链滴

理解 JVM 笔记系列（三）：对象 (附脑图)

作者：hudk
原文链接：https://ld246.com/article/1591964307288
来源网站：链滴

对象的创建过程

- 1、类加载检查

检查new指令的参数是否能在常量池中定位到一个类的符号引用，并且检查这个符号引用代表的类是已被加载、解析和初始化过。如果有，那必须先执行相应的类加载过程

- 2、分配内存

在类加载检查通过后，接下来虚拟机将为新生对象分配内存

方式： 指针碰撞、空闲列表

具体由那种方式来分配，跟实际运行时使用的垃圾收集算法有关，当垃圾收集使用标记整理算法时，闲内存就会比较规整，使用指针碰撞方式，当垃圾收集算法使用标记清除算法时，空闲内存可能比较碎，就需要使用空闲列表方式了，但是如果使用空闲列表方式拿到一块很大的空闲区域时，在这个大空闲区域里仍然会使用指针碰撞

并发问题解决方案： CAS、线程缓冲（Thread Local Allocation Buffer，TLAB）

- 3、设置默认值

除对象头内容以外的内容将会全部置0，这就是对象创建不用手动赋初始值就可以使用的原因。

- 4、设置对象头

类型信息、哈希码、分代年龄、锁等信息

- 5、执行构造方法

Java编译器会在遇到new关键字的地方同时生成invokespecial字节码指令，但如果直接通过其他方产生的则不一定如此，如果没有invokespecial字节码指令，则不会执行程序员自定义的构造方法

