

<<C++STL基础及应用>>

图书基本信息

书名：<<C++STL基础及应用>>

13位ISBN编号：9787302229544

10位ISBN编号：7302229546

出版时间：2010-10

出版时间：清华大学出版社

作者：金百东，刘德山 编著

页数：368

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<C++STL基础及应用>>

前言

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學理念是教學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，所选择的教學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學体系以及系统框架，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。

<<C++STL基础及应用>>

内容概要

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。

<<C++STL基础及应用>>

书籍目录

| | | | | | | | | |
|--------------|-----------------|-----|-------------------|--------|---------------------|---------------|---------------------|----------------|
| 第1章 STL概述 | 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 | 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 | 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 | 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 | 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 | 5.1 字符串创建及初始化 | 49 | 5.1.1 基本创建方式 |
| | | 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 |
| | | 54 | 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 |
| | | 55 | 5.6 综合示例 | 55 | 第6章 函数对象 | 6.1 简介 | 61 | 6.1.1 为何引入函数对象 |
| | | 62 | 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 |
| | | 66 | 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 |
| | | 72 | 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 |
| | | 79 | 第7章 通用容器 | 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 | 7.2.1 概述 | 85 | 7.2.2 初始化示例 |
| | | 88 | 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 |
| | | 94 | 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 |
| | | 100 | 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 |
| | | 104 | 7.4 list容器 | 104 | 7.4.1 常用函数 | 105 | 7.4.2 基本操作示例 | 106 |
| | | 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 |
| | | 123 | 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 |
| | | 124 | 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 |
| | | 128 | 7.7 bitset容器 | 128 | 7.7.1 常用函数 | 128 | 7.7.2 基本操作示例 | 129 |
| | | 132 | 7.8 集合 | 135 | 7.8.1 常用函数 | 135 | 7.8.2 基本操作示例 | 136 |
| | | 142 | 7.9 映射 | 142 | 7.9.1 常用函数 | 142 | 7.9.2 基本操作示例 | 143 |
| | | 145 | 7.10 再论迭代器 | 150 | 第8章 非变异算法 | 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 |
| | | 163 | 8.3 计数 | 171 | 8.3.1 主要函数 | 171 | 8.3.2 示例分析 | 172 |
| | | 174 | 8.4 比较 | 174 | 8.4.1 主要函数 | 174 | 8.4.2 示例分析 | 175 |
| | | 180 | 第9章 变异算法 | 9.1 复制 | 180 | 9.1.1 主要函数 | 180 | 9.1.2 示例分析 |
| | | 182 | 9.2 交换 | 182 | 9.2.1 主要函数 | 182 | 9.2.2 示例分析 | 183 |
| | | 184 | 9.3 变换 | 184 | 9.3.1 主要函数 | 184 | 9.3.2 示例分析 | 185 |
| | | 188 | 9.4 替换 | 188 | 9.4.1 主要函数 | 188 | 9.4.2 示例分析 | 190 |
| | | 191 | 9.5 填充 | 191 | 9.5.1 主要函数 | 191 | 9.5.2 示例分析 | 192 |
| | | 193 | 9.6 生成 | 193 | 9.6.1 主要函数 | 193 | 9.6.2 示例分析 | 194 |
| | | 199 | 9.7 删除 | 199 | 9.7.1 主要函数 | 199 | 9.7.2 示例分析 | 200 |
| | | 204 | 9.8 唯一 | 204 | 9.8.1 主要函数 | 204 | 9.8.2 示例分析 | 205 |
| | | 207 | 9.9 反转 | 207 | 9.9.1 主要函数 | 207 | 9.9.2 示例分析 | 208 |
| | | 209 | 9.10 环移 | 209 | 9.10.1 主要函数 | 209 | 9.10.2 示例分析 | 210 |
| | | 212 | 9.11 随机 | 212 | 9.11.1 主要函数 | 212 | 9.11.2 示例分析 | 212 |
| | | 215 | 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 | 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 |
| | | 230 | 10.3.2 示例分析 | 230 | 10.4 归并 | 232 | 10.4.1 主要函数 | 232 |
| | | 234 | 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 |
| | | 236 | 10.5.2 示例分析 | 236 | 10.6 堆操作 | 241 | 10.6.1 主要函数 | 241 |
| | | 242 | 10.6.2 示例分析 | 242 | 10.7 最大和最小 | 245 | 10.7.1 主要函数 | 245 |
| | | 247 | 10.7.2 示例分析 | 247 | 10.8 词典比较 | 247 | 10.8.1 主要函数 | 247 |
| | | 248 | 10.8.2 示例分析 | 248 | 10.9 排列生成器 | 249 | 10.9.1 主要函数 | 249 |
| | | 250 | 10.9.2 示例分析 | 250 | 10.10 数值算法 | 251 | 10.10.1 主要函数 | 251 |
| | | 253 | 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 | 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 |
| | | 269 | 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 |
| | | 283 | 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 |
| | | 293 | 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 |
| | | 306 | 11.2.8 图 | 306 | 11.3 在Visual C++中应用 | 316 | 11.3.1 Scribble绘图程序 | 317 |
| | | 324 | 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 |
| | | 360 | 11.3.5 STL 动态链接库 | 360 | 参考文献 | | | |

<<C++STL基础及应用>>

章节摘录

(1) 序列性容器：按照线性排列来存储某类型值的集合，每个元素都有自己特定的位置，顺序容器主要有vector、deque和list。

vector：就是动态数组。

它是在堆中分配内存，元素连续存放，有保留内存，如果减少大小后内存也不会释放。

新值大于当前大小时才会再分配内存。

对最后元素操作最快（在后面添加删除最快），此时一般不需要移动内存。

只有保留内存不够时，才需要对中间和开始处进行添加删除元素操作，这时需要移动内存，如果你的元素是结构或是类，那么移动的同时还会进行构造和析构操作。

vector的一大特点是可直接访问任何元素。

deque：与vector类似，支持随机访问和快速插入删除，它在容器中某一位置上的操作所花费的是线性时间。

与vector不同的是，deque还支持从开始端插入、删除数据。

由于它主要对前端、后端进行操作，因此也叫做双端队列。

list：又叫链表，是一种双线性列表，只能顺序访问（从前向后或者从后向前），与前面两种容器类有一个明显的区别就是它不支持随机访问。

要访问表中某个下标处的项需要从表头或表尾处（接近该下标的一端）开始循环。

<<C++STL基础及应用>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>