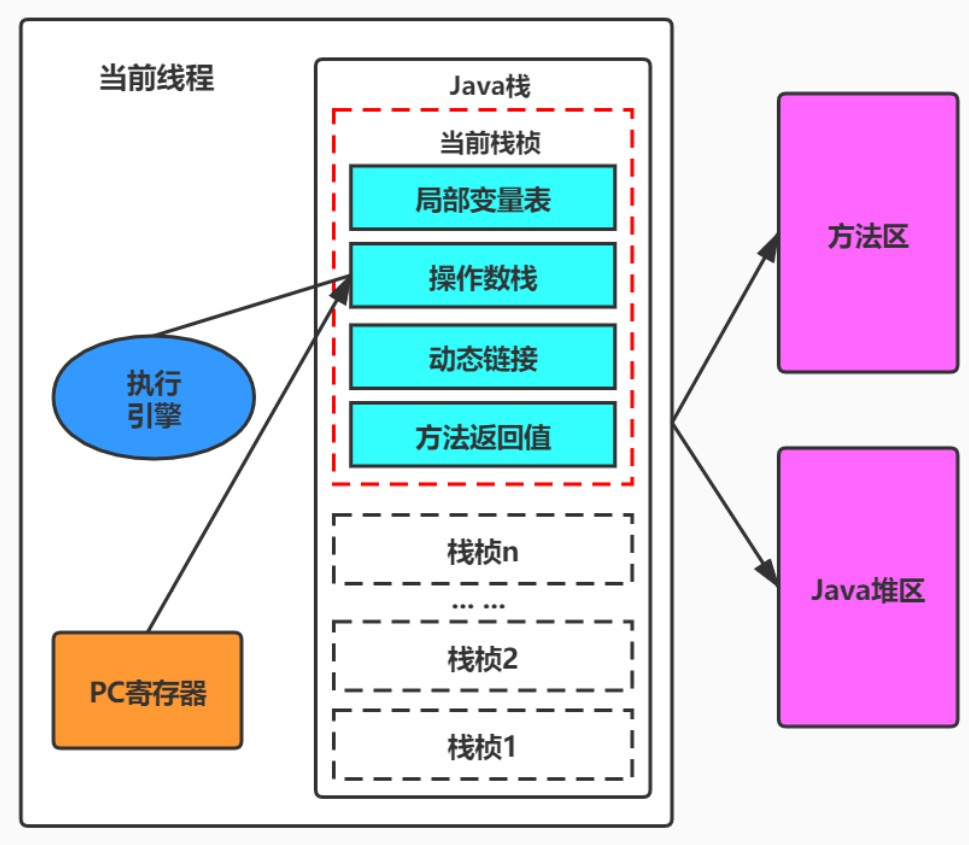












下面是之前已经发布的上篇内容：

1. JVM 内存与垃圾回收篇概述
2. 如何看待 Java 上层技术与 JVM
3. 为什么要学习 JVM
4. 面向人群和课程特点
5. 官方规范下载与参考书目
6. 跨平台的语言 Java 和跨语言的平台JVM
7. 字节码与多语言混合编程
8. Java 及 JVM 历史上的重大事件
9. 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
17. KVM、CDC、CLDC 的介绍
18. Azul VM 和 BEA Liquid VM 的介绍
19. Apache Harmony 的介绍
20. Microsoft JVM 和 TaobaoJVM 24-Dalvik VM 及其他虚拟机的介绍25-Graal VM 的介绍
21. 内存结构概述
22. 概述类的加载器及类加载过程28-类的加载过程一：Loading
23. 类的加载过程二：Linking
24. 类的加载过程三：Initialization 31-几种类加载器的使用体会
25. 引导类、扩展类、系统类加载器的使用及演示
26. 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
27. 双亲委派机制的工作原理及演示
28. 双亲委派机制的优势
29. 沙箱安全机制
30. 类的主动使用与被动使用等
31. 运行时数据区内部结构
32. JVM 中的线程说明
33. PC 寄存器概述
34. PC 寄存器的使用举例
35. 解决 PC 寄存器两个面试问题
36. 虚拟机栈的主要特点
37. 虚拟机栈的常见异常与如何设置栈大小
38. 栈的存储结构和运行原理
39. 栈桢的内部结构
40. 局部变量表结构的认识
41. 字节码中方法内部结构的剖析
42. 变量槽 slot 的理解与演示
43. 静态变量与局部变量的对比及小结
44. 操作数栈的特点
45. 涉及操作数栈的字节码指令执行分析
46. 栈顶缓存技术
47. 动态链接的理解与常量池的作用
48. 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

1. invokedynamic 指令的使用
2. 方法重写的本质与虚方法表的使用
3. 方法返回地址的说明
4. 栈桢中的一些附加信息
5. 虚拟机栈的 5 道面试题
6. 本地方法接口的理解
7. 本地方法栈的理解

