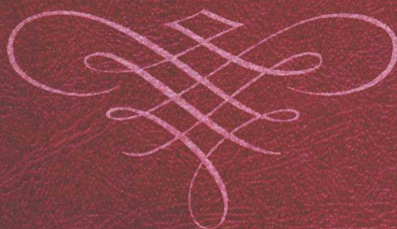


中国高等学校计算机科学与技术专业（应用型）规划教材

丛书主编 陈明

C++ STL基础及应用

金百东 刘德山 编著



清华大学出版社

中国高等学校计算机科学与技术专业（应用型）规划教材

丛书主编 陈明

C++ STL基础及应用

金百东 刘德山 编著



清华大学出版社

北京

内 容 简 介

本书全面系统地介绍标准模板库(STL)泛型应用开发技术。基础知识部分包括模板、迭代器、输入输出流、字符串、函数对象、通用容器、非变异算法、变异算法、排序等;集成应用部分包括 STL 算法的综合应用、在数据结构中的应用、在 Visual C++ 上的应用等。本书从应用出发,每章都包含大量的示例和详细的结果分析,旨在使读者学会 STL 各个知识体系的应用方法,体会 STL 思维的巧妙之处。对某些稍难示例的设计思想也做了详细的说明。

本书可作为专业技术人员、大专院校计算机专业的本科生、研究生学习 C++ 泛型编程的教材或参考书。本书对编写 Java 泛型程序也有一定的指导意义。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C++ STL 基础及应用/金百东,刘德山编著. —北京:清华大学出版社,2010.10

(中国高等学校计算机科学与技术专业(应用型)规划教材)

ISBN 978-7-302-22954-4

I. ①C… II. ①金… ②刘… III. ①C 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 105401 号

责任编辑:谢 琛 赵晓宁

责任校对:白 蕾

责任印制:王秀菊

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:24

字 数:567 千字

版 次:2010 年 10 月第 1 版

印 次:2010 年 10 月第 1 次印刷

印 数:1~3000

定 价:36.00 元

产品编号:036456-01

编委 会

主任：陈 明

副主任：蒋宗礼 卢先和

委员：常 虹 陈国君 陈 峻 陈晓云 陈芙蓉
丛 琳 方路明 段友祥 高文胜 巩君华
关 永 郭 禾 郝 莹 何胜利 何晓新
贺安坤 胡巧多 李陶深 李仲麟 刘东升
刘贵龙 刘晓强 刘振华 路 游 马杰良
毛国君 苗凤君 宁 玲 施海虎 宋长龙
宋立军 孙践知 孙中胜 汤 庸 田俊峰
万本庭 王让定 王锁柱 王 新 王兆青
王智广 王志强 谢 琛 谢书良 徐孝凯
徐子珊 杨建刚 姚 琳 叶春蕾 叶俊民
袁 薇 张建林 张 杰 张 武 张晓明
张艳萍 周 苏 曾 一 訾秀玲

序 言

应用是推动学科技术发展的原动力,计算机科学是实用科学,计算机科学技术广泛而深入地应用推动了计算机学科的飞速发展。应用型创新人才是科技人才的一种类型,应用型创新人才的重要特征是具有强大的系统开发能力和解决实际问题的能力。培养应用型人才的教学理念是教学过程中以培养学生的综合技术应用能力为主线,理论教学以够用为度,所选择的教学方法与手段要有利于培养学生的系统开发能力和解决实际问题的能力。

随着我国经济建设的发展,对计算机软件、计算机网络、信息系统、信息服务和计算机应用技术等专业技术方向的人才的需求日益增加,主要包括软件设计师、软件评测师、网络工程师、信息系统监理师、信息系统管理工程师、数据库系统工程师、多媒体应用设计师、电子商务设计师、嵌入式系统设计师和计算机辅助设计师等。如何构建应用型人才培养的教学体系以及系统框架,是从事计算机教育工作者的责任。为此,中国计算机学会计算机教育专业委员会和清华大学出版社共同组织启动了《中国高等学校计算机科学与技术专业(应用型)学科教程》的项目研究。参加本项目的研究人员全部来自国内高校教学一线具有丰富实践经验的专家和骨干教师。项目组对计算机科学与技术专业应用型学科的培养目标、内容、方法和意义,以及教学大纲和课程体系等进行了较深入、系统的研究,并编写了《中国高等学校计算机科学与技术专业(应用型)学科教程》(简称《学科教程》)。《学科教程》在编写上注意区分应用性人才与其他人才在培养上的不同,注重体现应用型学科的特征。在课程设计中,《学科教程》在依托学科设计的同时,更注意面向行业产业的实际需求。为了更好地体现《学科教程》的思想与内容,我们组织编写了《中国高等学校计算机科学与技术专业(应用型)规划教材》,旨在能为计算机专业应用型教学的课程设置、课程内容以及教学实践起到一个示范作用。本系列教材的主要特点如下:

