Atitit vm 内存管理 堆和方法区

堆和方法区

然后，在Java8中，时代变了，Hotspot取消了永久代。永久代真的成了永久的记忆。永久代的参数-XX:PermSize和-XX：MaxPermSize也随之失效。

元空间(Metaspace)

对于Java8，HotSpots取消了永久代，那么是不是就没有方法区了呢？当然不是，方法区只是一个规范，只不过它的实现变了。

在Java8中，元空间(Metaspace)登上舞台，方法区存在于元空间(Metaspace)。同时，元空间不再与堆连续，而且是存在于本地内存（Native memory）

本地内存（Native memory），也称为C-Heap，是供JVM自身进程使用的。当Java Heap空间不足时会触发GC，但Native memory空间不够却不会触发GC

永久代为什么被替换了

思考一下，为什么使用元空间替换永久代？

表面上看是为了避免OOM异常。因为通常使用PermSize和MaxPermSize设置永久代的大小就决定了永久代的上限，但是不是总能知道应该设置为多大合适, 如果使用默认值很容易遇到OOM错误。

当使用元空间时，可以加载多少类的元数据就不再由MaxPermSize控制, 而由系统的实际可用空间来控制。

更深层的原因还是要合并HotSpot和JRockit的代码，JRockit从来没有所谓的永久代，也不需要开发运维人员设置永久代的大小，但是运行良好。同时也不用担心运行性能问题了,在覆盖到的测试中, 程序启动和运行速度降低不超过1%，但是这点性能损失换来了更大的安全保障。