

- 运行时数据区

- Java虚拟机在执行Java的过程中，会把它所管理的内存划分为若干个不同的数据区域。

- 线程私有

- 程序计数器

- 程序计数器是一块较小的内存空间，它可以看作是当前线程所执行的字节码的行号指示器，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要执行的字节码指令，分支、循环、跳转、异常处理及线程恢复等功能都需要依赖这个计数器来完成。
- 由于Java虚拟机是痛过线程轮流切换并分配持利器执行时间的方式来实现的，因此，为了线程切换后能恢复到正确的位置，每条线程都需要一个独立的程序计数器，各条线程之间计数器互不影响，独立存储，我们将这类内存区域称为**线程私有**的内存。
- 如果线程正在执行的是一个Java方法，这个计数器记录的是正在执行的虚拟机字节码指令；如果正在执行的是Native方法，这个计数器值为空。此内存区域是**唯一一个**在Java虚拟机规范中**没有规定任何OutOfMemoryError**情况的区域

- Java虚拟机栈

- 生命周期与线程相同，它描述的是Java方法执行的内存模型：每个方法在执行的同时都会创建一个栈帧用于存储局部变量表、操作数栈、动态链接、方法出口等信息。每个方法从调用到执行完毕就对应一个栈帧在虚拟机中入栈到出栈的过程
- 我们常说的堆栈中的栈就是虚拟机栈
- 局部变量表存放了编译器可知的各种基本数据类型、对象引用和returnAddress(指向了一条字节码指令的地址)。其中64位长度的long和double类型的数据会占用2个局部变量空间，其余的数据类型占用1位局部变量表所需的内存空间在编译时期完成分配，方法运行期间不会改变局部变量表的大小
- Java虚拟机规范中，对这个区域规定了两种异常状况：如果线程请求的栈深度大于虚拟机所允许的深度，将抛出**StackOverflowError**异常；如果虚拟机可以动态扩展，但扩展时无法申请到足够的内存，就会抛出**OutOfMemoryError**异常

- 本地方法栈

- 与虚拟机栈发挥的作用相似，不同的是虚拟机栈为虚拟机执行Java方法(也就是字节码)服务，而本地方法栈则为虚拟机使用到的Native方法服务。与虚拟机栈一样，本地方法栈也会抛出StackOverflowError和OutOfMemoryError异常

- 线程共享

- Java堆

- 对于大多数应用来说，Java堆(Java Heap)是Java虚拟机所管理的内存中**最大**的一块。
- 在Java虚拟机启动的时候创建，此内存唯一的目的是**存放对象的实例**。在Java虚拟机规范中的描述是：**所有对象的实例及数组都要在堆上分配**。但随着JIT编译器的发展与逃逸分析技术的逐渐成熟，就不是那么绝对了

- Java堆是**垃圾收集器管理的主要区域**，因此很多时候也被称为“GC堆”。从内存回收的角度上看，由于现在的收集器基本都是采用分代算法，所有Java堆还可以分为：新生代和老年代。再细致点有Eden空间、From Survivor空间和To Survivor空间等。
- 从内存分配的角度上看，线程共享的Java堆中可以划分出很多线程私有的分配缓冲区。Java堆可以处于物理上不连续的内存空间中，只要逻辑上是连续的即可
- 方法区
 - 存储已被虚拟机加载的类信息、常量、静态变量和即时编译器编译后的代码等数据。
 - 此空间除了和Java堆一样不需要连续的内存和可以选择固定大小或者可扩展外，还可以选择不实现垃圾收集。但并不是数据进了方法区就永久存在了，**这个区域的内存回收目标主要是针对常量池的回收和对类型的卸载**。
 - 当方法区无法满足内存分配的需要时，将抛出**OutOfMemoryError**异常
- 运行时常量池
 - 方法区的一部分，Class文件中除了有类的版本、字段、方法和接口等描述信息外，还有一项信息是**常量池**，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进入方法区的运行时常量池中存放。
 - 一般来说，除了保存Class文件中描述的符号引用外，还会把翻译出来的直接引用也存储在运行时常量池中
 - 运行时常量池对于Class常量池的另外一个重要特征是具备**动态性**，运行期间也可能将新的常量放入池中，比如String类的intern()方法
 - 会抛出**OutOfMemoryError**异常
- 直接内存
 - 不是Java虚拟机规范中定义的部分，但是使用频繁且也可能抛出**OutOfMemoryError**异常
 - 在JDK1.4中新加入了NIO(New Input/Output)类，引入了一种基于通道与缓冲区的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堆中的DirectByteBuffer对象那个作为这块内存的引用进行操作。
 - 本地直接内存不受Java堆大小限制，但是受到本机总内存大小及处理器寻址空间的限制