



链滴

1.10对象的创建和生命周期

作者: [someone756](#)

原文链接: <https://ld246.com/article/1441537769968>

来源网站: [链滴](#)

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

在使用对象时，最关键的问题之一便是它们的生成和销毁的方式。每个对象为了生存都需要资源，尤其是内存。当我们不再需要一个对象时，必须被清理掉，使其占有的资源可以被释放和重用。在相对简单的编程情况下，怎么清理对象似乎不什么挑战：你创建了对象，根据需要使用它，然后它应该被销毁。然而，你很可能遇到相对复杂的情况。

例如，假设你正在为某个机场设计空中通管理系统（同样的模型在仓库货柜管理系统、录像带出租系统或宠物寄宿店也适用）。一开始问题乎很简单：创建一个容器来保存所有的飞机，然后为美宜佳飞进空中交通控制区域的飞机创建一个新飞机对象，并将其置于容器中。对于清理工作，只需在飞机飞离此区域是删除相关的飞机对象即可。

但是，可能还有别的系统记录着有关飞的数据，也许这些数据不需要向主控制功能那样立即引人注目。例如，他可能记录着所有飞离机场的型飞机的飞行计划。因此你需要有第二个容器来存放小型飞机；无论何时，只要创建的是小型飞机对，那么它同时应该被置入第二个容器中。然后某个后台进程在空闲时对第二个容器内的对象进行操作

现在问题变得困难了：怎样才能知道何销毁这些对象呢？当处理完某个对象之后，系统某个其它部分可能还在处理它。在其他很多场合中也遇到同样的问题，在必须明确删除对象的编程系统中（例如：C++），此问题会变得十分麻烦。

对象的数据位于何处？怎样控制对象的明周期？C++认为效率控制是最重要的议题，所以给程序员提供了选择的权利。为了追求最大的执行度，对象的存储空间和生命周期可以在编写程序时确定，这可以将对象置于堆栈（它们优势被称为自变量（automatic variable）或限域变量（scoped variable））或静态存储区域内来实现。这种方式存储空间分配和释放置于优先考虑的位置，某些情况下这样控制非常有价值。但是也牺牲了灵活性。为必须在编写程序时知道对象确切的数量、生命周期和类型。如果试图解决更一般化的问题，例如计机辅助设计、仓库管理或者空中交通控制，这种方式就显得过于受限了。

第二种方式是在被称为对（heap）的内池中动态的创建对象。在这种方式中，知道运行时才知道需要多少对象，它们的生命周期如何，以及们的具体类型是什么。这些问题的答案只能在程序运行时相关代码被执行到的那一刻才能确定。如果要一个对象，可以在需要的时候直接在堆中创建。因为存储空间运行的时候是被动态管理的，所以需要大量的时间在堆中分配存储空间，这可能要远远大于在堆栈中创建存储空间的时间。在堆栈中创建存空间和释放存储空间通常各需要一条汇编指令即可，分别对应将栈顶指针向下移动和将栈顶指针向上动。创建堆存储空间的时间依赖于存储机制的设计。

动态方式有这样一个一般性的逻辑假设对象趋向于变得复杂，对于查找和释放存储空间的开销不会对对象的创建造成重大打击。动态方式所来的更大的灵活性正是解决一般化编程问题的要点所在。

Java完全采用了动态内存分配方式。每想要创建新对象时，就要使用new关键字来构建此对象的动态实例。

还有一个议题，就是对象的生命周期。于允许在堆上创建对象的编程语言，编译器可以确定对象存活的时间，并可以自动销毁它。然而，如在栈上创建对象，编译器就会对它的生命周期一无所知。在像C++这样的语言中，必须通过编程方式确定何时销毁对象，这可能会因为不能正确处理而导致内存泄露（C++程序中的家常便饭）。Java提了被称为“垃圾回收器”的机制，它可以自动发现对象何时不再使用，并继而销毁它。圾回收器非常有用，因为它减少了所必须考虑的议题和必须编写得代码。更重要的是，垃圾回收器提了更高层的保障，可以避免暗藏的内存泄露问题，这个问题已经使许多C++项目折戟沉沙。

Java的垃圾回收器被设计用来处理内存放问题（尽管它不包括清理对象的其他方面）。垃圾回收器“知道”对象何时不再被使，并自动释放对象占用的内存。这一点同所有对象都是继承自单根基类Object以及只能以一种方式创对象（在堆上创建）这两个特性结合起来，使得Java编程的过程跟C++相比要简单得多，所以要做出决策和要克服的障碍也要少很多。

总结

- 由空中交通管理系统引出对象应该何时被销毁。

li>
 C++将对象放在堆栈中，存储空间跟生命周期在编写程序确定。Java将对象放在堆中，需要时直接new，不用的时候垃圾回收器自动回收。
 因为Java是动态管理存储空间，所以需要大量时间分配存储空间，其所用时间可能要远远大于C++在堆栈中创建存储空间的时间。
 动态方式逻辑假设：对象越复杂，查找和释放对象用的时间所占比例越小。
 自动释放对象占用内存同所有对象都是单根继承自基类Object以及只能以一种方式创建对象（在堆上创建）这两个特性结合起来使Java编程比C++编程简单。

<p> ----END----