本次发布的内容：

1. JVM 学习路线与内容回顾
2. 堆空间的概述\_进程中堆的唯一性
3. 堆空间关于对象创建和和 GC 的概述
4. 堆的细分内存结构
5. 堆空间大小的设置和查看
6. OOM 的说明与举例
7. 新生代与老年代中相关参数的设置
8. 图解对象分配的一般过程
9. 对象分配的特殊情况
10. 代码举例与 JVisualVM 演示对象的分配过程
11. 常用优工具概述与 Jprofiler 的演示 76-MinorGC、MajorGC 和 FullGC 的对比 77-GC 举例与日志分析
12. 体会堆空间分代的思想
13. 总结内存分配策略
14. 堆空间为每个线程分配的 TLAB
15. 小结堆空间的常用参数设置
16. 通过逃逸分析看堆空间的对象分配策略
17. 代码优化之栈上分配
18. 代码优化之同步省略
19. 代码优化之标量替换
20. 代码优化及堆的小结
21. 方法区的垃圾回收行为
22. 运行时数据区的总结与常见大厂面试题说明
23. 方法区概述\_栈堆方法区间的交互关系
24. 方法区的基本理解
25. Hotspot 中方法区的演进
26. 设置方法区大小的参数
27. OOM：PermGen 和OOM：Metaspace 举例
28. 方法区的内部结构 1
29. 方法区的内部结构 2
30. class 文件中常量池的理解
31. 运行时常量池的理解
32. 图示举例方法区的使用
33. 方法区在 jdk6、jdk7、jdk8 中的演进细节
34. StringTable 为什么要调整位置
35. 如何证明静态变量存在哪
36. 对象实例化的几种方式
37. 字节码角度看对象的创建过程
38. 对象创建的六个步骤
39. 对象的内存布局
40. 对象访问定位
41. 直接内存的简单体验
42. 使用本地内存读写数据的测试
43. 直接内存的 00M 与内存大小的设置
44. 执行引擎的作用及工作过程概述
45. Java 程序的编译和解释运行的理解
46. 机器码\_指令\_汇编\_高级语言理解与执行过程
47. 解释器的使用
48. HotspotVM 为何解释器与 JIT 编译器并存
49. 热点代码探测确定何时 JIT
50. Hotspot 设置模式\_C1 与 C2 编译器
51. Graal 编译器与 AOT 编译器
52. String 的不可变性
53. String 底层 Hashtable 结构的说明
54. String 内存结构的分配位置
55. 两个案例熟悉 String 的基本操作
56. 字符串拼接操作的面试题讲解
57. 字符串变量拼接操作的底层原理
58. 拼接操作与 append 操作的效率对比
59. intern()的理解
60. 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

1. StringTable 的垃圾回收测试
2. G1 垃圾收集器的 String 去重操作
3. 垃圾回收相关章节的说明
4. 什么是 GC，为什么需要 GC
5. 了解早期垃圾回收行为
6. Java 自动内存管理介绍
7. 垃圾回收相关算法概述
8. 引用计数算法的原理及优缺点
9. Java 代码举例\_Python 的引用计数实施方案
10. 可达性分析算法与 GC Roots 142-对象的 finalization 机制
11. 代码演示可复活的对象
12. 使用 MAT 查看 GC Roots
13. 使用 JProfiler 进行 GC Roots 溯源
14. 使用 JProfiler 分析 OOM
15. 标记-清除算法原理及优缺点
16. 复制算法原理及优缺点
17. 标记-压缩算法原理及优缺点
18. 不同指标上对比三种算法
19. 分代收集算法的说明
20. 增量收集算法原理及优缺点
21. 分区算法的说明
22. 垃圾回收相关概念的概述155-System.gc()的理解
23. 手动 gc 理解不可达对象的回收行为
24. 内存溢出的分析
25. 内存泄漏的分析
26. StopTheWorld 事件的理解
27. 程序的并行与并发
28. 垃圾回收的并行与并发
29. 安全点与安全区域的说明
30. Java 中几种不同引用的概述
31. 强引用：不回收
32. 软引用：内存不足即回收
33. 弱引用：发现即回收
34. 虚引用：对象回收跟踪
35. 终结器引用的介绍
36. 垃圾回收器章节概览
37. 垃圾回收器的分类
38. GC 性能指标的整体说明
39. 吞吐量与暂停时间的对比说明
40. 垃圾回收器的发展迭代史
41. 垃圾回收器的组合关系
42. 如何查看默认的垃圾回收器
43. Serial 与 Serial Old 垃圾回收器的介绍
44. 如何设置使用 Serial 垃圾回收器
45. ParNew 垃圾回收器的介绍
46. 如何设置使用 ParNew 垃圾回收器
47. Parallel 与Parallel Old 垃圾回收器的介绍
48. Parallel 垃圾回收器的相关参数设置
49. CMS 垃圾回收器概述与工作原理
50. CMS 的特点与弊端分析
51. CMS 垃圾回收器的参数设置
52. CMS 的小结及后续 JDK 版本中的变化
53. 认识 G1 垃圾回收器
54. G1 垃圾回收器的优势和不足
55. G1 的参数设置
56. G1 在生产环境的适用场景
57. region 的使用介绍
58. G1 垃圾回收器的主要回收环节
59. 记忆集与写屏障
60. G1 垃圾回收过程的详细说明
61. 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

1. 常用的显示 GC 日志的参数
2. GC 日志中垃圾回收数据的分析
3. 举例说明日志中堆空间数据如何解读
4. 日志分析工具的使用
5. 新时期的 Epsilon 和 Shenandoah 垃圾回收器
6. 革命性的 ZGC 的性能介绍
7. 其他的厂商的垃圾回收器
8. 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**