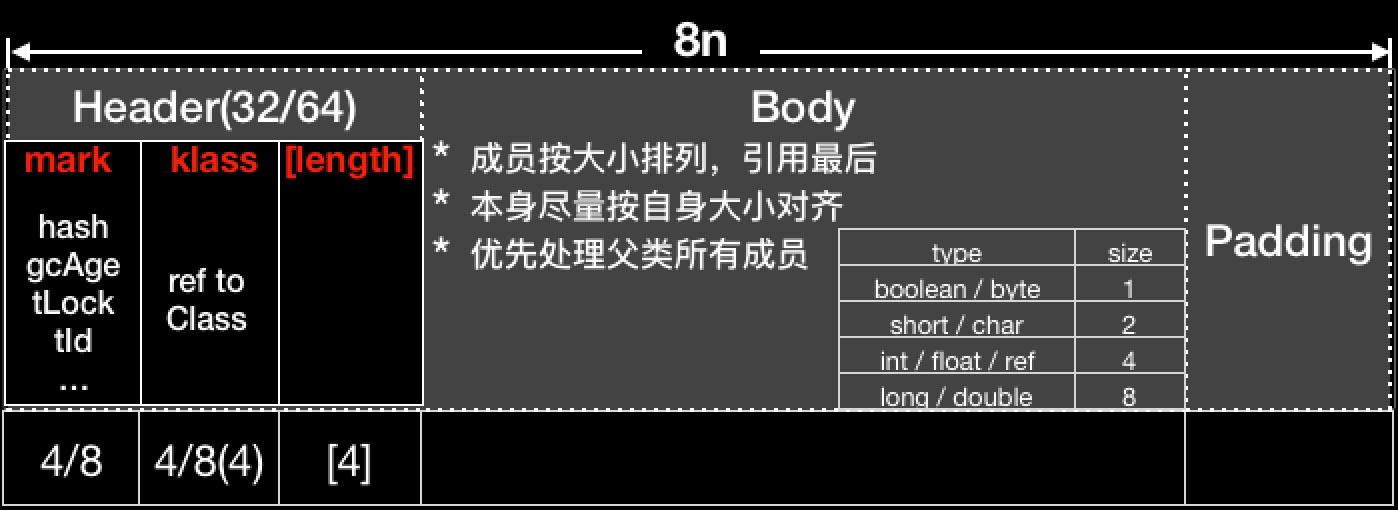
**对象的结构**

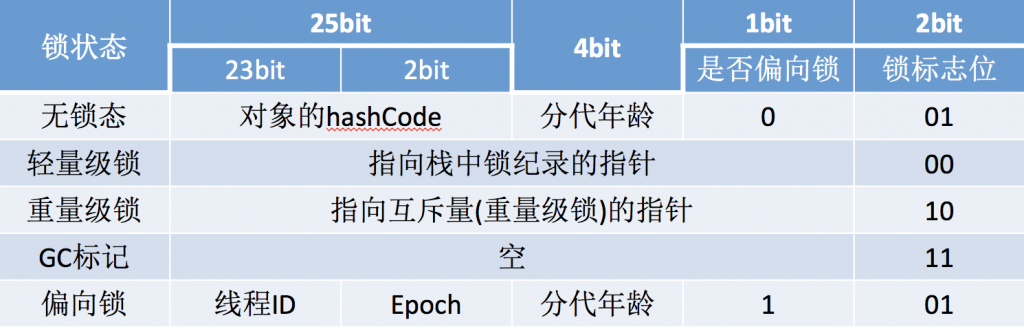
在HotSpot虚拟机中，Java对象的存储格式也是一个协议或者数据结构，底层是用C++代码定义的。Java对象结构大致如下图所示



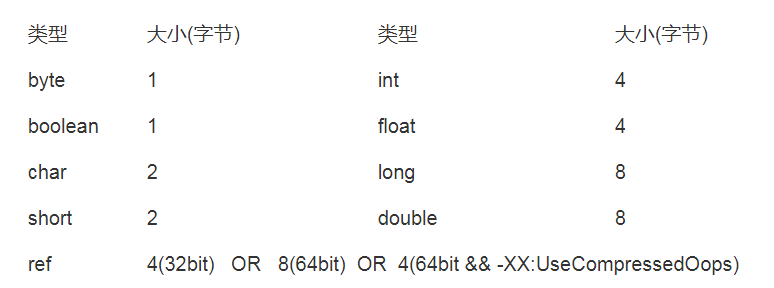
Java对象从整体上可以分为三个部分，对象头、实例数据和对齐填充。

****对象头：Instance Header****，Java对象最复杂的一部分，采用C++定义了头的协议格式，存储了Java对象hash、GC年龄、锁标记、class指针、数组长度等信息。

因为对象头信息只是对象运行时自身的一部分数据，相比实例数据部分，头部分属于与业务无关的额外存储成功。为了提高对象对堆空间的复用效率，Mark Word被设计成一个非固定的数据结构以便在极小的空间内存储尽量多的信息，它会根据对象的状态复用自己的存储空间



****实例数据：Instance Data****，这部分数据才是真正具有业务意义的数据，实际上就是当前对象中的实例字段。在VM中，对象的字段是由基本数据类型和引用类型组成的。其所占用空间的大小如下所示：



说明：其中ref表示引用类型，引用类型实际上是一个地址指针，32bit机器上，占用4字节，64bit机器上，在jdk1.6之后，如果开启了指针压缩(默认开启: -XX:UseCompressedOops，仅支持64位机器)，则占用4字节。Java对象的所有字段类型都可映射为上述类型之一，因此实例数据部分的大小，实际上就是这些字段类型的大小之和。当然，实际情况可能比这个稍微复杂一点，如字段排序、内部padding以及父类字段大小的计算等。

****对齐填充：Padding****，VM要求对象大小须是8的整体数，该部分是为了让整体对象在内存中的地址空间大小达到8的整数倍而额外占用的字节数。