使用最广jvm sun hotspot vm

运行时数据区

1. 程序计数器：线程所执行的字节码的行号指示器（**线程私有**）
2. Java虚拟机栈：**线程私有，生命周期与线程相同**，每个方法会在其中创建**栈帧**存储：

**局部变量表**、操作数、动态链表、方法出口。其中局部变量包括存放**基本数据类型、对象引用、和returnaddress，**局部变量表所需的内存空间在**编译期间分配完成，其空间不会在运行期间改变。**

1. 本地方法栈，与2类似，不过是提供给native（jni）方法服务的
2. Java堆：所有线程共享，几乎所有对象和全部数组都要在堆上分配内存，垃圾管理器主要管理的也是堆
3. 方法区：**存储类信息、常量、静态变量、和jit代码**（堆的逻辑部分），其中的**运行常量池**存放编译期生成的字面量和符号引用（符号引用是一个字符串，它给出了被引用的内容的名字­­————此处指类名）（运行期也可能将常量放入池中）

对象的创建：

1. new之后根据类名去常量池定位到符号引用，并检查该引用是否被加载、解析初始化过、未加载则加载
2. 分配内存（空闲列表或指针碰撞）
3. 内存空间初始化为0
4. 虚拟机对对象设置（如对象是哪个类的实例），将信息存放在objectheader里
5. Init

Oom：（out of memory）

堆溢出：部分对象生命周期过长，导致它引用的对象也无法释放造成恶性拓展。

类加载过程

拓展栈时无法申请到足够空间：

Sof：（stack over flow）

栈深度大于虚拟机允许最大深度

Javac与jit

Javac是编译成字节码，运行时还要一句一句转义，jit是编译成机器码

线程状态

存储区域

垃圾回收（内存泄漏）

Nio

Classloader