JVM

1. 编译器优化
   1. 公共子表达式的消除
   2. 指令重排
   3. 内联
   4. 逃逸分析（方法逃逸，线程逃逸）
   5. 栈上分配
   6. 同步消除
2. 类加载

类加载器

* 1. 双亲委派机制
  2. Bootstrap类加载机制
  3. Extension类加载机制
  4. System类加载器
  5. 自定义类加载器

1. GC
   1. 分代回收（年轻代，老年代，持久代）
   2. 串行回收器
   3. 并行回收器
   4. CMS
   5. G1
   6. ZGC
2. 内存模型

程序计数器 线程独占

本地方法栈

线程独占

栈 线程独占

方法区域 线程共享

堆 线程共享

* 1. 栈 (线程私有)

用来保存局部变量表，操作栈，动态链接，方法出口等信息，调用方法的时候执行入栈，调用返回的时候执行出栈

* 1. 本地方法栈(线程私有)

与java栈类似，执行本地方法的时候调用

* 1. 程序计数器(线程私有)

保存这当先线程执行的字节码的位置，每个线程工作的时候都有一个独立的计数器，只会执行java方法服务

* 1. 堆(线程共享)

堆是jvm管理中最大的一块，目的是为了存放对象实例，几乎所有的对象都放在这里，当堆内没有可用的内存的时候，会报出oom异常,根据jvm的生命周期不同，jvm对内存进行了分代管理，由垃圾回收器来进行垃圾回收管理

* 1. 方法区(线程共享)

非堆区用于存储虚拟机加载的类信息，常量，静态变量，即时编译器编译后的代码和数据

* 1. 类的加载与卸载

1. 性能调优
   1. JVM参数
   2. 性能分析工具（MAT,JMC,JStack,JStat）
2. 执行模式
   1. 解释模式
   2. 编译模式
   3. 回合模式