

JVM笔记无特殊说明，基于jdk14

[零、参考资料 2](#_Toc38475864)

[一、对象 2](#_Toc38475865)

[二、GC基础 2](#_Toc38475866)

[三、GC实现 2](#_Toc38475867)

# 零、参考资料

Java虚拟机规范(Java SE 8版)

深入理解Java虚拟机——JVM高级特性与最佳实践 第三版

实战Java虚拟机——JVM故障诊断与性能优化

垃圾回收的算法与实现

垃圾回收算法手册

深入理解计算机系统

Stackoverflow:

https://stackoverflow.com/questions/16549066/java-major-and-minor-garbage-collections

OpenJdk：

https://openjdk.java.net/jeps/122

http://cr.openjdk.java.net/~sundar/8022483/webrev.01/raw\_files/new/src/share/classes/com/sun/tools/hat/resources/oqlhelp.html

Oracle官方DOC：

https://docs.oracle.com/javase/8/javase-books.htm

https://docs.oracle.com/en/java/javase/13/

https://docs.oracle.com/en/java/javase/14/

https://docs.oracle.com/javase/specs/jvms/se8/jvms8.pdf

https://docs.oracle.com/javase/specs/jvms/se14/jvms14.pdf

https://docs.oracle.com/javase/specs/index.html

https://www.oracle.com/webfolder/technetwork/tutorials/mooc/JVM\_Troubleshooting/week1/lesson1.pdf

https://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html

资料整理说明：

1. 参考《深入理解Java虚拟机——JVM高级特性与最佳实践》第三版，中文
2. 垃圾回收算法手册
3. 垃圾回收的算法与实现
4. 基于JDK14&Hotspot，但因为JDK8是目前主流版本，已被抛弃但JDK8中部分GC收集器内容还是会做简单整理
5. 不局限于JVM的GC内容，还包括垃圾回收算法基本实现理论

# 一、对象

## 1对象内存布局（HotSpot）

### 1.1、概述

1.1.1、布局划分为三部分：对象头(Header)、实例数据(Instance)、对齐填充(Padding)

1.1.2、对象头包含两类信息：用于存储对象自身的运行时数据、对象指向它的类型元数据的指针(类型指针)

### 1.2对象头(Header)

1.2.1对象自身的运行时数据(Mark Word)

a、这部分数据的长度在32位和64位的未开启压缩指针的JVM中分别为32bit和64bit。

b、Mark Word是一个有着动态定义的数据结构，目的在于在极小的空间存储更多的数据，根据对象的状态复用自己的存储空间

c、32位HotSpot中，如果对象未被同步锁锁定的状态下，Mark Word的32bit存储空间中，25bit用于存储对象哈希码，4bit存储对象分代年龄，2bit存储锁标志位，1bit固定为0。其他状态存储内容如下表

|  |  |  |
| --- | --- | --- |
| 存储内容 | 标志位 | 状态 |
| 对象哈希吗、对象分代年龄 | 01 | 未锁定 |
| 指向锁记录的指针 | 00 | 轻量级锁定 |
| 指向重量级锁的指针 | 10 | 膨胀(重量级锁定) |
| 空、不需要记录信息 | 11 | GC标记 |
| 偏向线程ID、偏向时间戳、对象分代年龄 | 01 | 可偏向 |

1.2.2类型指针

a、JVM通过这个指针来确定该对象是哪个类型的实例

b、并不是所有JVM实现都必须在对象数据上保留类型指针。

c、如果对象是数组，对象头中还必须有一块用于记录数组长度的数据

1.3、实例数据

## 2、New对象()

2.1、JVM创建对象过程

## 3、内存分配

## 4、对象查找

## 5、对象已死？

### 4.1、年龄记录

### 4.1、引用计数

### 4.2、可达性分析算法

### 4.2.1、GC ROOT

# 二、GC基础

## 1分区策略

### 1.1分代设计

### 1.2全区域设计

### 1.3大对象空间

## 2基本算法

### 2.1标记-清除

### 2.2标记-复制

### 2.3标记-整理

# 三、GC性能分析

# 四、GC实现

1 Serial收集器

2 ParNew

3 Parallel Scavennge

4 Serial Old

5 Parallel Old

6 CMS

7 G1

8 Shenanndoah

9 ZGC

10 Epsilon