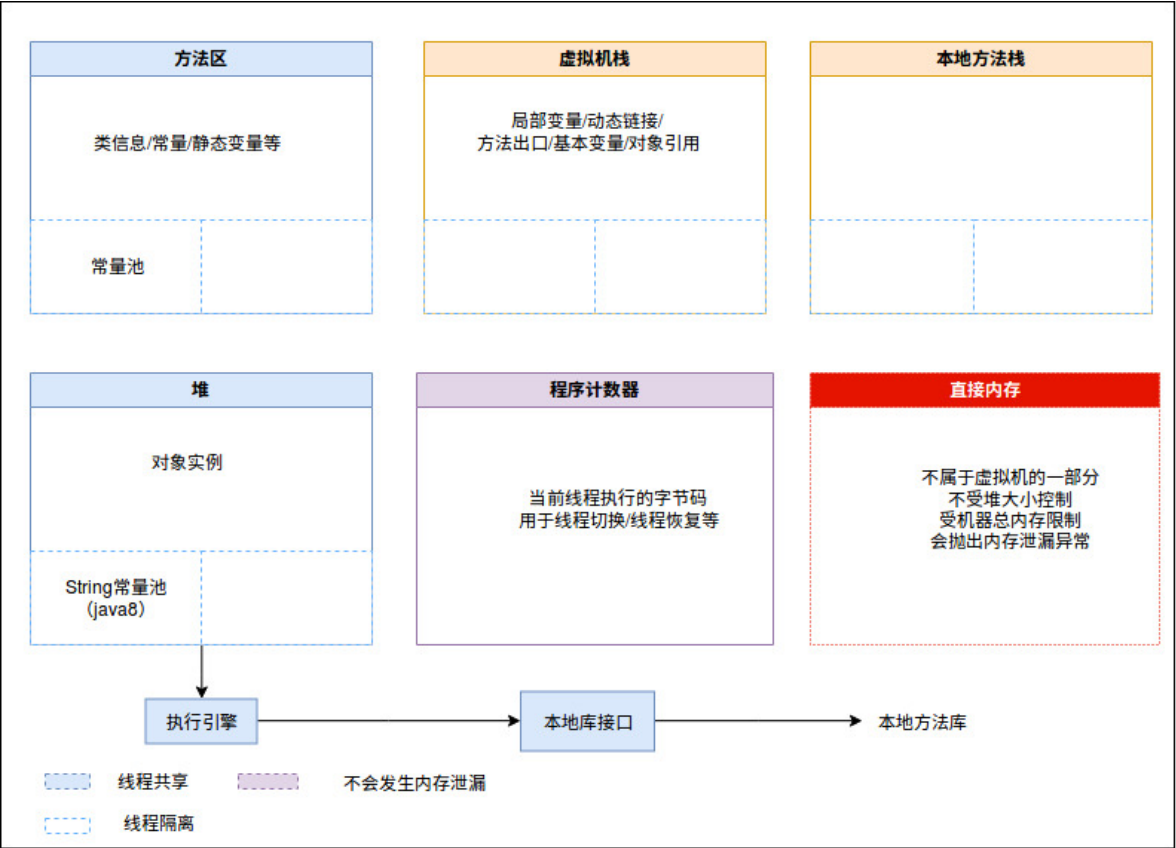
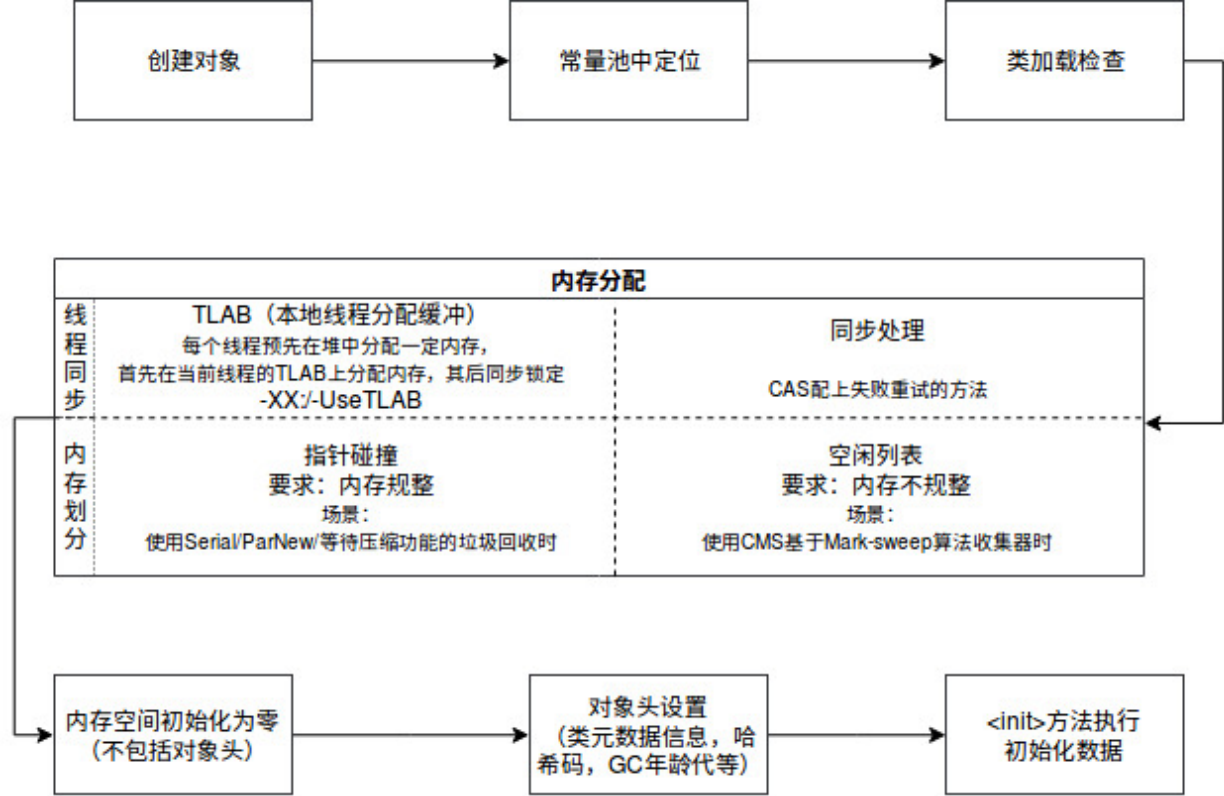


