

书后附有包含了书中全部代码的磁盘，极具实用价值!

# 学 用

## Visual C++ 5

[美] Michael Hyman 著  
Bob Arnson

马岚 韩传钊 崔新军 张谦 等译  
马岚 王森 审校

- Visual C++轻松愉快编程——全新内容!
- 掌握C++基本内容的首选工具
- 用最易懂的语言告诉你出了问题该怎么办



电子工业出版社

Publishing House Of Electronics Industry  
URL:<http://www.phei.com.cn>

免费Visual C++ 5速查表

# 学用 Visual C++ 5

[美] Michael Hyman 著  
Bob Arnson

马 岚 韩传钊 等译  
崔新军 张 谦

马 岚 王 森 审校

电子工业出版社  
Publishing House of Electronics Industry

## 内容简介

本书是一本面向 Visual C++ 初学者的使用指南,将引导你逐渐熟悉并能够运用它编写自己的代码行。本书共分四部分:第一部分是 Visual C++ 快速入门,帮你安装、运行并介绍 Visual C++ 的主要特征;第二部分重点讲解 C++ 编程基础知识(其中许多知识同样适用于 C);第三部分侧重于面向对象编程的内容,包括类、模板以及其他许多 C++ 的特性;最后的十准则集粹里汇集了初级 C++ 用户常会遇到的各种问题的解决方法和诀窍,另外还有一些常用的备查信息。本书所附软盘提供了书中全部范例程序的源代码。

**Visual C++ 5 For Dummies** by Michael Hyman & Bob Armon



Copyright ©1998 by Publishing House of Electronics Industry

Original English language edition copyright ©1996 by IDG Books Worldwide, Inc.

All rights reserved including the right of reproduction in whole or in part in any form.

This edition published by arrangement with the original publisher, IDG Books Worldwide, Inc., Foster City, California, USA.

For Dummies is a trademark of International Data Group.

本书中文简体专有翻译出版权由美国 IDG Books Worldwide, Inc. 公司授予电子工业出版社及其所属今日电子杂志社。未经许可,不得以任何手段和形式复制或抄袭本书内容。该专有出版权受法律保护,侵权必究。

书 名: 学用 Visual C++ 5

著 者: [美] Michael Hyman & Bob Armon

译 者: 马岚 韩传制 崔新军 张谦 等

审 校 者: 马岚 王森

责任编辑: 王玉国

特约编辑: 周宏敏

印 刷 者: 北京大中印刷厂

出版发行: 电子工业出版社出版、发行 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

经 销: 各地新华书店经销

开 本: 787 × 1092 1/16 印张: 24.5 字数: 588 千字

版 次: 1998 年 6 月第 1 版 1998 年 6 月第 1 次印刷

书 号: ISBN 7-5053-4301-7  
TP·1955

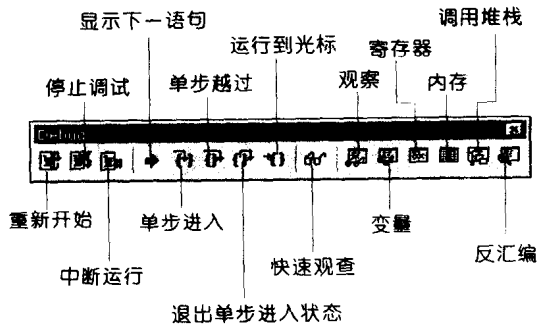
定 价: 51.00 元(含磁盘 1 张)

著作权合同登记号 图字: 01-97-0946

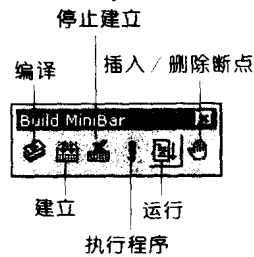
凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换  
版权所有·翻印必究