1. 完全按照《学科教程》的体系组织编写本系列教材,特别是注意在教材设置、教材定位和教材内容的衔接上与《学科教程》保持一致。

2. 每门课程的教材内容都按照《学科教程》中设置的大纲精心编写,尽量体现应用型教材的特点。

3. 由各学校精品课程建设的骨干教师组成作者队伍,以课程研究为基础,将教学的研究成果引入教材中。

4. 在教材建设上,重点突出对计算机应用能力和应用技术的培养,注重教材的实践性。

5. 注重系列教材的立体配套,包括教参、教辅以及配套的教学资源、电子课件等。

高等院校应培养能为社会服务的应用型人才,以满足社会发展的需要。在培养模式、教学大纲、课程体系结构和教材都应适应培养应用型人才的目标。教材体现了培养目标和育人模式,是学科建设的结晶,也是教师水平的标志。本系列教材的作者均是多年从事计算机科学与技术专业教学的教师,在本领域的科学研究与教学中积累了丰富的经验,他们将教学研究和科学研究的成果融入教材中,增强了教材的先进性、实用性和实践性。

目前,我们对于应用型人才培养的模式还处于探索阶段,在教材组织与编写上还会有这样或那样的缺陷,我们将不断完善。同时,我们也希望广大应用型院校的教师给我们提出更好的建议。

《中国高等学校计算机科学与技术专业(应用型)规划教材》主编



2008年7月

前 言

STL(Standard Template Library, 标准模板库)是 C++ 泛型标准化内容的重要组成部分,主要由容器、迭代器和算法三部分组成,其中封装了数据结构中的绝大部分内容。运用 STL 开发应用程序可以使我们共享各种容器及算法,避免了低层次的各种容器及常用算法的反复开发。在代码一致性、升级、维护等方面都有很大的优越性。因此,学习 STL 是进行深层次开发 C++ 应用程序的重要途径。但是,目前市场上关于 STL 的书籍很多是译著,在思考方法上可能与我们的学生不一致,学习起来很吃力。所以,本书力求把多年的 STL 编程经验按照学生的思维方式进行编排,希望学生们能很快学会 STL 泛型编程方法,体会 STL 泛型编程的乐趣。

全书共分 11 章,第 1~10 章侧重于基础知识部分,第 11 章侧重于综合应用部分,内容如下。

第 1 章介绍 STL 历史,研究的主要内容,以及本书示例中用到的开发环境。

第 2 章通过示例说明 STL 中的内存管理思想,重要的 traits 模板技术,模板与操作符重载的关系。

第 3 章介绍 STL 中引入迭代器的原因,并通过自定义迭代器示例加深理解迭代器的内涵。

第 4 章介绍标准输入输出流、文件输入输出流、字符串输入输出流。

第 5 章介绍字符串创建方式及增、删、改、查等常用功能应用方法。

第 6 章介绍引入函数对象的原因,系统函数对象有哪些,自定义函数对象应用方法。

第 7 章介绍 vector、deque、list、queue、stack、priority_queue、bitset、set 和 map 等通用容器的用法,并强调了容器适配器的作用。

第 8~10 章主要是讲算法。第 8 章介绍非变异算法,包括循环、查询、计数、比较等功能;第 9 章介绍变异算法,包含复制、交换、变换、替换、填充、生成、删除、唯一、反转、环移、随机、划分等功能;第 10 章介绍排序及相关操作的算法。

第 11 章侧重于集成应用,包括算法综合应用、在数据结构中应用、在 Visual C++ 中应用三部分。算法综合应用主要介绍在多变态、文件解析、综合查询中的 STL 应用方法;在数据结构中应用介绍全排列、频度、最长公共子序列、大整型数加法、乘法、矩阵、回溯、字符串表达式、图中的 STL 应用方法。在 Visual C++ 中应用介绍用 STL 容器存储绘图信息,容器+算法实现数据保存与查询问题,并介绍 STL 与动态链接库的接口问题等。

本书第 2、3、6、7、10、11 章由金百东编写,第 1、4、5、8、9 章由刘德山编写。因本书程序较多,全书变量均用正体。

本书内容循序渐进,示例丰富,第 1~10 章的所有示例复制下来编译后就可以运行。第 11 章某些程序由于较大,做了简化处理。示例结果都做了必要的说明,对一些稍难的题目,对其设计思想也做了相应的论述,帮助读者加深对 STL 的理解。