对象的内存布局

- 对象头

Mark Word： 哈希码、GC分代年龄、锁状态标志、线程持有的锁引用、偏向线程ID、偏向时间戳

类型指针： 指向它的类型元数据的指针，Java虚拟机通过这个指针来确定该对象是哪个类的实例

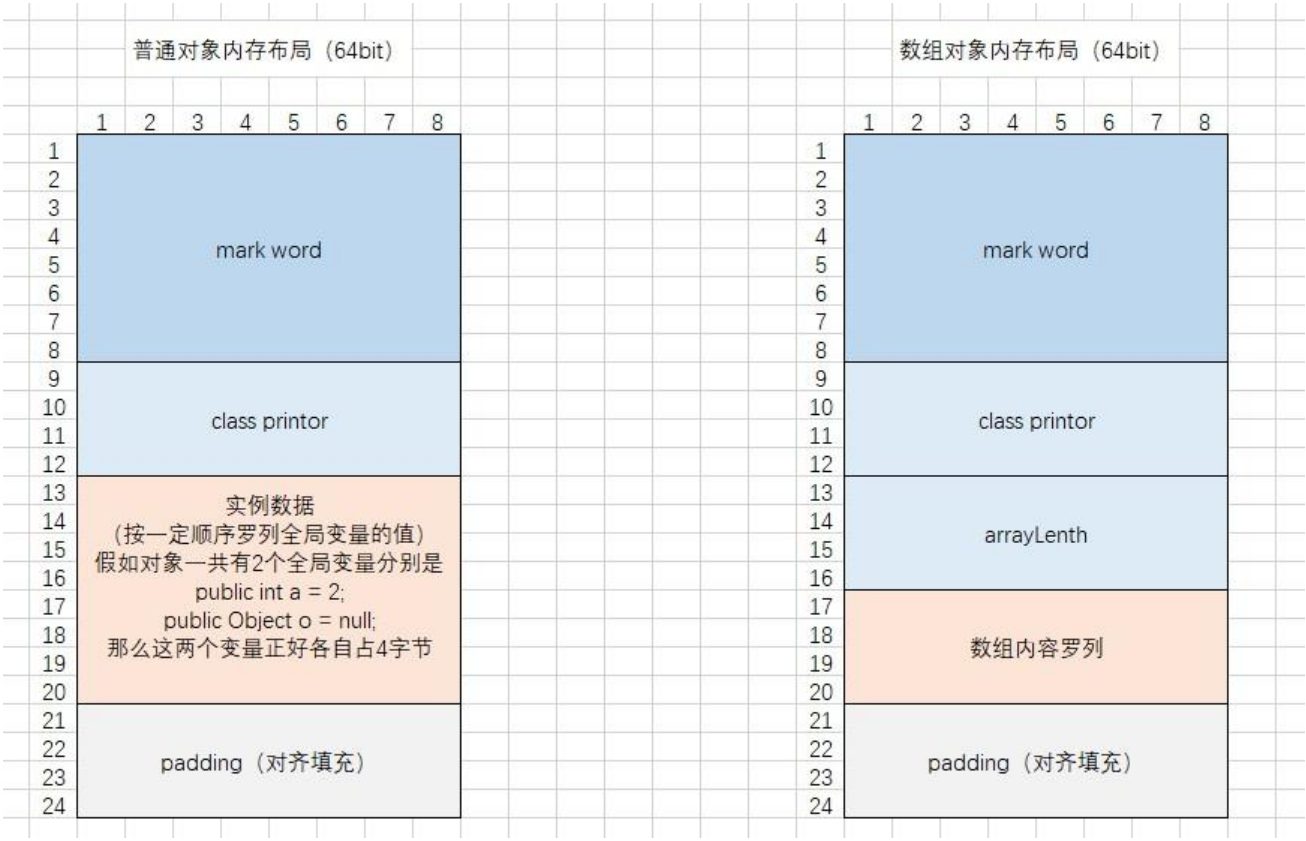
数组长度： （只有数组需要，一般对象没有这个部分）

- 对象体

对象的成员变量列表，按照一定顺序排列，其中longs/doubles类型占用8个字节（64bit）,int占用4字节（32bit），shorts/chars占用2个字节（16bit）,bytes/booleans占用1个字节（8bit）,引用类占用4个字节

- 对象填充

由于HotSpot虚拟机的自动内存管理系统要求对象起始地址必须是8字节的整数倍，换句话说就是任对象的大小都必须是8字节的整数倍。如果对象实例数据加上对象头部分没有对齐的话，就需要通过齐填充来补全。