# 《学用 Visual C++ 5》速查表

## Debug 工具条



## Build MiniBar



## Visual C++ 键盘快捷键

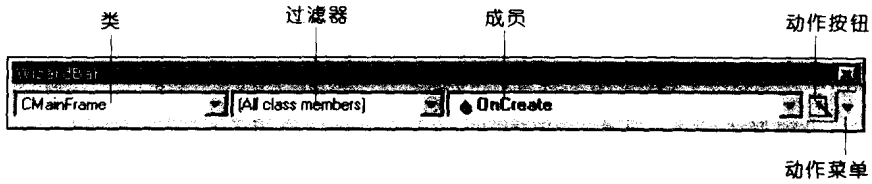
|                 |            |
|-----------------|------------|
| 建立项目            | F7         |
| 编译当前文件          | Ctrl + F7  |
| 非调试状态运行程序       | Ctrl + F5  |
| 调试程序            | F5         |
| 插入或删除断点         | F9         |
| 迅速观察一个变量        | Shift + F9 |
| 停止调试            | Shift + F5 |
| 单步越过            | F10        |
| 单步进入            | F11        |
| 获得关于关键词的帮助      | F1         |
| 切换到下一个代码窗口      | Ctrl + F6  |
| 关闭当前窗口          | Ctrl + F4  |
| 打开 Class Wizard | Ctrl + W   |

## 规则表达式匹配命令

|                 |       |
|-----------------|-------|
| 匹配任意单个字符        | .     |
| 匹配任意个字符         | *     |
| 匹配一个或多个前置字符     | +     |
| 匹配一行的开始         | ^     |
| 匹配一行的结束         | \$    |
| 匹配一个或多个指定的字符    | [字符]  |
| 除指定字符以外, 匹配所有字符 | [^字符] |

# 《学用 Visual C++ 5》速查表

## WizardBar



## 使用 AppWizard

1. 选 **File**→**New**。  
出现 **New** 对话框。这个对话框包含所能生成内容的列表，其中包括项目。
2. 选中 **Projects** 标签。
3. 选中列表中的 **MFC AppWizard(exe)**。
4. 在 **Project name** 输入项目名。
5. 单击 **OK**。
6. 使用 **AppWizard** 对话框决定程序使用 **AppWizard** 使用的外观，是否数据库一类的 **ActiveX** 用有什么高级 **AppWizard** 变生成的类。
7. 单击 **Finish** 按钮。  
出现 **New Project** 对话框，显示要信息。
8. 单击 **OK** 按钮。

## 生成一个新项目

1. 单击 **New**。
2. 单击 **Projects** 对话框。
3. 单击 **MFC AppWizard(exe)** **Project type** 列表。
4. 单击 **OK** 按钮，选择要创建的项目类型。
5. 在 **Project name** 域，输入项目名称。
6. 单击 **AppWizard** 域旁边的省略号按钮，从列表中选一个存放项。

## 译者序

你可能早已听说过 C++ 语言具有功能强大、应用面广、可移植性强的三大优势,并期盼着能用它编写实用程序和开发高质量的应用软件,但苦于找不到一条尽快掌握其编程的捷径。因为要熟悉 C++ 语言并不是一件容易的事,它是一种复杂的语言,它的许多规则、特性、面向对象的编程机制、容易混淆的概念和术语,学习起来都很费力。现在好了,我们向你推荐《学用 Visual C++ 5》这本书,它会教你安装和使用 Visual C++ 这个 C++ 实用程序开发工具。学习以后你会感到,原来编写 C++ 语言程序这样简单有趣,你竟能在几小时之内完成一个专业程序员需花几天才能编写出的程序。

这本书是为初学者写的,它好就好在不是先介绍 C++ 语言所有的细枝末节,然后再介绍如何去编写程序和提高编程技巧,而是择其要点,用读者容易理解的方式先讲述 Visual C++ 5.0 的基本用法,然后介绍 C++ 编程基本知识和面向对象编程的各种特性,并在最后一部分中列举了用户经常会遇到的各种问题和相应的解决方案。即使你毫无编程经验,也能很快掌握本书所讲的知识。当然,有基础的读者更轻松一些。

参加本书翻译的有:马岚(前言和第一部分)、韩传钊(第二部分)、崔新军(第三部分)、张谦(第四部分和附录)。全文由马岚、王森审校。

由于译者水平所限,难免出现差错,敬请读者指正。

译者

一九九七年九月

## 关于作者

Michael Hyman 在 Northwest 一个大公司负责多媒体技术,并且是 Windows Tech Journal 和 Microsoft Interactive developer 的专栏作家。Michael 还写过其他许多计算机书籍,其中包括 PC Roadkill、Borland C++ For Dummies 和 Visual C++ For Dummies。