由于作者水平有限,时间紧迫,书中难免有疏漏之处,恳请广大读者批评指正,不胜感激。

编 者
2010 年 6 月

目 录

第 1 章 STL 概述	1
1.1 STL 历史	1
1.2 STL 内容	2
1.3 建立 STL 程序的方法	3
1.4 命名空间	5
第 2 章 模板	7
2.1 通过模板初识 STL 思维	7
2.2 traits 技术	10
2.3 模板与操作符重载	14
第 3 章 迭代器	19
3.1 什么是迭代器	19
3.2 迭代器类位置	24
3.3 进一步理解迭代器	27
3.4 STL 迭代器	28
第 4 章 输入输出流	33
4.1 标准输入输出流	33
4.1.1 插入符与提取符	33
4.1.2 get 系列函数	35
4.1.3 处理流错误	36
4.2 文件输入输出流	38
4.2.1 文件打开	38
4.2.2 文件关闭	38
4.2.3 文件读写	38
4.3 字符串输入输出流	43
4.4 综合示例	44

第 5 章 字符串	49
5.1 字符串创建及初始化	49
5.1.1 基本创建方式	49
5.1.2 迭代器创建方式	50
5.2 字符串操作	50
5.2.1 插入操作	50
5.2.2 替换操作	51
5.3 字符串查询	52
5.4 字符串中删除字符	54
5.5 字符串比较	54
5.6 综合示例	55
第 6 章 函数对象	61
6.1 简介	61
6.1.1 为何引入函数对象	61
6.1.2 函数对象分类	62
6.1.3 简单示例	63
6.2 一元函数	64
6.3 二元函数	66
6.4 系统函数对象	68
6.4.1 算术类函数对象	69
6.4.2 关系运算类函数对象	72
6.4.3 逻辑运算类函数对象	74
6.4.4 函数适配器	74
6.5 综合示例	79
第 7 章 通用容器	83
7.1 概述	83
7.1.1 容器分类	83
7.1.2 容器共性	84
7.1.3 容器比较	85
7.2 vector 容器	85
7.2.1 概述	85
7.2.2 初始化示例	86
7.2.3 增加及获得元素示例	88
7.2.4 修改元素示例	92

7.2.5	删除元素示例	93
7.2.6	进一步理解 vector	94
7.2.7	综合操作示例	95
7.3	deque 容器	99
7.3.1	常用函数	99
7.3.2	基本操作示例	100
7.3.3	综合操作示例	102
7.4	list 容器	104
7.4.1	常用函数	105
7.4.2	基本操作示例	106
7.4.3	综合操作示例	109
7.5	队列和堆栈	115
7.5.1	常用函数	115
7.5.2	容器配接器	116
7.5.3	基本操作示例	117
7.5.4	综合操作示例	120
7.6	优先队列	123
7.6.1	常用函数	123
7.6.2	基本操作示例	124
7.6.3	综合操作示例	125
7.7	bitset 容器	128
7.7.1	常用函数	128
7.7.2	基本操作示例	129
7.7.3	综合操作示例	132
7.8	集合	135
7.8.1	常用函数	135
7.8.2	基本操作示例	136
7.8.3	综合操作示例	139
7.9	映射	142
7.9.1	常用函数	142
7.9.2	基本操作示例	143
7.9.3	综合操作示例	145
7.10	再论迭代器	150
第 8 章 非变异算法		155
8.1	循环	155
8.1.1	主要函数	155
8.1.2	示例分析	156

8.2	查询	160	
8.2.1	主要函数	160	
8.2.2	示例分析	163	
8.3	计数	171	
8.3.1	主要函数	171	
8.3.2	示例分析	172	
8.4	比较	174	
8.4.1	主要函数	174	
8.4.2	示例分析	175	

第9章 变异算法 179

9.1	复制	180	
9.1.1	主要函数	180	
9.1.2	示例分析	181	
9.2	交换	182	
9.2.1	主要函数	182	
9.2.2	示例分析	183	
9.3	变换	184	
9.3.1	主要函数	184	
9.3.2	示例分析	185	
9.4	替换	188	
9.4.1	主要函数	188	
9.4.2	示例分析	190	
9.5	填充	191	
9.5.1	主要函数	191	
9.5.2	示例分析	192	
9.6	生成	193	
9.6.1	主要函数	193	
9.6.2	示例分析	194	
9.7	删除	199	
9.7.1	主要函数	199	
9.7.2	示例分析	200	
9.8	唯一	204	
9.8.1	主要函数	204	
9.8.2	示例分析	205	
9.9	反转	207	
9.9.1	主要函数	207	