对象的访问定位

● 句柄访问

优点： 所有对该对象的引用，指向的地址稳定不变，可以一直指向句柄的地址就可以，当对象位置发变化，只需要集中去句柄区修改一次对应的句柄值就可以了，

缺点： 每次访问对象，都会多一次间接访问的开销，积少成多对性能有一定影响

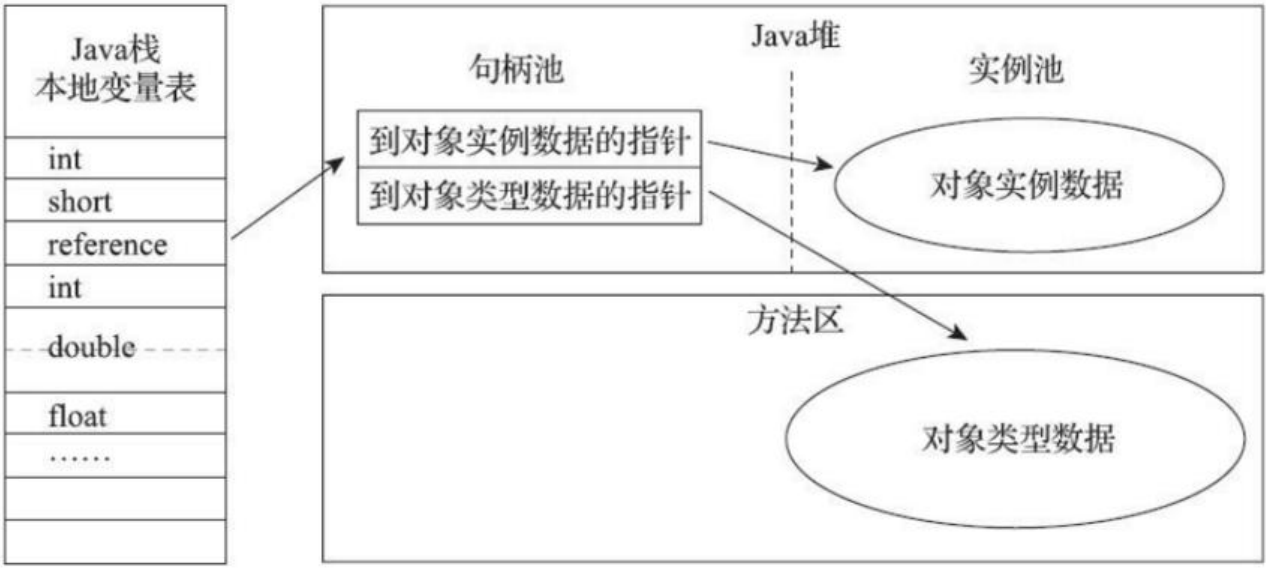


图2-2　通过句柄访问对象

● 直接指针访问

优点： 速度快

原文链接：理解 JVM 笔记系列（三）：对象 (附脑图)

