



链滴

# JVM 对象创建与访问

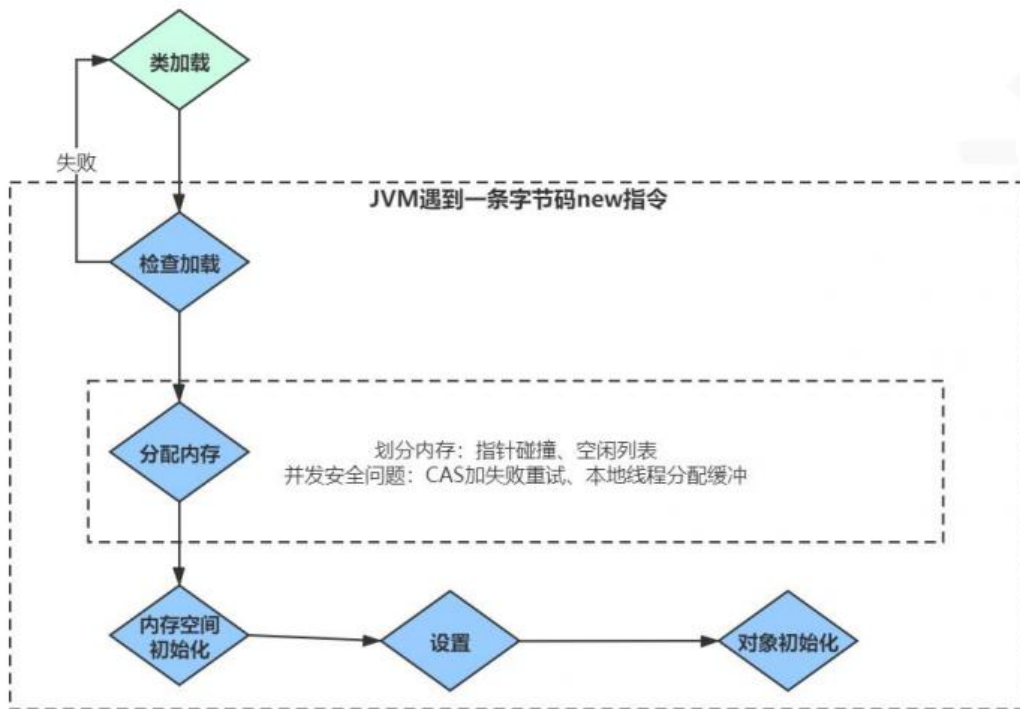
作者: [thas](#)

原文链接: <https://ld246.com/article/1607958891218>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 对象创建



## • 创建过程

- 检查加载 检查符号引用, 如未解析, 则类加载
- 分配内存
- 内存空间初始化 赋初值
- 对象头设置
- 初始化

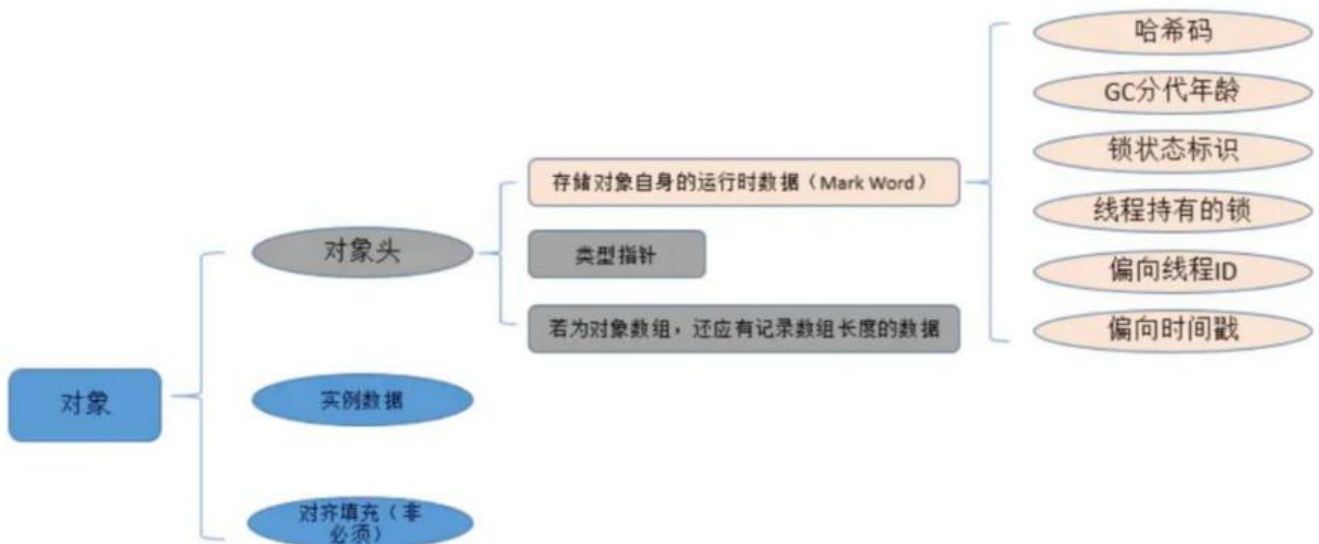
## • 划分内存方式

- 指针碰撞(内存整理)
- 空闲列表(标记清理)

## • 并发安全

- TLAB(线程本地缓冲)
- CAS抢占

# 对象内存布局



## 对象访问定位

- 句柄 间接引用堆上的实例(Hotspot使用) 安全, 方便GC
- 直接引用

## 对象分配策略

- 栈上分配
  - 经过逃逸分析, 无法逃逸的对象可能在栈上直接分配
  - 标量替换, 如果无法逃逸的对象只有标量属性赋值和访问操作, 则不创建对象, 直接拆分成栈上标量
- TLAB线程本地分配
- 正常堆上分配
  - 优先Eden区
  - 大对象直接分配在老年代
- 分配空间担保原则 新生代垃圾进入老年代前的担保
  - 默认: 当老年代可用空间大于新生代总对象大小, 则可分配; 如不满足, 进入担保策略, 老年代可用空间大于历史进入老年代的垃圾平均值, 则可分配, 如果分配失败, 则FullGC; 如不满足, 直接FullGC
  - 关闭HandlePromotionFailure担保: 当老年代可用空间大于新生代总对象大小, 则可分配; 如不足, 直接FullGC