# 尚硅谷宋红康JVM全套教程（详解java虚拟机）

B站：<https://www.bilibili.com/video/BV1PJ411n7xZ?spm_id_from=333.1007.top_right_bar_window_history.content.click&vd_source=c93327379747783f97e9bff6769f415f>

目录

[尚硅谷宋红康JVM全套教程（详解java虚拟机） 1](#_Toc109511479)

[01-JVM内存与垃圾回收篇概述 2](#_Toc109511480)

[02-如何看待上层技术与JVM 3](#_Toc109511481)

[开发遇到如下问题 3](#_Toc109511482)

[我们写的应用程序 3](#_Toc109511483)

[架构师每天都在思考什么 3](#_Toc109511484)

[知乎上有条帖子：应该如何看招聘信息，直通年薪50万+？ 3](#_Toc109511485)

[03-为什么要学习JVM 4](#_Toc109511486)

[为什么学习jvm 4](#_Toc109511487)

[Java vs c++ 4](#_Toc109511488)

[04-面向人群和课程特点 5](#_Toc109511489)

[05-官方规范下载与参考书目 5](#_Toc109511490)

[06-跨平台的语言java和跨语言的平台jvm 7](#_Toc109511491)

[跨平台的语言java 7](#_Toc109511492)

[跨语言的平台jvm(其它语言经过编译后，也能在java虚拟机运行，只关心字节码文件) 8](#_Toc109511493)

[07-字节码与多语言混合编程 8](#_Toc109511494)

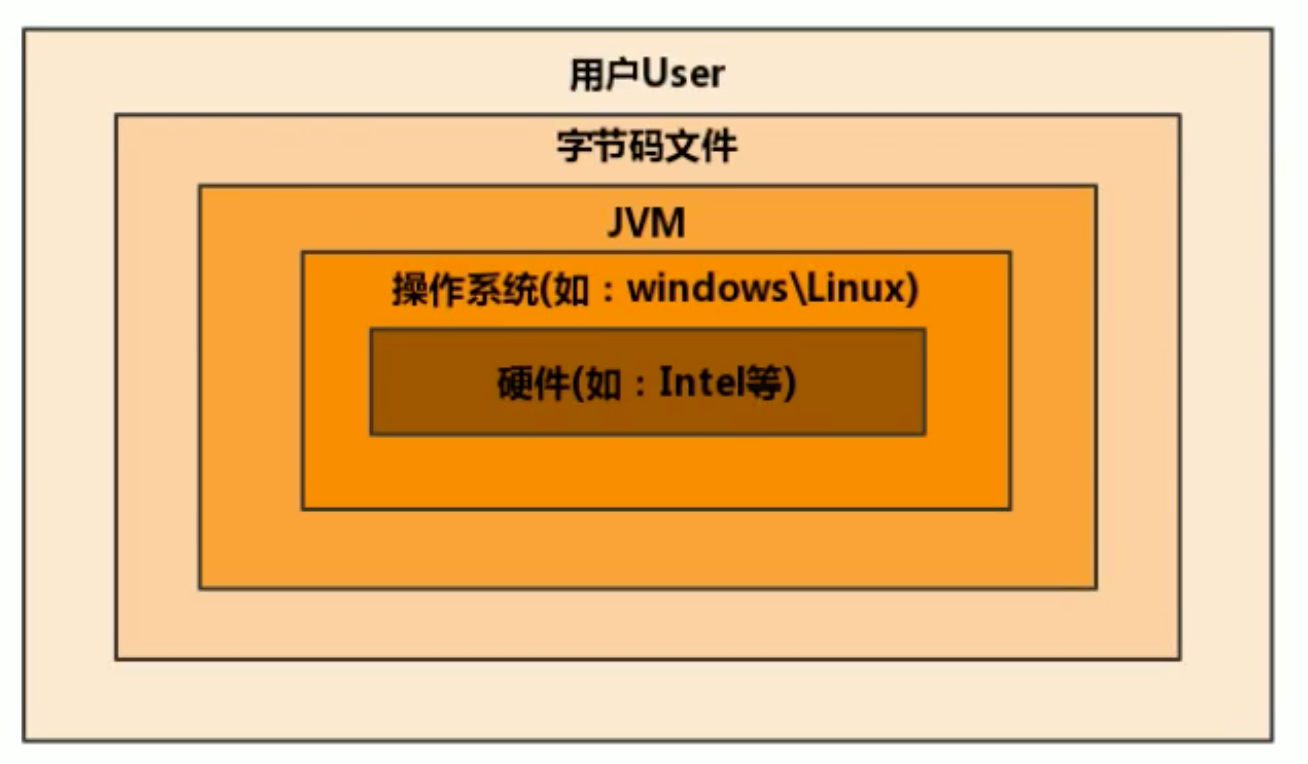
[08-java及JVM历史上的重大事件 9](#_Toc109511495)

