

附录 C

STL 简述

STL 即标准模板库 (Standard Template Library)，是 C++ 标准中最重要的组成部分，接下来的文字会让读者对它有个快速但比较全面的了解。

STL 的历史

STL 最初并不是 C++ 的一部分，甚至都不是用 C++ 写成的。

惠普实验室的 Alexander Stepanov (即 STL 之父) 在 20 世纪 80 年代使用 Ada 语言实现了一个泛型算法库——也就是 STL 的前身。在 C++ 增加了 template 特性后，Alexander Stepanov 与同在惠普实验室的 Meng Lee 合力，把 STL 用 C++ 重新改写，形成了 STL 的最初形态^①。1994 年，在 Bjarne Stroustrup (C++ 之父) 和 Andy Koenig (C++ 社区第二号人物) 的极力推荐下，C++ 标准草案在最后截至期限前接纳 STL 进入了 C++ 标准化进程。1998 年 C++ 标准正式通过，STL 也因此正式成为 C++ 标准的一部分。

STL 运用模板技术把数据结构与算法分离，再使用迭代器把两者粘结在一起，为程序架构提供了最大的弹性。它是一个划时代的作品，是第一个将泛型编程发挥得淋漓尽致的产品，其先进的设计思想、优雅的程序架构深刻地影响了 C++ 标准库结构和 C++ 开发范式。时至 21 世纪的第二个十年，每一个 C++ 程序员都应当像熟悉 if、for 等 C++ 基本元素一样熟悉 STL 的使用。

STL 主要有六个组成部分：容器、算法、迭代器、函数对象、适配器和内存分配器。

容器

这也许是 STL 中最引人注意的部分，它实现了计算机科学领域中数个最重要最基本的数据

^① 有一种说法，STL 实际上是两位作者 STepanov 和 Lee 名字的缩写。

结构。

STL 容器可分为两大类：序列式容器和关联式容器。

序列式容器包括 `vector`、`deque` 和 `list`，分别实现了动态数组、双端队列和双向链表。关联式容器包括 `set`、`multi_set`、`map` 和 `multi_map`，分别实现了基于二叉树的集合和映射。

算法

STL 共包含近百个算法，也都是计算机科学中的经典算法，分为非变动型算法、变动型算法、移除型算法、变序型算法、排序算法、已序区间算法和数值算法共七大类。其中即有简单的 `for_each`、`copy`、`count`、`fill`，也有复杂的 `sort`、`find`。

使用 STL 算法，我们不需要编写循环语句就可以高效地完成同样的工作。

迭代器

迭代器是 C/C++ 指针的泛化，也是迭代器模式的最佳实践。通过操作符重载和模板技术，迭代器支持 `++`、`--` 等操作符，并可以如指针一样操作容器元素。

STL 中的迭代器分为输入迭代器、输出迭代器、前向迭代器、双向迭代器和随机迭代器共五类。

函数对象

早期函数对象又被称为仿函数（functor），是指重载了 `operator()` 操作符的类。其行为类似普通函数，但拥有普通函数不具有的优点，可以称为“智能函数”。函数对象主要用于搭配容器和算法共同工作。

适配器

适配器运用适配器模式对前述的容器、迭代器、函数对象进行适配，使程序代码可以更加灵活地组合。容器的适配器有 `stack`、`queue` 和 `priority_queue`，迭代器的适配器有逆向迭代器、插入迭代器和流迭代器等，函数对象的适配器有标准适配器和函数适配器，如 `not1`、`mem_fun` 等。

内存分配器

内存分配器用来解决不同平台的内存模型差异问题，把底层的内存分配与释放以策略模式封装，主要供 STL 内部实现使用，对 STL 用户几乎是透明的。

STL 的实现

STL 只是一个标准规范，并不是一个真正的实体，因此存在着很多个不同的具体实现，例如 `libc++` (Apple)、`libstdc++` (GNU)、Dinkumware STL (用于 VC)、`STLport` (高度可移植)、`libstdcxx`

Boost 程序库完全开发指南（第二版）

（Apache）等等。一个好的、符合标准的 STL 实现能够大大地提高 C++ 程序员的开发效率。

版权所有