9.9.2	示例分析	208
9.10	环移	209
9.10.1	主要函数	209
9.10.2	示例分析	210
9.11	随机	212
8.11.1	主要函数	212
9.11.2	示例分析	212
9.12	划分	215
9.12.1	主要函数	215
9.12.2	示例分析	216
第 10 章	排序及相关操作	219
10.1	排序	220
10.1.1	主要函数	220
10.1.2	示例分析	222
10.2	第 n 个元素	227
10.2.1	主要函数	227
10.2.2	示例分析	227
10.3	二分检索	228
10.3.1	主要函数	228
10.3.2	示例分析	230
10.4	归并	232
10.4.1	主要函数	232
10.4.2	示例分析	233
10.5	有序结构上的集合操作	234
10.5.1	主要函数	234
10.5.2	示例分析	236
10.6	堆操作	241
10.6.1	主要函数	241
10.6.2	示例分析	242
10.7	最大和最小	245
10.7.1	主要函数	245
10.7.2	示例分析	247
10.8	词典比较	247
10.8.1	主要函数	247
10.8.2	示例分析	248
10.9	排列生成器	249

10.9.1	主要函数	249
10.9.2	示例分析	250
10.10	数值算法	251
10.10.1	主要函数	251
10.10.2	示例分析	253
10.11	自定义 STL 风格函数	254
第 11 章	STL 应用	257
11.1	算法的综合运用	257
11.1.1	在多态中的应用	257
11.1.2	set、map 应用	261
11.1.3	ini 文件解析	264
11.1.4	综合查询	269
11.2	在数据结构中的应用	280
11.2.1	全排列应用	280
11.2.2	频度问题	283
11.2.3	最长公共子序列问题	285
11.2.4	大整型数加法、乘法类	288
11.2.5	矩阵问题	293
11.2.6	回溯问题	296
11.2.7	字符串表达式	300
11.2.8	图	306
11.3	在 Visual C++ 中应用	316
11.3.1	Scribble 绘图程序	317
11.3.2	数据库操作程序	324
11.3.3	文本文件排序、查询	337
11.3.4	基于配置文件的查询程序	346
11.3.5	STL 与动态链接库	360
参考文献		369

第 1 章 STL 概述

1.1 STL 历史

20 世纪 70 年代,被誉为标准模板库(Standard Template Library, STL)之父的 Alexander Stepanov 开始考虑:在保证效率的前提下,将算法从诸多具体应用之中抽象出来的可能性。为了验证自己的思想,他和纽约州立大学教授 Deepak Kapur,伦塞里尔技术学院教授 David Musser 共同开发了一种叫做 Tecton 的语言。尽管这次尝试没有取得实用性的成果,但却给了 Stepanov 很大的启示。

在随后的几年中,他又和 David Musser 等人先后用 Schema 语言(一种 Lisp 语言的变种)和 Ada 语言建立了一些大型程序库。Alexander Stepanov 逐渐意识到:在当时的面向对象程序设计思想中存在一些问题,比如抽象数据类型概念所存在的缺陷。Stepanov 希望通过对软件领域中各组成部分的分类,逐渐形成一种软件设计的概念性框架。

1987 年,在贝尔实验室工作的 Alexander Stepanov 开始首次采用 C++ 语言进行泛型软件库的研究。由于当时的 C++ 语言还没有引入模板(template)语法,研究方法只能采用继承机制。尽管如此,Stepanov 还是开发出了一个庞大的算法库。与此同时,在与 Andrew Koenig(前 ISO C++ 标准化委员会主席)和 Bjarne Stroustrup(C++ 语言的创始人)等顶级大师们的共事过程中,Stepanov 开始注意到 C/C++ 语言在实现其泛型思想方面所具有的潜在优势。就拿 C/C++ 中的指针而言,它的灵活与高效运用使后来的 STL 在实现泛型化的同时更是保持了高效率。另外,在 STL 中占据极其重要地位的迭代子概念便是源自于 C/C++ 中原生指针(native pointer)的抽象。

1988 年,Alexander Stepanov 开始进入惠普的 Palo Alto 实验室工作,在随后的 4 年中,他从事的是有关磁盘驱动器方面的工作。直到 1992 年,由于参加并主持了实验室主任 Bill Worley 所建立的一个有关算法的研究项目,才使他重新回到了泛型化算法的研究工作上。项目自建立之后,参与者从最初的 8 人逐渐减少,最后只剩下两个人——Stepanov 和 Meng Lee。经过长时间的努力,最终完成了一个包含有大量数据结构和算法部件的庞大运行库。这便是现在 STL 的雏形。