[09-虚拟机和java虚拟机介绍 9](#_Toc109511496)

[虚拟机概念 9](#_Toc109511497)

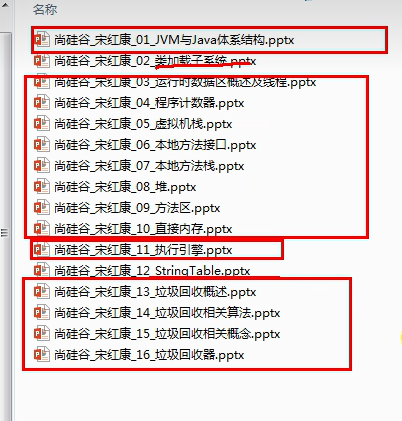
[Java虚拟机 10](#_Toc109511498)

[10-JVM的位置（JVM是运行在操作系统之上的，它与硬件没有直接的交互） 10](#_Toc109511499)

[ 10](#_Toc109511500)

[11JVM整体结构 11](#_Toc109511501)

# 01-JVM内存与垃圾回收篇概述

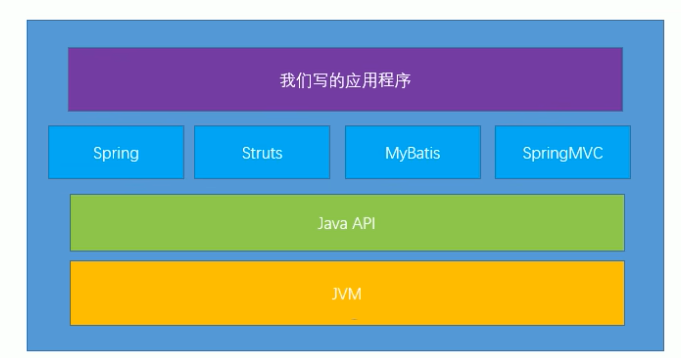


# 02-如何看待上层技术与JVM

## 开发遇到如下问题

* 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参数设置一脸懵逼，直接默认吧，然后就GG
* 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，然后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

## 我们写的应用程序



## 架构师每天都在思考什么

* 应该如何让我的系统更快？
* 如何避免系统出现瓶颈？

## 知乎上有条帖子：应该如何看招聘信息，直通年薪50万+？

* + 参与现有系统的性能优化，重构，保证平台性能和稳定性
  + 根据业务场景和需求，决定技术方向，做技术选型
  + 能够独立架构和设计海量数据下高并发分布式解决方案，满足功能和非功能需求
  + 解决各类潜在系统风险，核心功能的架构与代码编写
  + 分析系统瓶颈，解决各种疑难杂症，性能调优等

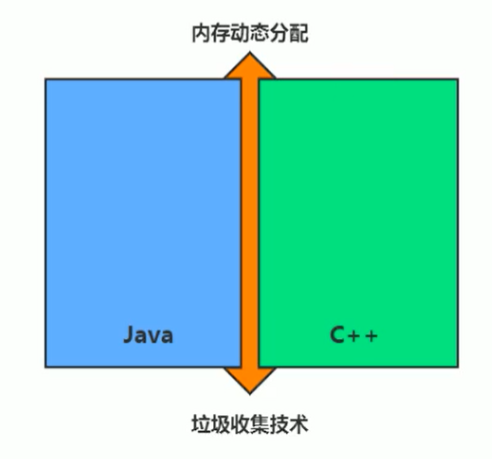
# 03-为什么要学习JVM

## 为什么学习jvm

* 面试的需要（BATJ、TMD，PKQ等面试都爱问）
* 中高级程序员必备技能：项目管理、调优的需求
* 追求极客的精神，比如：垃圾回收算法、JIT（即时编译器）、底层原理

