**jvm相关**

# java类从编译到运行完整过程是怎么样的：

编译、加载、运行

# 什么是类加载器？

Bootstrap根类加载器、Extensions扩展类加载器、application系统类加载器

# 类装载流程是怎样的？

加载、连接、初始化(类的生命周期从类被加载、连接和初始化开始，到类被卸载结束)

# 类加载双亲委派机制

类加载器用来把类的class文件加载到Java虚拟机中（内存中）。从jdk1.2开始，类的加载过程采用父亲委托机制，这种机制能更好地保证Java平台的安全。在此委托机制中，除了JVM自带的根加载器以外，其余的类加载器有且只有一个父加载器。当Java程序请求加载器loader1加载A类时，loader1首先委托自己的父加载器去加载A类，若父加载器能加载，则由父加载器完成加载任务，否则才由加载器loader1本身加载A类。

# jvm内存逻辑模型

栈、堆、方法区、程序计数器、本地方法区

堆(heap)：是java虚拟机所管理的内存中最大的一块内存区域，也是被各个线程共享的内存区域，该内存区域存放了对象实例及数组（但不是所有的对象实例都在堆中）

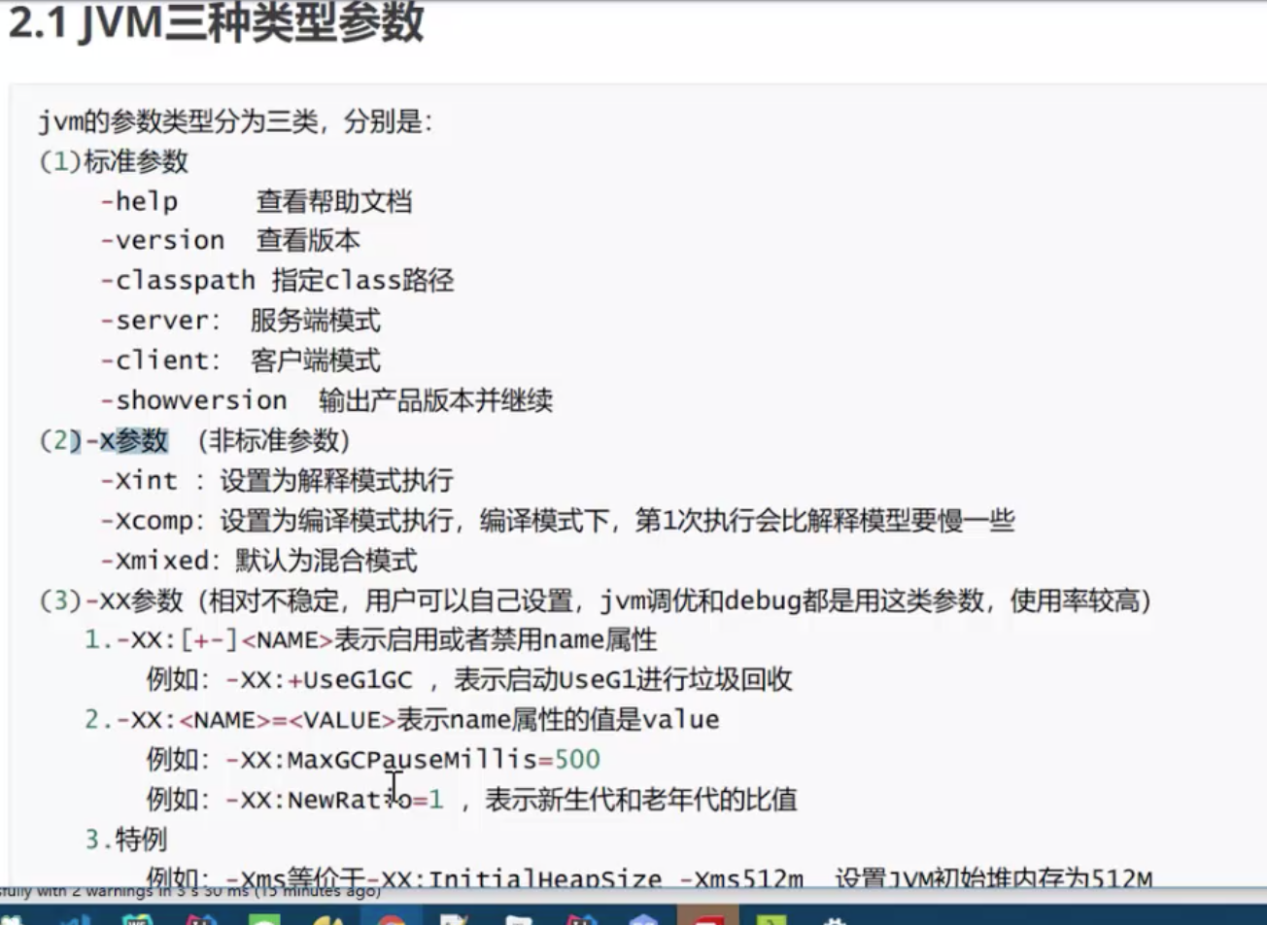
方法区(Method Area):方法区也称"永久代"，它用于存储虚拟机加载的类信息、常量、静态变量、是各个线程共享的内存区域

虚拟机栈(JVM Stack):描述的是java方法执行的内存模型：每个方法被执行的时候都会创建一个"栈帧",用于存储局部变量表(包括参数)、操作栈、方法出口等信息。

本地方法栈(Native Stack):与虚拟机栈基本类似，区别在于虚拟机栈为虚拟机执行的java方法服务，而本地方法栈则是为Native方法服务。(栈的空间大小远远小于堆)

程序计数器(PC Register):是最小的一块内存区域，它的作用是当前线程所执行的字节码的行号指示器，在虚拟机的模型里，字节码解释器工作时就是通过改变这个计数器的值来选取下一条需要执行的字节码指令，分支、循环、异常处理、线程恢复等基础功能都需要依赖计数器完成。

# jvm三种类型参数



# 垃圾回收

<https://www.cnblogs.com/sunshisonghit/p/6694590.html>