



链滴

【原理基础】《JAVA 编程思想》学习笔记 - JAVA 存储位置说明

作者: [moonce](#)

原文链接: <https://ld246.com/article/1539617393582>

来源网站: [链滴](#)

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

<h4 id="1-寄存器">1.寄存器</h4>

<p>这是最快的存储区，因为它位于不同于其他存储区的地方——处理器内部。但是寄存器的数量极有限，所以寄存器根据需求进行分配。你不能直接控制，也不能在程序中感觉到寄存器存在的任何迹。</p>

<h4 id="2-堆栈">2.堆栈</h4>

<p>位于通用的 RAM（随机访问存储器）中，但通过堆栈指针可以从处理器那获得直接支持。堆栈指针若向下移动，则分配新的内存；若向上移动，则释放那些内存。这是一种快有效的分配存储方法，仅次于寄存器。创建程序时，Java 系统必须知道存储在堆栈内所有的确切生命期，以便于上下移动指针。这一约束限制了程序的灵活性，所以虽然某些 Java 数据存储于堆栈中——特别是对象引用，但是 Java 对象并不存储在堆栈中。</p>

<h4 id="3-堆">3.堆</h4>

<p>一种通用的内存池（位于 RAM 区），用于存放所有的 Java 对象。堆不同于堆栈的好处是：编译器不需要知道存储的数据在堆里存活多长时间。因此，在堆里分配存储有很大的灵活性。当需要一个对象时，只需要 new 写一行简单的代码，当执行这行代码时，会自动在堆里进行存储分配。当然，为种灵活性必须付出相应的代价：用堆进行存储分配和清理可能比用堆栈进行存储分配需要更多的时间</p>

<h4 id="4-常量存储">4.常量存储</h4>

<p>常量值通常直接存放在程序代码内部，这样做是安全的，因为它们永远不会被改变。有时，在嵌入式系统中，常量本身会和其他部分隔离开，所以在这种情况下，可以选择将其存放在 ROM（只读存储器）中。</p>

<h4 id="5-非RAM存储">5.非 RAM 存储</h4>

<p>如果数据完全存活于程序之外，那么它可以不受程序的任何控制，在程序没有运行时也可以存在其中两个基本的例子是流对象和持久化对象。在流对象中，对象转化成字节流通常被发送给另一台机器。在“持久化对象”中，对象被存放于磁盘上，因此，即使程序终止，他们可以保持自己的状态。这种存储方式的技巧在于：把对象转化成可以存放在其他媒介上的事物。在需时，可恢复成常规的、基于 RAM 的对象。Java 提供了对轻量级持久化的支持而侏儒 JDBC 和 Hibernate 这样的机制提供了更加复杂的对在数据库中存储和读取对象信息的支持。</p>