## Java vs c++

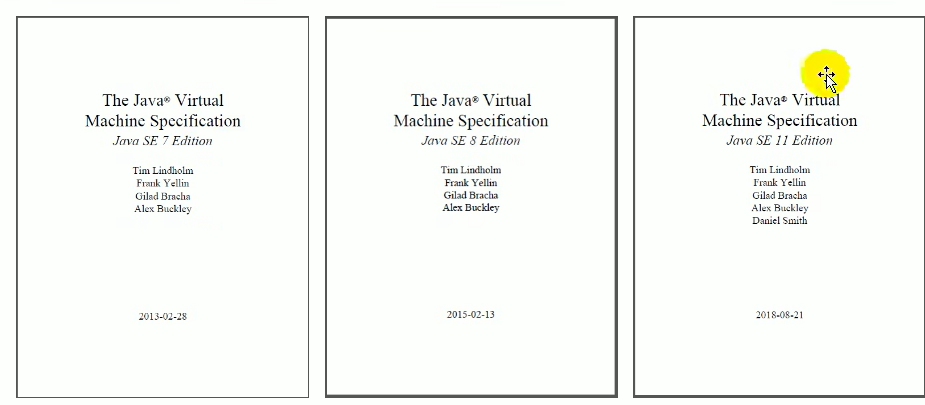
* + - 垃圾收集机制为我们打理了很多繁琐的工作，大大提高了开发的效率，但是，垃圾收集也不是万能的，懂得JVM内部的内存结构、工作机制，是设计高扩展性应用和诊断运行时问题的基础，也是Java工程师进阶的必备能力。
    - C语言需要自己来分配内存和回收内存，Java全部交给JVM进行分配和回收。

