

java认证辅导:深入Java对象及元素的存储区域Java认证考试

PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/594/2021_2022_java_E8_AE_A4_E8_AF_81_c104_594712.htm

在JAVA平台上开发应用程序的时候，有一个很大的特点就是其是在应用程序运行的时候才建立对象。换句话说，在程序运行的时候，才会最终确定对象的归属，即对象应该存储在什么地方。由于存储在不同的区域，其在性能上会有所不同。为此作为Java程序开发人员需要了解各个存储区域的特点以及对性能的影响。然后再根据需要进行调整应用程序的区域分配。总的来说，在操作系统中有五个地方可以用来保存应用程序运行中的数据。这类区域的特点以及对性能的影响分析如下。

存储区域一：寄存器 虽然同在内存中，但是不同的区域由于用途不同，其性能也有所不同。如就拿Java应用程序来说，寄存器由于其处于处理器的内部，为此这个区域存取数据最快。跟内存中的其他存储区域有着天壤之别。那么我们把所有对象都放到这个区域内，不就可以提高Java应用程序的性能了吗?理论上是如此，但是在现实中是行不通的。因为这个寄存器的数量是非常有限的。在内存中的寄存器区域是由编译器根据需要来分配的。我们程序开发人员不能够通过代码来控制这个寄存器的分配。所以说，这第一个存储区域寄存器，我们只能够看看，而不能够对其产生任何的影响。

存储区域二：堆栈 对象的创建有两种方式，一是在应用程序开发的过程中就创建对象.二是在程序运行的过程中要用到对象的时候再来创建对象。前者比后者性能要高，而后者比前者要灵活。这主要是因为前者创建对象的时候，就是这个堆栈中创建的。虽然其创建的

对象没有保存在寄存器中，但是通过这个对象的推栈指针可以直接从处理器哪里获得相关的支持。如堆栈指针往上移动的时候，则释放原有对象占用的内存。如堆栈指针向下移动时，则为对象分配新的内存。所以，如果把对象存放在这个堆栈中，虽然性能没有像存放在寄存器中那么理想，但是仍然比存储在其他地方要好的多。由于Java程序是在程序运行过程中才根据需要来创建对象。为此对象就不能够保存在这个堆栈中。不过Java应用程序也不能够白白的浪费这个宝贵的空间。为此虽然Java对象本身没有保存在这个堆栈中(不是不保存而是这里没有他的容身之地)，但是还是应该把一些可以放的内容放到这个堆栈中，以提高应用程序的性能。如可以把一些对象引用存放在这个堆栈中。另外对于一些基本的数据类型对象，Java程序也往往把他们放置在堆栈中，以提高数据处理的性能。如一些整数型、字符型的数据对象，这些对象有些共同的特点，如对象比较小、是Java程序提供的标准对象等等。对于这些对象由于每个应用程序基本上都需要用到，而且我们程序开发人员只能够引用这些对象，而不能够对其进行更改。为此Java程序在处理的时候，往往一开始就创建了对对象(即直接在堆栈中创建对象并保存)，而不像其他对象一样，在需要的时候才创建。只所以在堆栈中创建这些对象，还有一个重要的原因。因为如果在堆栈中创建对象的话，Java编辑器必须知道存储在堆栈内所有数据的确切大小和生命周期。为了得到这些信息，必须产生相关的代码来获得这些信息，以便其操作堆栈指针。普通的对象大小、生命周期等等难以预先获得，为此在堆栈中创建普通的对象，对于Java应用程序来说并不是很合适。相反，这些Java编译器预定义的对

象大小并不会随着机器硬件架构的变化和用户需求的变化而变化.而且这些对象往往从始之终都会存在的，所以也不存在生命周期的问题。所以把这些对象放置在堆栈中是合理的，也是可实现的。如此处理，不仅不会影响到对象的灵活性，而且还可以提供比较好的性能。

存储区域三：堆堆虽然跟堆栈一样，都是随机访问存储器中的区域，但是两者有很大的不同。因为在堆中，没有堆栈指针，为此也就无法直接从处理器那边获得支持。为此其性能跟堆栈比起来，就有一定的差距。通常情况下，除上面所说的一些预定义对象之外，其他的对象都是保存在这个堆中的。或者说，利用new关键字创建的对象都是保存在堆中的。保存在堆中其好处也是显而易见的。如Java编译器不需要知道从堆里需要分配多少存储区域，也不必知道存储的数据在堆里会存活多长时间。所以在堆里分配存储有很大的灵活性。当需要对象时，我们可以使用New关键字建立一个对象。然后系统会自动给这个对象在堆中分配一个区域让其作为归宿。不过其最大的不足之处，就是在堆中创建对象与分配存储区域，要比在堆栈中慢许多。鱼与熊掌不能兼得呀。

存储区域四：静态存储区域与常量存储区域在Java对象中有一些特殊的元素。如有些元素是比较特别的(如利用关键字Static定义的变量)。这些变量对于其他对象来说，可能就是静态的。为了更好的管理这些变量，Java在内存中专门划分了一个静态存储区域来管理这些元素。这里的静态存储区域就是指在固定的位置存放应用程序运行时一直存在的数据。这里需要明确的一点就是，Java对象是不保存在这个地方的，而只是把对象中的一些特殊元素放置这里。由于位置固定，所以下次调用的时候就省去了查找的

麻烦。为此其对于提供应用程序的性能是有利的。作为我们程序开发人员来说，在书写代码的时候，就需要灵活应用Static这个关键字。笔者的意见是，能用则用.不能用的时候也要想着法儿用。特别是有些元素用不用Static关键字时对于程序功能没有影响，此时我们要理直气壮的在元素前面加上Static关键字。在Java对象中还有一类特殊的元素，我们叫做常量。由于常量的值是稳定不变的，如圆周率。为此把他们放在代码的内部是可行的。不过有些时候，在进行一些嵌入式系统开发的时候，我们往往不这么做。而是会把常量元素跟代码分开来保存。如我们会根据情况把常量的值存放在一些只读存储器中。这主要是为了一些特殊的功能考虑的。如出于版权控制的需要。如在打印机上为了保护原装耗材的版权，往往把常量跟代码分开存放。

存储区域五：非RAM存储

有时候，有些程序运行所需要的数据我们还会放置在其他地方。如在一些系统中需要用到流对象，这个对象的数据并没有保存在上面所谈到的任何一个存储区域，这个对象直接被转为为字节流，发送到其他的主机上去了。另外有一种叫做持久化的对象，其是被存储在硬盘中的。这些对象平时在应用程序开发过程中用到的并不是很多，大家只需要了解有这些对象的存在即可。等到需要用到时候，再去深入研究也不迟。从上面的分析中我们可以看到，对象的归属我们程序开发人员很难控制。寄存器是编译器来管理的。而堆与堆栈又基本上受到开发平台的限制，我们程序人员也没有这个能耐来干涉他们。其实我们主要能够调整与控制的就是第四个存储区域，即静态存储与常量存储。笔者的建议是，对于非嵌入式程序，能够利用静态存储来实现的，就尽量采用静

态存储。而对于常量来说，需要根据需要实现的功能来判断是否需要把常量存储在只读存储器中。有时候对于版权的保护等等需要用到这个只读存储器。更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详细资料 100Test 下载频道开通，各类考试题目直接下载。详细请访问

www.100test.com