



链滴

# java 数据的存储

作者: [litongfei](#)

原文链接: <https://ld246.com/article/1503378895260>

来源网站: [链滴](#)

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

## \*\* 1.寄存器。 \*\*

这是最快的存储区，因为它位于不同于其他存储区的地方--处理器的内部。但是寄存器的数量极其有限，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象。

## \*\* 2.堆栈。 \*\*

位于通用RAM（随机访问存储器）中，但通过堆栈指针可以从处理器那里获得直接支持。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。这是一种快速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项的确切生命周期，以便上下移动堆栈指针。这一约束限制了程序的灵活性，所以虽然某些java数据存储在堆栈中--特别是对象引用，但是java对象并不存储于其中。

## \*\* 3.堆。 \*\*

一种通用的内存池（也位于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好的灵活性。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，为这种灵活性必须要付出相应的代价：用堆里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候。

## \*\* 4.常量存储。 \*\*

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，可以选择将基存放在ROM中。

## \*\* 5.非RAM存储。 \*\*

如果数据完全存活于程序之外，那么这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，通常被发送能另一台机器。在“持久化对象”中，对象存入于磁盘上，因此，即使程序终止，它们仍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有其他媒介的事物，在需要时恢复成常规的、基于RAM的对象。java提供了对轻量级持久化的支持，而诸如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。