缺点： 对象每次位置发生变化，都需要去遍历所有引用了这个对象的指针值并更改

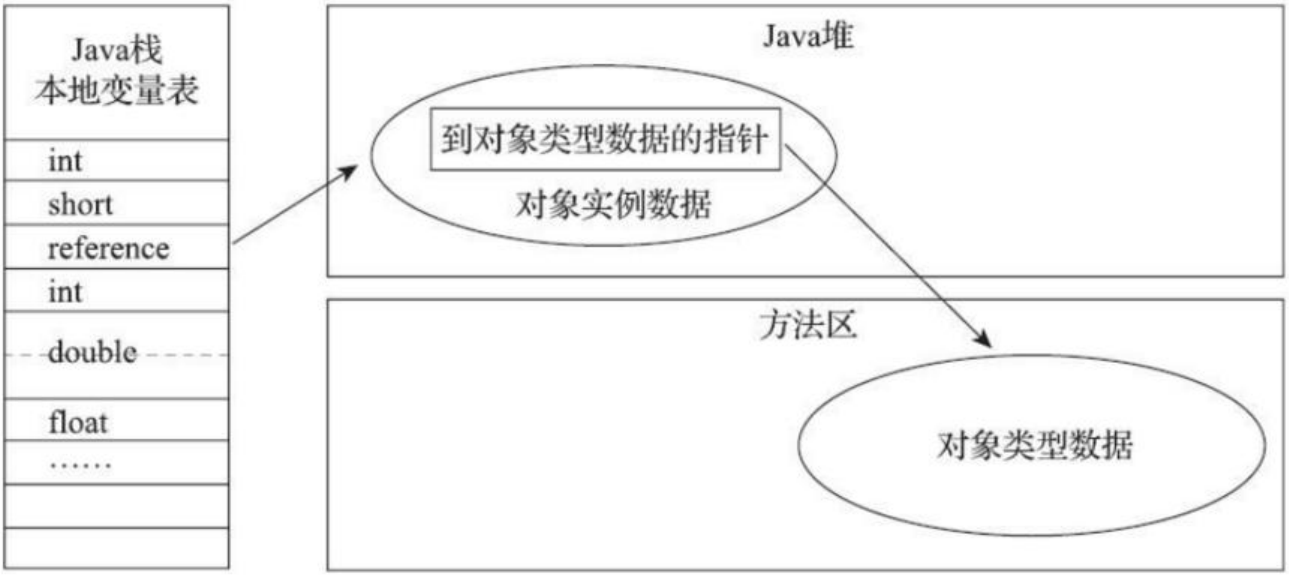
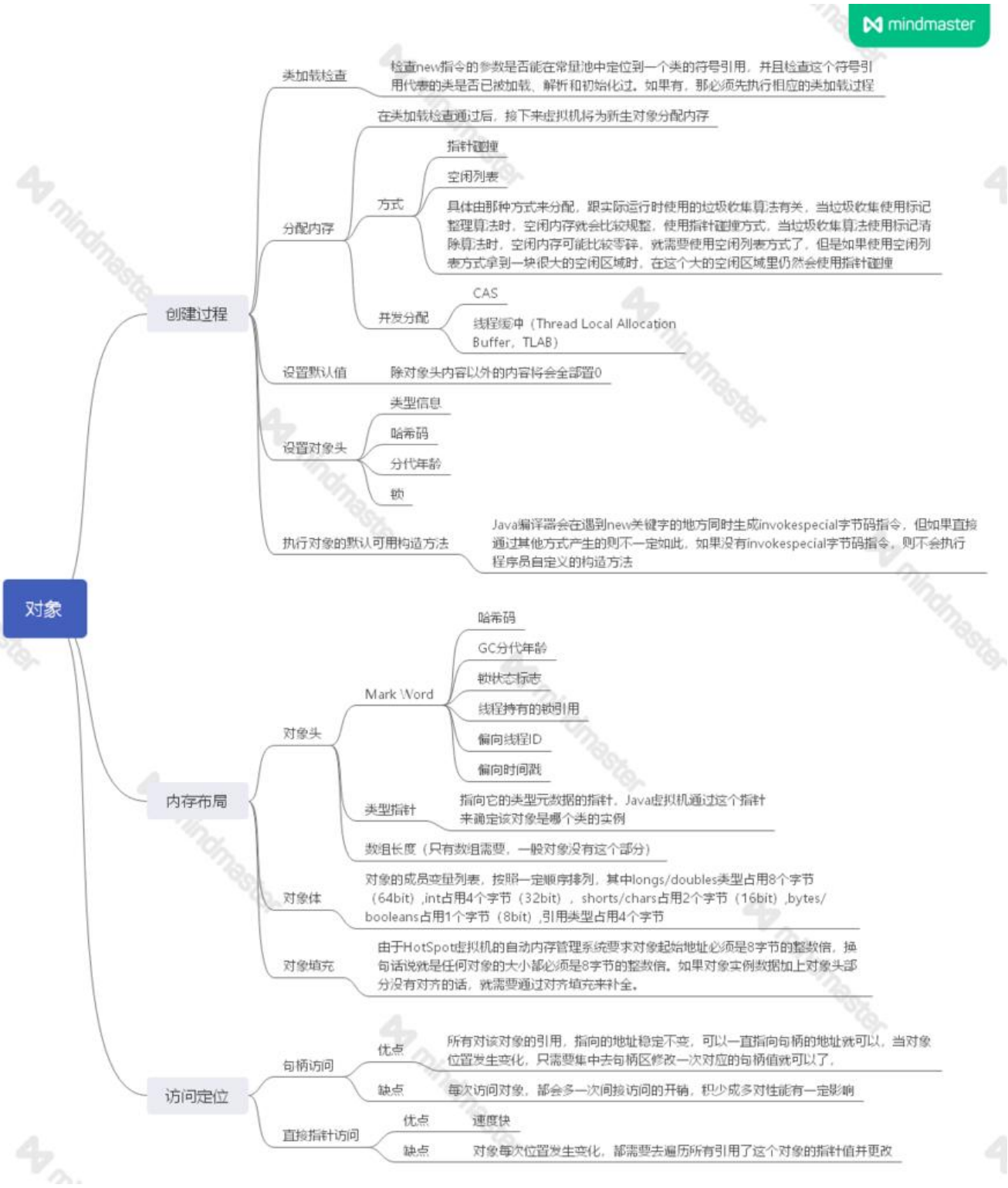


图2-3　通过直接指针访问对象

附脑图

原文链接：理解 JVM 笔记系列（三）：对象 (附脑图)



原文链接：理解 JVM 笔记系列（三）：对象 (附脑图)