1、内存划分

2、内存溢出和类型

3、GC垃圾回收

Jvm内存区域划分以及GC垃圾回收

Java虚拟机的内存根据用途被划分为：

程序计数器

java栈

本地方法栈

Java堆

方法区

如下图



这些区域都是用来存储什么呢

1、程序计数器

程序计数器的内存区域很小，用来记录当前线程正在执行的虚拟机字节码指令地址。是线程私有的，每个线程都有自己的程序计数器。

程序计数器对于线程有什么用呢，线程在时间片耗尽以后，会被挂起，再次运行的时候，就需要通过自己的程序计数器找到执行到那里了，

程序计数器只记录了一个值，所以这个区域不需要太大空间，程序计数器也是唯一没有内存溢出的区域。

2、java栈

Java栈主要服务于java方法，每个方法执行是都会创建一个栈帧，栈帧中存储变量表、操作数栈、动态链接、方法出口等等，java栈也是线程私有的，生命周期和线程一致。

1、局部变量表中存储了编译器可知的基本类型数据、对象引用和returnAddress类型。

通常说法，基本类型是存储在java栈中的，这不全对，这里的基本类型指的是局部变量中的基本类型（如，在方法中的临时变量），存储在堆中的对象的属性中的基本类型存储在堆中。

2、操作数栈，是FIFO（先入先出）的，方法执行过程中字节码指令被压入栈，字节码指令的写入和提取就是入栈和出栈操作。

3、动态链接，每个栈帧都包含一个指向运行时常量池中该栈帧所属方法的引用，持有这个引用是为了支持方法调用过程中的动态连接(Dynamic Linking)。个人理解，class文件的所有变量和方法引用都存储在方法区的常量池中，方法中调用方法或变量是，需要通过动态链接调用。

3、本地方法栈

本地方法栈和java栈作用差不多，不过服务的对象是native方法。

4、java堆

这是最大的一块内存区域，通常的内存主要就是堆，垃圾回收机制（GC）也是针对的堆。

Java线程共享堆内存，堆可以分为两部分：年轻代和年老代，年轻代又可以分为三部分：Eden、From Survivor、To Survivor。为什么要这么分会在GC部分具体细说

堆的大小可以通过参数 –Xms、-Xmx 来指定。

5、方法区

方法区也被成为永久代（通常不进行GC），方法区在线程中也是共享的。

方法区中存储了：

1、class的类信息

2、常量

3、静态变量

4、普通方法在编译器编译后生成的字节码

二、内存溢出

内存溢出包括哪些呢

1、堆溢出

2、方法区溢出

3、栈溢出

4、本地内存溢出