**Hotspot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，像Groovy, Scala等语言生成的Java字节码也可以由Java虚拟机执行。**

**虚拟机运行时数据区**

**程序计数器：看做是当前线程所执行的字节码的行号指示器。**

**Java虚拟机栈：和程序计数器一样，Java 虚拟机栈（Java Virtual Machine Stacks）也是线程私有的。它的生命周期与线程相同。虚拟机栈描述的是Java 方法执行的内存模型：每个方法被执行的时候都会同时创建一个栈帧（Stack Frame ①）用于存储局部变量表、操作栈、动态链接、方法出口等信息。**

**局部变量表存放了编译期可知的各种基本数据类型（boolean、byte、char、short、int、float、long、double）、对象引用（reference 类型，它不等同于对象本身，根据不同的虚拟机实现，它可能是一个指向对象起始地址的引用指针，也可能指向一个代表对象的句柄或者其他与此对象相关的位置）和returnAddress 类型（指向了一条字节码指令的地址）。**

**本地方法栈：本地方法栈（Native Method Stacks）与虚拟机栈所发挥的作用是非常相似的，其区别不过是 虚拟机栈为虚拟机执行Java 方法（也就是字节码）服务，而本地方法栈则是为虚拟机使用到的Native 方法服务。**

**方法区：方法区（Method Area）与Java 堆一样，是各个线程共享的内存区域，它用于存储已被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等数据。用于保存系统的类信息，类的字段、方法、常量池等。**

**Java堆：Java 堆（Java Heap）是Java 虚拟机所管理的内存中最大的一块。Java 堆是被所有线程共享的一块内存区域，在虚拟机启动时创建。此内存区域的唯一目的就是存放对象实例，几乎所有的对象实例都在这里分配内存。**

**运行时常量池：**

**运行时常量池（Runtime Constant Pool）是方法区的一部分。Class 文件中除了有类的版本、字段、方法、接口等描述等信息外，还有一项信息是常量池（Constant PoolTable），**

**用于存放编译期生成的各种字面量和符号引用，这部分内容将在类加载后存放到方法区的运行时常量池中。**

**直接内存：直接内存（Direct Memory）并不是虚拟机运行时数据区的一部分，也不是Java虚拟机规范中定义的内存区域，但是这部分内存也被频繁地使用，而且也可能导致OutOfMemoryError 异常出现，所以我们放到这里一起讲解。在JDK 1.4 中新加入了NIO（New Input/Output）类，引入了一种基于通道（Channel）与缓冲区（Buffer）的I/O 方式，它可以使用Native 函数库直接分配堆外内存，然后通过一个存储在Java 堆里面的DirectByteBuffer 对象作为这块内存的引用进行操作。这样能在一些场景中显著提高性能，因为避免了在Java 堆和Native 堆中来回复制数据。显然，本机直接内存的分配不会受到Java 堆大小的限制，但是，既然是内存，则肯定还是会受到本机总内存（包括RAM 及SWAP 区或者分页文件）的大小及处理器寻址空间的限制。服务器管理员配置虚拟机参数时，一般会根据实际内存设置-Xmx等参数信息，但经常会忽略掉直接内存，使得各个内存区域的总和大于物理内存限制（包括物理上的和操作系统级的限制），从而导致动态扩展时出现OutOfMemoryError异常。**

**Java虚拟机的基本结构**

类加载子系统

Java堆

本地方法栈

直接内存

Java堆

方法区

PC寄存器

垃圾回收系统

**类加载子系统和方法区**

**类加载子系统负责从文件系统或者网络中加载Class信息，加载的类信息存放在一块方法区的内存空间。除了类信息外方法区可能还会存放运行时常量池信息。包括字符串字面量和数字字面量（这部分常量信息是Class文件中常量池部分的内存映射）**

**Java堆**

**Java堆在虚拟机启动时建立。他是Java程序最主要的内存工作区域。几乎所有的Java对象实例都存放在Java堆中。堆空间是所有线程共享的。**

**直接内存**

**直接内存实在Java堆外的、直接向系统申请的内存空间。通常，访问直接内存的速度会优于Java堆。因此出于性能考虑，读写频繁的场合可能会考虑使用直接内存。由于直接内存在堆外，因此它的大小不会直接受限于Xmx指定的最大堆大小，但是系统内存是有限的，Java堆和直接内存的总和依然受限于操作系统能给出的最大内存。**

**垃圾回收系统**

**垃圾回收器可以对方法区、Java堆和直接内存进行回收。完成包括Java堆、方法区和直接内存中的全自动化管理。**

**Java虚拟机栈：和程序计数器一样，Java 虚拟机栈（Java Virtual Machine Stacks）也是线程私有的。它的生命周期与线程相同。虚拟机栈描述的是Java 方法执行的内存模型：每个方法被执行的时候都会同时创建一个栈帧（Stack Frame ①）用于存储局部变量表、操作栈、动态链接、方法出口等信息。**

**函数如何调用：出入Java栈**

**线程执行的基本行为是函数调用，每次函数调用的数据都是通过Java栈传递的。每一次函数调用，都会有一个对应的栈帧被压入Java栈；每一个函数调用结束，都会有一个栈帧被弹出Java栈。**

**在一个栈帧中，至少要包含局部变量表、操作数栈和帧数据区几个部分。**