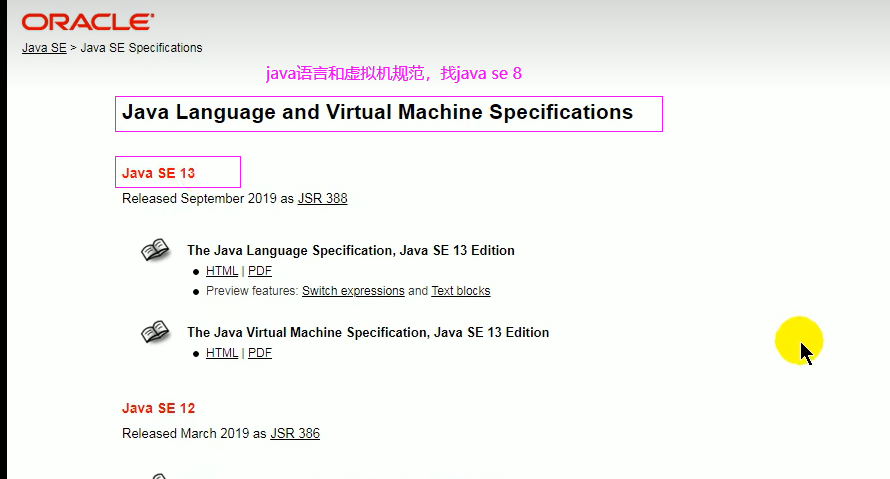


# 04-面向人群和课程特点

通俗易懂说人话+图解

# 05-官方规范下载与参考书目







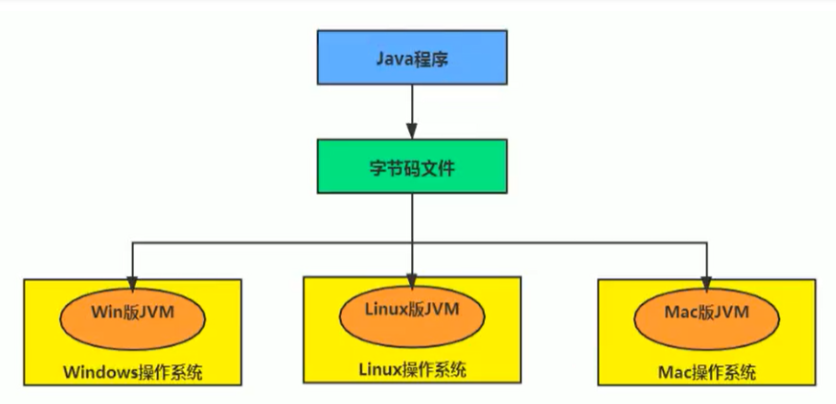




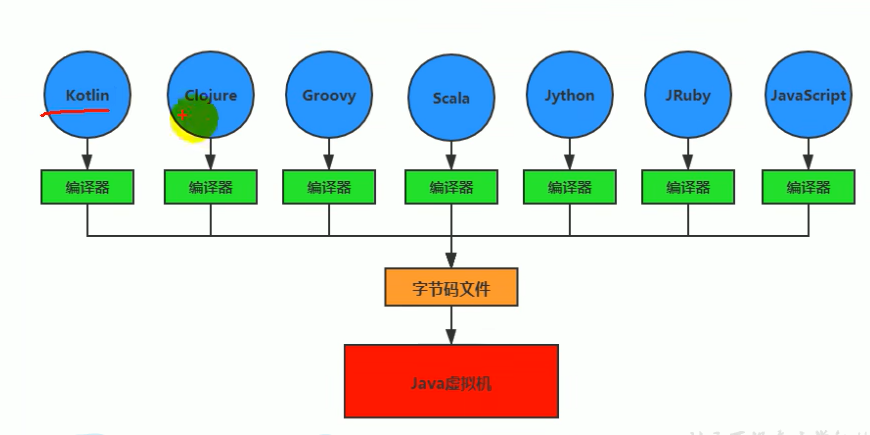
# 06-跨平台的语言java和跨语言的平台jvm

互联网基于js，人工智能基于python，微服务基于go

## 跨平台的语言java



## 跨语言的平台jvm(其它语言经过编译后，也能在java虚拟机运行，只关心字节码文件)

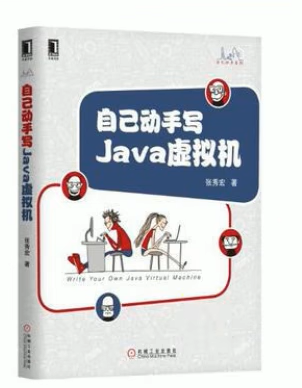


Java不是最强大的语言，但是jvm是最强大的虚拟机

# 07-字节码与多语言混合编程

Cpu+操作系统+编译器 三大难题

Java虚拟机从java语言的虚拟机向多语言虚拟机方向发展



Go语言写的java虚拟机，垃圾回收用的go语言，go具有完整的垃圾回收机制，但是c语言需要自己去管理内存。

# 08-java及JVM历史上的重大事件

# 09-虚拟机和java虚拟机介绍

## 虚拟机概念

所谓虚拟机（Virtual Machine），就是一台虚拟的计算机。它是一款软件，用来执行一系列虚拟计算机指令。大体上，虚拟机可以分为系统虚拟机和程序虚拟机。

大名鼎鼎的Virtual Box，VMware就属于系统虚拟机，它们完全是对物理计算机的仿真，提供了一个可运行完整操作系统的软件平台。

程序虚拟机的典型代表就是Java虚拟机，它专门为执行单个计算机程序而设计，在Java虚拟机中执行的指令我们称为Java字节码指令。

无论是系统虚拟机还是程序虚拟机，在上面运行的软件都被限制于虚拟机提供的资源中。

## Java虚拟机

Java虚拟机是一台执行Java字节码的虚拟计算机，它拥有独立的运行机制，其运行的Java字节码也未必由Java语言编译而成。

JVM平台的各种语言可以共享Java虚拟机带来的跨平台性、优秀的垃圾回器，以及可靠的即时编译器。

Java技术的核心就是Java虚拟机（JVM，Java Virtual Machine），因为所有的Java程序都运行在Java虚拟机内部。

Java虚拟机就是二进制字节码的运行环境，负责装载字节码到其内部，解释/编译为对应平台上的机器指令执行。每一条Java指令，Java虚拟机规范中都有详细定义，如怎么取操作数，怎么处理操作数，处理结果放在哪里。

