Java中堆栈详解

1.寄存器：最快的存储区，由编译器根据需要进行分配，我们在程序中无法控制。

2.栈：存放基本类型的变量数据和对象的引用，但对象本身不存在栈中，而是存入在堆（new出来的对象）或者常量池中（对象可能在常量池中）字符串常量对象存放在常量池中。  
3.堆：存放所有new出来的对象。

4.静态域：存入所有的静态成员（static定义的）。

5.常量池：存放字符串常量和基本类型的常量（public static final）。有时，在嵌入式系统中，常量本身会和其它部分分割离开，所以在这种情况下，可以选择将其放到ROM中。

6.非RAM存储，硬盘等永久存储空间。

注意：1.栈和常量池中的数据可以被共享，堆中的数据不能被共享；栈中的数据的大小和生命周期是可以确定的，当没有引用指向数据时，这个数据就会消失。堆中的数据则由GC负责回收，因此，大小和生命周期不需要确定。

2.如果是编译期已经创建好（直接用双引号定义的），就存储在常量池中；如果是运行期（new出来的）才能确定的就存储在堆中。对于equals相等的字符串，在常量池中永远只有一份，在堆中有多份。

3.对于new产生的一个字符串，会先去常量池中去寻找是否有这个对象，若是没有，则在常量池中创建一个此对象，然后再在堆中创建一个这个对象的拷贝对象；若是有，则直接在堆中创建一个这个对象的拷贝对象。

4.对于成员变量和局部变量：成员变量就是方法外部，类的内部定义的变量；局部变量就是方法或者语句块内定义的变量。局部变量必须初始化；

5.形式参数是局部变量，局部变量的数据存在于栈内存中。栈内存中的局部变量随着方法的消失面消失。成员变量存储在堆中的对象里面，由垃圾回收器进行回收。

6.形参也为局部变量。

JVM中的堆栈：

1.JVM是基于堆栈的虚拟机。JVM为每个新创建的线程都分配一个堆栈。也就是说，对于一个JAVA程序来说，它的运行就是通过对堆栈的操作来完成的。堆栈以帧为单位保存线程的有状态。JVM对堆栈只进行两种操作：以帧为单位的压栈和出栈操作。

2.我们知道，某个线程正在执行的方法称为此线程的当前方法，我们可能不知道，当前方法使用的帧称为当前帧。当线程激活一个Java方法，JVM就会在线程的Java堆栈里新压一个帧。这个帧自然成为了当前帧。在此方法执行期间，这个帧将用来保存参数，局部变量，中间计算过程和其它数据。这个帧在这里和编译原理中的活动纪录的概念是并差不多的。

3.从Java的这种分配机制来看，堆栈以可以这样理解：堆栈是操作系统建立在某个进程时或者线程（在支持多线程的操作系统中是线程）为这个线程建立的区域，该区域具有先进后出的特性。

4.每一个Java应用都唯一对应一个JVM实例，每一个实例唯一对应一个堆。应用程序在运行中所创建的所有的类实例或数组都放在这个堆中，并由应用所有的线程共享。跟C/C++不同，Java中分配堆内存是自动初始化的。Java中所有的对象存储空间都是在堆中分配的，但是这个对象的引用却是在堆栈中分配，也就是说在建立一个对象时从两个地方都分配内存，在堆中分配的内存实际建立这个对象，而在堆栈中的内存只是一个指向这个堆对象的（引用）而已。

JAVA堆栈：

1.栈与堆都是Java中用来在Ram中存放数据的地方。与C++不同，Java自动管理栈和堆，程序员不能直接地设置栈或堆。

2.Java的堆是一个运行时数据区，类的（对象从中分配空间。这些对象通过new、newarray、anewarray和multianewarray等指令建立）它们不需要程序代码来显示的释放。堆是由垃圾回收器来负责的，堆的优势是可以动态地分配内存大小，生存期也不必事先告诉编译器，因为它是在运行时动态分配内存的，Java的垃圾收集器会自动收走这些不再使用的数据，但缺点是，由于要在运行时动态的分配内存，存取速度慢。

3.栈的优势是，存取速度比堆要快，仅次于寄存器，栈数据可以共享。但缺点是，存在栈中的数据大小与生存期必须是确定的，缺乏灵活性。栈中主要存放一些基本类型的变量（int,short,logn,byte,float,double,Boolean,char）和对象句柄。

**总结：一个类文件首先被加载进方法区，一些符号引用被解析成直接引用或者等到分配时运行时分派，经过一系列的加载过程（class文件的常量池被加载到方法区的运行时常量池，各种其它的静态存储结构被加载为方法区运行时数据结构等）**

**然后程序通过class对象来访问方法区里面的各种类型数据，当加载完之后，程序发现了main方法，也就是程序入口，那么程序就在栈里面创建一个栈帧，逐行读取方法里的代码所转换成的指令，而这些指令大多已经被解析成直接引用了，那么程序通过持有这些直接引用使用指令方法去方法区中寻找变量对应的字面量来进行方法的操作。**

**操作完之后，方法返回给调用方，该栈帧出栈，内存空间被GC回收，堆里被new的那些也就被垃圾回收机制GC了。**

**全程包括以下几步：源码编写——编译（javac编译和jit编译，java语法糖）—— 类文件被加载到虚拟机（类class文件结构，虚拟机运行时内存分析，类加载机制）——虚拟机执行二进制字节码（虚拟机字节码执行系统）——垃圾回收（JVM垃圾回收机制）**