

JVM入门与实战课堂笔记01

1.为什么可以用Object类

2.Java启动类是哪个

3.双亲委派与沙箱安全机制

4.native ->thread ->start-
>start0 ->JNI

5.PC寄存器->存储指令地址

6.方法区->所有被线程独享
就没有优化空间

7.栈-> 程序=算法+数据结
构:

程序=框架+业务逻辑-

栈：先进后出，队列：先进先出

8.垃圾回收算法：

引用计数法

复制算法

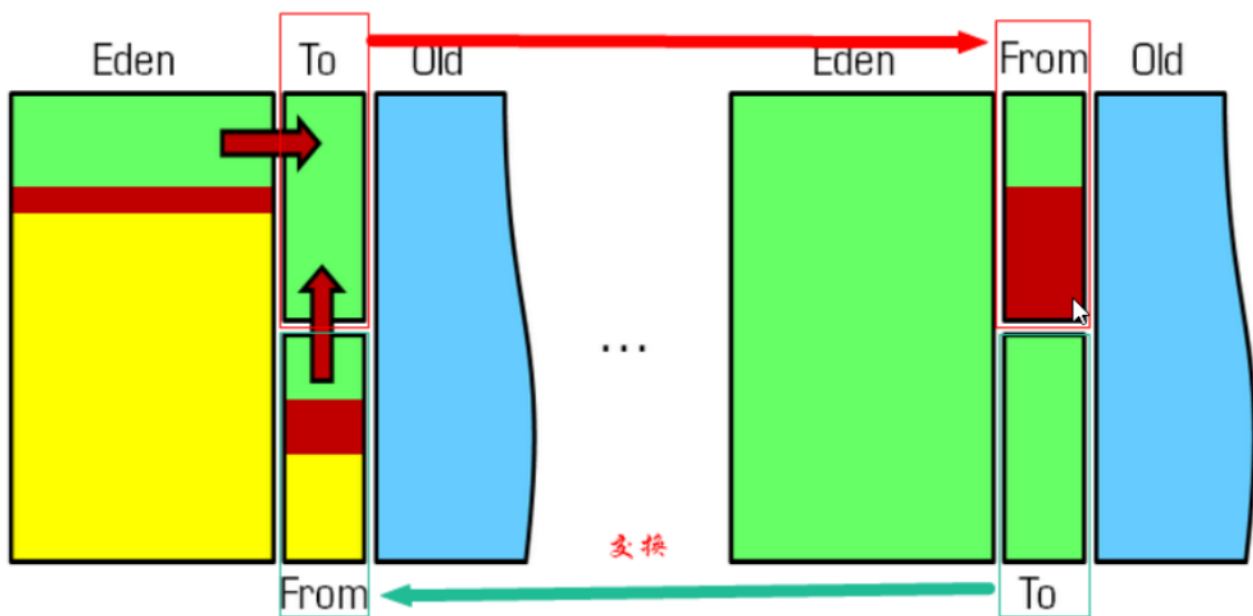
Minor GC会把Eden中的所有活的对象移动到Survivor区，如果Survivor区中放不下，剩下的活的对象被移动到Old区，收集后Eden为空

当对象在Eden(包括一个Survivor区域)出生后，再经过一次Minor GC后，如果对象还存活，并且能够被另外一块Survivor区域所容纳，则使用复制算法将这些存活的对象复制到另外一块Survivor区中，然后清理所使用的Eden以及Survivor区域，并且将这些对象的年龄设置为1，以后对象在Survivor区，每熬过一次Minor GC，对象年龄+1，当对象年龄到15岁，这些对象就成为老年代

原理：

- 从根集合(GC Root)开始，通过Tracing从From中找到存活对象，拷贝到To中；
- From、To交换身份，下次内存分配从To开始；





标记清除

1. 标记 (Mark) :

从根集合开始扫描，对存活的对象进行标记。



2. 清除 (Sweep) :

扫描整个内存空间，回收未被标记的对象，使用free-list记录空闲区域。



✔ 不需要额外空间

- ❌ 两次扫描，耗时严重；
- ❌ 会产生**内存碎片**

标记整理

标记-压缩 (Mark-Compact)

原理：

1. 标记 (Mark)：

与 **标记-清除** 一样。



2. 压缩 (Compact)：

再次扫描，并往一端 **滑动** 存活对象。



✓ 没有内存碎片，可以利用bump ✗ 需要移动对象的成本

标记清除压缩