

# STL 简介，标准模板库

<http://www.cnblogs.com/lancidie/archive/2011/01/19/1939458.html>

这篇文章是关于C++语言的一个新的扩展——标准模板库的（Standard Template Library），也叫STL。

当我第一次打算写一篇关于STL的文章的时候，我不得不承认我当时低估了这个话题的深度和广度。有很多内容要涵盖，也有很多详细描述STL的书。因此我重新考虑了一下我原来的想法。我为什么要写这篇文章，又为什么要投稿呢？这会有什麼用呢？有再来一篇关于STL的文章的必要吗？

当我翻开Musser and Saini的页时，我看到了编程时代在我面前消融。我能看到深夜消失了，目标软件工程出现了。我看到了可维护的代码。一年过去了，我使用STL写的软件仍然很容易维护。让人吃惊的是其他人可以没有我而维护的很好！

然而，我也记得在一开始的时候很难弄懂那些技术术语。一次，我买了Musser&Saini，每件事都依次出现，但是在那以前我最渴望得到的东西是一些好的例子。

当我开始的时候，作为C++一部分的Stroustrup还没出来，它覆盖了STL。

因此我想写一篇关于一个STL程序员的真实生活的文章可能会有用。如果我手上有一些好的例子的话，特别是象这样的新题目，我会学的更快。

另外一件事是STL应该很好用。因此，理论上说，我们应该可以马上开始使用STL。

什麼是STL呢？STL就是Standard Template Library，标准模板库。这可能是一个历史上最令人兴奋的的工具的最无聊的术语。从根本上说，STL是一些“容器”的集合，这些“容器”有list,vector,set,map等，STL也是算法和其他一些组件的集合。这里的“容器”和算法的集合指的是世界上很多聪明人很多年的杰作。

STL的目的是标准化组件，这样你就不用重新开发它们了。你可以仅仅使用这些现成的组件。STL现在是C++的一部分，因此不用额外安装什麼。它被内建在你的编译器之内。因为STL的list是一个简单的容器，所以我打算从它开始介绍STL如何使用。如果你懂得了这个概念，其他的就都没有问题了。另外，list容器是相当简单的，我们会看到这一点。

这篇文章中我们将会看到如何定义和初始化一个list，计算它的元素的数量，从一个list里查找元素，删除元素，和一些其他的操作。要作到这些，我们将会讨论两个不同的算法，STL通用算法都是可以操作不止一个容器的，而list的成员函数是list容器专有的操作。

这是三类主要的STL组件的简明纲要。STL容器可以保存对象，内建对象和类对象。它们会安全的保存对象，并定义我们能够操作的这个对象的接口。放在蛋架上的鸡蛋不会滚到桌上。它们很安全。因此，在STL容器中的对象也很安全。我知道这个比喻听起来很老土，但是它很正确。

STL算法是标准算法，我们可以把它们应用在那些容器中的对象上。这些算法都有很著名的执行特性。它们可以给对象排序，删除它们，给它们计数，比较，找出特殊的对象，把它们合并到另一个容器中，以及执行其他有用的操作。

STL iterator就象是容器中指向对象的指针。STL的算法使用iterator在容器上进行操作。Iterator设置算法的边界，容器的长度，和其他一些事情。举个例子，有些iterator仅让算法读元素，有一些让算法写元素，有一些则两者都行。Iterator也决定在容器中处理的方向。

你可以通过调用容器的成员函数begin()来得到一个指向一个容器起始位置的iterator。你可以调用一个容器的end()函数来得到过去的最后一个值（就是处理停在那的那个值）。

这就是STL所有的东西，容器、算法、和允许算法工作在容器中的元素上的iterator。算法以合适、标准的方法操作对象，并可通过iterator得到容器精确的长度。一旦做了这些，它们就在也不会“跑出边界”。还有一些其他的对这些核心组件类型有功能性增强的组件，例如函数对象。我们将会看到有关这些的例子，现在，我们先看一看STL的list。