虚拟内存机制如图



虚拟机内存分配机制



GC算法

Gc详解					
------	--	--	--	--	--

		步骤	优点	缺点
Gc算法	标记-清除算法 (基础算法)	1: 标记需要回收对象 2: 清除被标记的对象		1效率问题，标记和清楚效率均不足 2: 空间问题。大量的空间碎片，导致在存储大对象时可能无法找到连续内存而触发GC
	复制算法	1: 内存分为两部分 (A, B) 2: A内存使用完，将存活的对象复制到B，并清除A	1: 不用考虑内存碎片	内存缩小一半
	标记整理算法	1: 标记需要回收对象 2: 存活对象向一端移动，清除掉端边界以外的内存	1: 没有内存碎片 2: 没有空间浪费 3: 没有较多的复制操作	
	分代收集算法			
HotSpot 算法实现		1: safePoint safe Region，当线程处于这两个状态时，认为引用关系不会变化，从而可以执行GC 2: 抢先式中断,GC中断所有线程，如果线程不处于安全状态，则恢复线程达到安全状态后中断 2: 主动式中断，GC设定标志，线程执行时查看标志，为真时中断		
垃圾收集器	Serial	1: 单线程收集器，		

(复制算法)	停止所有工作 2: 能与CMS协作		
ParNew收集器 (复制算法)	1: 多线程收集器, 停止所有线程 2: 能与CMS协作		
Paraller Scavenger (复制算法)	1: 并行收集器, 着重与吞吐量		
Serial old (标记整理算法)	1: 与Paraller Scavenger结合使用 2: 作为CMS收集器的后备		
Paraller old (标记整理算法)			
CMS (标记清除)	1: 初始标记, 标记GC root直接关联的对象 (停止所有线程) 2: 并发标记 3: 重新标记, 修正引用变化的部分	1: 最短停顿	1: cpu资源敏感 2: 会产生空间碎片 (导致full GC)

			(停止所有线程) 4: 并发清除 (与用户线程并发)		
		G1不成熟 不做考虑			

垃圾收集器协作

垃圾收集相关的常用参数

UseSerialGC	虚拟机运行在Client模式下的默认值，打开此开关后，使用Serial Old的收集器组合进行内存回收
UseParNewGC	打开此开关后，使用ParNew+Serial Old的收集器组合进行内存回收
UseConcMarkSweepGC	打开此开关后，使用ParNew+CMS+Serial Old的收集器组合进行内存回收，Concurrent Mode Failure失败后的后备收集器使用
UseParallelGC	虚拟机运行在Server模式下的默认值，打开此开关后，使用Parallel Scavenge + Parallel Old的收集器组合进行内存回收
UseParallelOldGC	打开此开关后，使用Parallel Scavenge + Parallel Old的收集器组合进行内存回收
SurvivorRatio	新生代中Eden区域与Survivor区域的容量比值，默认值为8，代表Eden区域是Survivor区域的8倍
PretenureSizeThreshold	直接晋升到老年代的对象大小，设置这个参数后，大于这个参数的对象在Minor GC时直接晋升到老年代
MaxTenuringThreshold	晋升到老年代的对象年龄，每个对象在坚持过一次Minor GC之后，年龄就会+1，当年龄达到这个参数时，对象就会晋升到老年代
UseAdaptiveSizePolicy	动态调整Java堆中各个区域的大小以及进入老年代的年龄
HandlePromotionFailure	是否允许分配担保失败，即老年代的剩余空间不足以应付新生代的空间需求时，是否允许部分对象晋升失败
ParallelGCThreads	设置并行GC时进行内存回收的线程数
GCTimeRatio	GC时间占总时间的比率，默认值为99，即允许1%的GC时间。100表示GC时间不能超过总时间的100%
MaxGCPauseMillis	设置GC的最大停顿时间，仅在使用Parallel Scavenge收集器时有效
CMSInitiatingOccupancyFraction	设置CMS收集器在老年代空间被使用多少后触发垃圾收集。默认值为0.95，即当老年代空间被使用到95%时触发垃圾收集

UseCMSCompactAtFullCollection: 设置CMS收集器在完成垃圾收集后是否要进行一次内存碎片整理, 仅在使用CMS收集器时生效

CMSFullGCsBeforeCompaction: 设置CMS收集器在进行若干次垃圾收集后再启动一次内存碎片整理。仅在使用CMS收集器时生效

内存分配

- 1: 优先在eden分配
- 2: 大对象直接进入老年代
- 3: 长期存活对象直接进入老年代
- 4: 相同年龄对象的大小综合占survivor空间的一半, 则大于等于此年龄的对象直接进入老年代