1993年,当时在贝尔实验室的 Andrew Koenig 看到了 Stepanov 的研究成果,在他的鼓励与帮助下,Stepanov 于 1993 年 9 月在圣何塞为 ANSI/ISO C++ 标准委员会做了一个相关演讲(题为“The Science of C++ Programming”),向委员们讲述了其观念。然后又于 1994 年 3 月,在圣迭戈会议上向委员会提交了一份建议书,以期使 STL 成为 C++ 标准库的一部分。尽管这一建议十分庞大,以至于降低了被通过的可能性,但其所包含的新思想吸引了许多人的注意力。

随后,在众人的帮助之下,包括 Bjarne Stroustrup 在内,Stepanov 又对 STL 进行了改进。同时加入了一个封装内存模式信息的抽象模块,也就是现在 STL 中的 allocator,它使 STL 的大部分实现都可以独立于具体的内存模式,从而独立于具体平台。在 1994 年的滑铁卢会议上,委员们最终通过了提案,决定将 STL 正式纳入 C++ 标准化进程之中,随后 STL 便被放进了会议的工作文件中。自此,STL 终于成为 C++ 家族中的重要一员。

此后,随着 C++ 标准的不断改进,STL 也在不断地做着相应的演化。直至 1998 年,ANSI/ISO C++ 标准正式定案,STL 始终是 C++ 标准中不可或缺的一大部件。

1.2 STL 内容

STL 主要包含容器、算法和迭代器三大部分。

STL 容器包含了绝大多数数据结构,如数组、链表、队列、堆、栈和树等。开发者直接应用这些系统 STL 容器相关函数就可以了,而且这些函数都是带模板参数的,可以适应许多数据元素类型,功能非常强大。STL 算法包含了诸如增、删、改、查和排序等系统函数,开发者可以直接操作这些函数实现相应功能。STL 迭代器类似指针,通过它的有序移动把容器中的元素与算法关联起来,它是实现所有 STL 功能的基础所在。

当然,STL 也包含其他一些内容,如字符串、输入输出流等内容。

本书是以 Visual C++ 中内嵌的 STL 为基础讲述的,开发环境选择 Visual C++ 6.0 或 Visual Studio.net 都可以,用到的主要头文件如表 1.1 所示。

表 1.1 常用 STL 包含文件

索引	功能	包含文件	备注
1	迭代器	#include <iterator>	
2	输入输出流	#include <iostream>	标准输入输出流
		#include <fstream>	文件输入输出流
		#include <sstream>	字符串输入输出流
3	字符串	#include <string>	
4	函数对象	#include <functional>	

续表

索引	功能	包含文件	备注
5	通用容器	# include <vector>	向量容器
		# include <deque>	双端队列
		# include <list>	链表容器
		# include <queue>	队列、优先队列
		# include <stack>	堆栈
		# include <set>	集合、多集合、位集合
		# include <map>	映射、多映射
6	通用算法	# include <algorithm>	
7	数值算法	# include <numeric>	

STL 包含文件都不包含扩展名,存放位置一般是“VC 安装目录\include\”。

1.3 建立 STL 程序的方法

建立 STL 程序的方法有很多,下面介绍一种最常用的方式。以 Visual C++ 6.0 开发平台为例。

(1) 选择“文件”→“新建”命令,如图 1.1 所示。

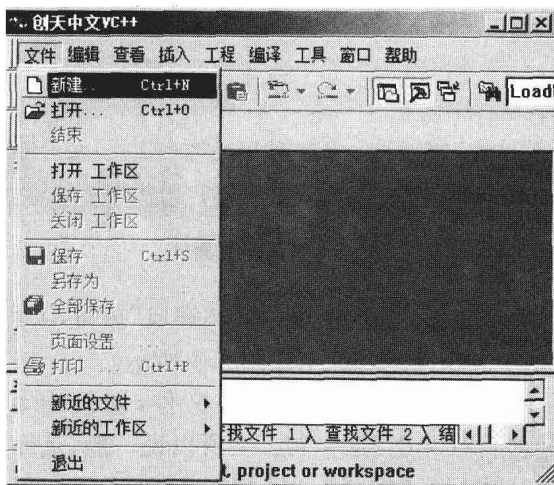


图 1.1 操作界面 1

(2) 出现图 1.2 所示“新建”对话框,选择“工程”选项卡,类型是 Win32 Console Application,输入工程名如 MyStl,单击“确定”按钮,出现图 1.3 所示对话框。