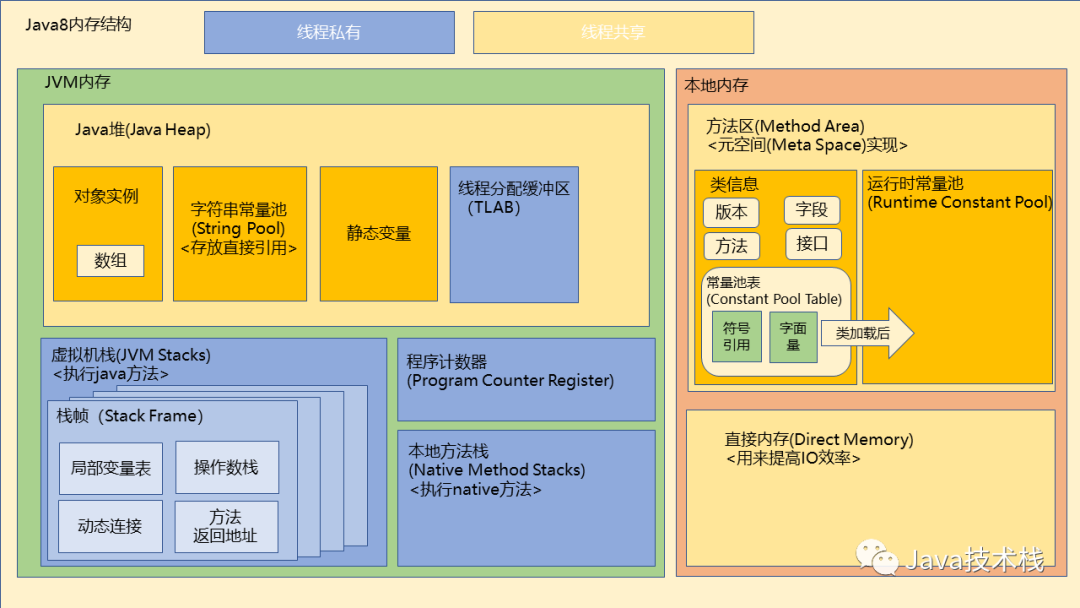
**Java 8 的内存结构**

<https://mp.weixin.qq.com/s/8uGOt1OJloZMHQl43vrbyQ>



## 虚拟机内存与本地内存的区别

Java虚拟机在执行的时候会把管理的内存分配成不同的区域，这些区域被称为虚拟机内存，同时，对于虚拟机没有直接管理的物理内存，也有一定的利用，这些被利用却不在虚拟机内存数据区的内存，我们称它为本地内存，这两种内存有一定的区别：

### [JVM内存](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)

* [受虚拟机内存大小的参数控制，当大小超过参数设置的大小时就会报OOM](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)

### [本地内存](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)

* [本地内存不受虚拟机内存参数的限制，只受物理内存容量的限制](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)
* [虽然不受参数的限制，但是如果内存的占用超出物理内存的大小，同样也会报OOM](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)

## java运行时数据区域

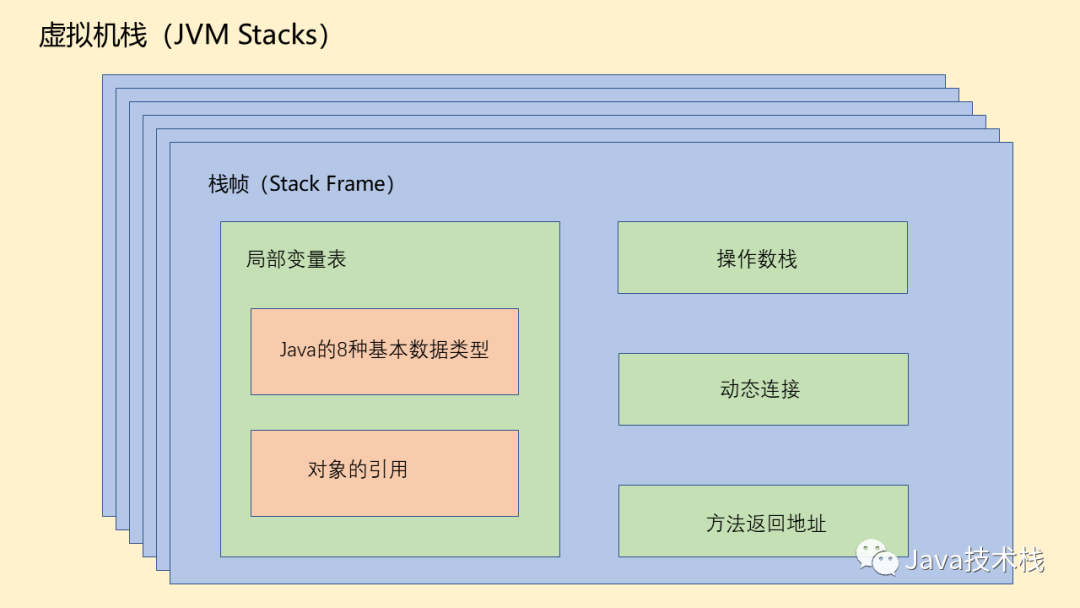
java虚拟机在执行过程中会将所管理的内存划分为不同的区域，有的随着线程产生和消失，有的随着java进程产生和消失，根据《Java虚拟机规范》的规定，运行时数据区分为以下一个区域：

### 程序计数器（Program Counter Register）

程序计数器就是当前线程所执行的字节码的行号指示器，通过改变计数器的值，来选取下一行指令，通过他来实现跳转、循环、恢复线程等功能。

* [在任何时刻，一个处理器内核只能运行一个线程，多线程是通过线程轮流切换，分配时间来完成的，这就需要有一个标志来记住每个线程执行到了哪里，这里便需要到了程序计数器。](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)
* [所以，程序计数器是线程私有的，每个线程都已自己的程序计数器。](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)

### 虚拟机栈（JVM Stacks）



**虚拟机栈是线程私有的，随线程生灭。虚拟机栈描述的是线程中的方法的内存模型：**

每个方法被执行的时候，都会在虚拟机栈中同步创建一个栈帧（stack frame）。

每个栈帧的包含如下的内容

* 局部变量表
* + 局部变量表中存储着方法里的java基本数据类型（byte/boolean/char/int/long/double/float/short）以及对象的引用（注：这里的基本数据类型指的是方法内的局部变量）
* 操作数栈
* 动态连接
* 方法返回地址

方法被执行时入栈，执行完后出栈

[**虚拟机栈可能会抛出两种异常：**](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)

* [如果线程请求的栈深度大于虚拟机所规定的栈深度，则会抛出StackOverFlowError即栈溢出](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)
* [如果虚拟机的栈容量可以动态扩展，那么当虚拟机栈申请不到内存时会抛出OutOfMemoryError即OOM内存溢出](http://mp.weixin.qq.com/s?__biz=MzI3ODcxMzQzMw==&mid=2247489332&idx=2&sn=65de5886e13b98116c8432d7d10ae4bc&chksm=eb539202dc241b14010f70edf89dc37c7629b5e2b7add50fd3f58070ecf2c14260196bf147d8&scene=21#wechat_redirect)

**本地方法栈（Native Method Stacks）**

本地方法栈与虚拟机栈的作用是相似的,都会抛出OutOfMemoryError和StackOverFlowError，都是线程私有的，主要的区别在于：

* 虚拟机栈执行的是java方法
* 本地方法栈执行的是native方法(什么是Native方法？)