特点

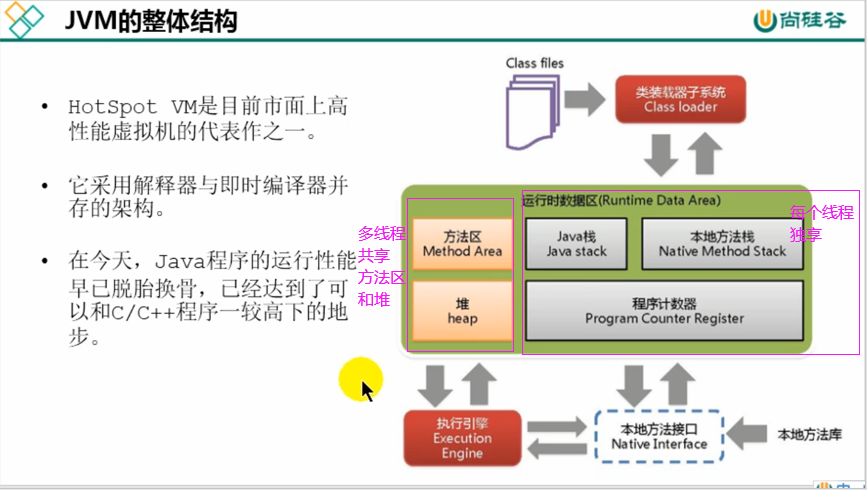
1. 一次编译，到处运行
2. 自动内存管理
3. 自动垃圾回收功能

# 10-JVM的位置（JVM是运行在操作系统之上的，它与硬件没有直接的交互）

# image-20201118090425668

系统虚拟机是对硬件的模拟，程序虚拟机是基于操作系统上解释运行某个程序我们需要的虚拟机环境

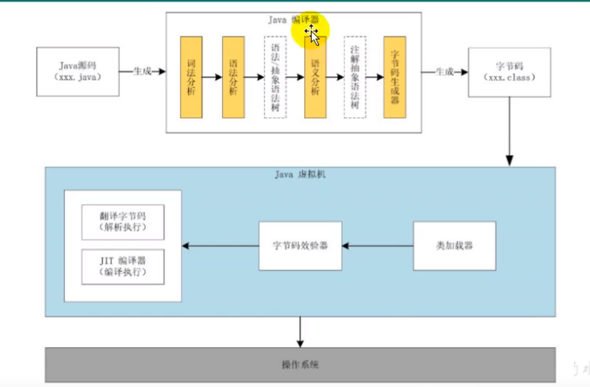
# 11JVM整体结构



解释器：逐行解释运行，保证即时响应。

即时编译器：反复执行的热点代码缓存起来放到方法区

# 12java代码执行流程

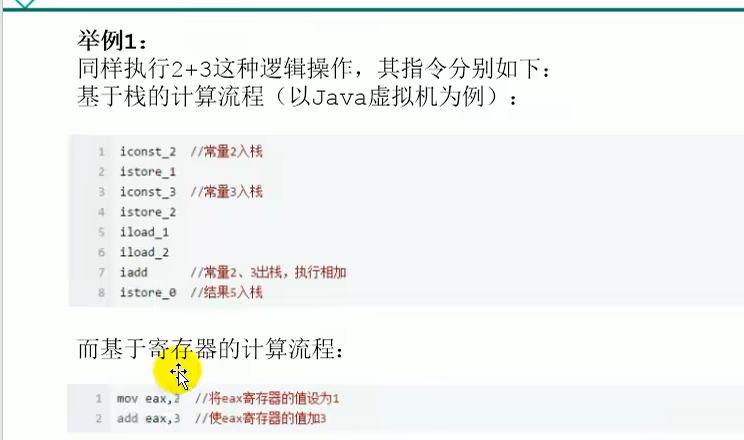


# 13.JVM的架构模型

# 

零地址指令是只有值没有地址，一地址指令是有一个地址有一个值。

栈是内存级别的不需要跟硬件大交道，可移植性性更好，更好实现跨平台。



从指令上来讲，基于栈的指令更多，寄存器的更少。但栈的指令集小是8为的指令集，寄存器是16位的指令集。