

java认证:Java开发平台中的生命周期管理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\\_ti2020/584/2021\\_2022\\_java\\_E8\\_AE\\_A4\\_E8\\_AF\\_81\\_c104\\_584689.htm](https://www.100test.com/kao_ti2020/584/2021_2022_java_E8_AE_A4_E8_AF_81_c104_584689.htm) 对象的生命周期没有大家想的那么复杂。其实其就是回答了两个简单的问题。什么时候需要创建对象？什么时候把对象消亡？这两个问题解决了，那么对象的生命周期问题也就解决了。一、传统的对象生命周期管理方法。为了对Java开发平台中的生命周期管理方法有一个更加深刻的理解，笔者认为大家需要对传统的生命周期管理方法先有一个了解。因为这有助于大家有一个相互的对比，从而会对Java语言中的生命周期管理方法会有一个比较直观的印象。在传统的编程语言中，如C语言，其考虑对象的生命周期，主要是从对象与应用程序的执行效率出发的。为了追求应用程序最大的执行效率，C语言要求在编写应用程序的时候程序开发人员必须确定对象的存储空间与生命周期，既可以通过堆栈或者限域变量来实现。这么设计的话，好处是很明显的。由于在程序开发的时候就涉及好了对象的存储空间与生命周期，那么在程序运行的时候就不需要再花精力去创建对象。应用程序开发人员需要知道的是，在堆栈中创建与释放对象的存储空间是一件很容易且不用花费多少时间的事情。如在堆栈中创建对象空间只需要一条将栈顶指针向下移动的汇编指令，另一条汇编指令对应释放存储空间所需的将栈顶指针往上移动即可。所以应用程序的执行速度就会比较高，可以将对象存储空间的分配和释放置于比较有限考虑的地位。但是这种管理方法的最大缺陷就是灵活性不高。因为在应用程序开发的时候就必须要知道对象的确切数量

、生命周期和存储空间。如果后续需要对应用程序进行二次开发或者功能升级，如果涉及到添加新对象的话，那么修改的工作量就会很大。而且对于普通的应用程序开发人员来说，很难对所需要的对象的数量与生命周期，做出一个准确的预计。可见，这种管理方式虽然可以保证应用程序的最大运行速度，但是其局限性也是蛮大的。其不仅灵活性不高，而且对于应用程序开发者有更高的要求。

## 二、Java平台下的对象生命周期管理方法。

在Java语言平台下，其与传统的对象管理方法有很大的不同。在Java语言平台下，叫做动态创建对象。简单地说，就是不是在应用程序开发的时候创建对象，而是在应用程序运行的过程中动态的根据需要建立对象。与上面的对象生命周期不同，动态的创建对象，只在应用程序运行的时候才会考虑需要创建多少对象，他们的生命周期有多久，对象的类型是什么等等。这些问题只有在应用程序代码被执行的那一刻才能够最终的确定下来。在应用程序开发的过程中不需要关注这些问题。如在Java语言开发中，一开始不需要建立对象。在需要对象的时候，可以利用new关键字来创建一个对象。很明显，由于对象的创建与消亡都是在应用程序运行的时候动态管理的，为此其灵活性要比传统的静态管理方式要高。当需要新的对象相时，只需要在代码中利用NEW关键字创建一个即可，而不需要对其他的内容进行更改。为此，象Java这种采用动态机制来管理对象生命周期的语言，往往被认为其在升级、二次开发上面都比传统的C语言程序要高。而且由于在需要的时候创建对象，为此对于应用程序开发者的要求也相对较低一点。应用程序开发人员进入Java语言的门槛也会低许多。但是其最大的缺陷就是会影

响到应用程序的执行效率。因为如果采用动态创建对象的话，则对象的存储空间是在应用程序运行时被动态管理的。所以此时应用程序就需要耗费额外的时间在堆中分配存储空间，这个时间就要远远的大于第一种静态管理对象在堆栈中创建存储空间的时间。为此在同等条件下，Java语言开发的应用程序效率就比C语言开发的应用程序要低一点。如果不采取其他措施来优化Java语言性能的话，那么这个差距会很大。为此Java语言采用这种对象生命周期管理方式，主要就是看中其对象管理过程中的灵活性。动态创建对象的管理方式可以方便解决一般化编程问题。这也会创建一些应用程序自定义平台提供了帮助。

### 三、通过垃圾回收器来自动回收不需要用的对象。

动态管理对象的方式其除了在性能上有所欠缺外，最重要的是其在对象的回收上也有比较大的缺陷。如通过堆栈来创建对象（第一种管理对象的方式），语言编译器可以确定对象存活的时间，并在时间到期后自动收回对象所占用的空间。而如果在堆上创建对象（第二种对象管理方式），则语言编译器就对应用程序对象的生命周期一无所知，不知道什么时候对象会消亡。为此如果不采取其他处理机制的话，则需要通过代码来判断对象的生命周期，并在适当的时候收回对象所占用的存储空间。可是这种通过代码来判断对象生命周期的方式，会在很大程度上造成应用程序的安全漏洞，若发现内存泄露等等的安全隐患。不过如果采用Java语言作为开发平台的话，则不用担心这个问题。因为在Java语言平台下，提供了一种叫做垃圾回收器的管理机制。垃圾回收器就好像是一个监视器，其时刻监视着应用程序的运行。它可以自动发现应用程序对象何时消亡，从而帮助应用程序释放

对象所占用的空间。也就是说，如果采用Java语言的垃圾回收器，不需要再通过编码来判断什么时候消亡对象。这项工作的话，Java语言编译器会帮助程序开发人员来完成。说的再直白一点，应用程序开发人员只需要负责创建对象即可。对象需要消亡的时候，其会自动消失（当然是在垃圾回收器的帮助下）。可见通过Java垃圾回收器的帮助，可以解决通过堆来创建对象的不足之处。应用程序开发人员可以不用考虑该在什么时候释放对象所占用的空间，不需要开发额外的代码来管理这个问题。从而可以在最大程度上避免应用程序内存泄露的安全隐患。另外需要注意的一点就是，虽然Java语言的垃圾回收器已经解决了对象回收与空间释放的问题。但是其对于性能方面的改善不能够起到任何的作用。所以这个性能上的缺陷仍然成为Java语言的一个瓶颈资源。当然这个问题也不是我们能够解决的，需要那些语言专家来想法设法提高动态分配对象空间的效率。我们应用程序开发者所能够做的就是通过优化Java语言代码，通过代码复用与继承等方式来提高应用程序的性能，让应用程序的性能有所改善。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](http://www.100test.com)