Michael 在普林斯顿大学获电子工程和计算机科学学位。工作不忙的时候,他就收拾收拾房间、读读小说或者给孩子换换尿布。

Bob Arnson 是 PennWell Publishing 出版的 VC++ Professional Borland C++ Professional 和 VB Tech Journal 的高级编辑。在加盟 PennWell 之前,他是 Oakley Publishing 的高级编辑,再以前,他是 Borland 的 Technical Publications 小组里的一个作家。Bob 写过其他几本关于 QuickBASIC、Visual Basic 和 Borland C++ 的书。为了避免网上冲浪时的信号忙,Bob 把自己的大部分时间花在安装新电话线上。

# 目 录

|                                |             |
|--------------------------------|-------------|
| 前言                             | (1)         |
| 本书是关于 C++ 的,还是关于 Visual C++ 的呢 | (1)         |
| 本书特点                           | (2)         |
| 本书的读者范围                        | (2)         |
| 本书的编排                          | (2)         |
| 关于附带的磁盘                        | (2)         |
| 本书中的图标                         | (3)         |
| C++ 因何得名                       | (3)         |
| <b>第一部分 使用 Visual C++</b>      | <b>(5)</b>  |
| <b>第一章 Visual C++ 软件包中有什么</b>  | <b>(7)</b>  |
| 有时候,小包装里也有好东西                  | (7)         |
| 启动一个上千行程序的编译器                  | (8)         |
| 调试器                            | (8)         |
| 对话、菜单和位图                       | (9)         |
| 集成                             | (9)         |
| 建立在一个强大的应用框架基础上                | (10)        |
| 对于这些库你永远不必付过期罚款                | (10)        |
| Maxwell Smart 实用工具             | (11)        |
| 在环境中获得信息的正确方式                  | (11)        |
| 帮你起步的范例程序                      | (12)        |
| <b>第二章 安装 Visual C++</b>       | <b>(13)</b> |
| 你有足够的剩余空间吗                     | (13)        |
| 启动安装程序                         | (14)        |
| 注册                             | (15)        |
| 典型安装(或者是怎样立刻把磁盘装满)             | (16)        |
| 最小安装                           | (16)        |
| 骨架安装(需要 CD)                    | (16)        |



|                                      |             |
|--------------------------------------|-------------|
| 个人化的自定义安装 .....                      | (17)        |
| 关于“我将不会常用 MFC 或者根本不用”的安装 .....       | (18)        |
| 但我不需要任何 MFC .....                    | (18)        |
| 关于“我将不会进行任何数据库编程”的安装 .....           | (18)        |
| 关于“我不需要 Maxwell Smart 实用工具”的安装 ..... | (18)        |
| 关于我不需要 Books Online”的安装 .....        | (18)        |
| 关于我不需要 ActiveX 控件”的安装 .....          | (19)        |
| 预备,各就各位,开始 .....                     | (19)        |
| 等一等! 我改主意了 .....                     | (19)        |
| 如果它不安装该怎么办 .....                     | (19)        |
| <b>第三章 生成 Windows 程序 .....</b>       | <b>(21)</b> |
| 到了启动 Visual C++ 的时候了 .....           | (21)        |
| 你要问问自己,我怎样才能到这儿 .....                | (21)        |
| 偷偷摸摸的预演 .....                        | (22)        |
| 我们动身去看看向导 .....                      | (23)        |
| 使用 AppWizard 的前 4 步 .....            | (24)        |
| Step1:应用的类型 .....                    | (24)        |
| Step2:数据库支持 .....                    | (25)        |
| Step3:ActiveX 支持 .....               | (25)        |
| Step4:丰富的修饰内容 .....                  | (26)        |
| AppWizard 还能做什么 .....                | (27)        |
| 我想要一个迷人的标题 .....                     | (29)        |
| 我想让它干点什么 .....                       | (29)        |
| 开始写代码 .....                          | (30)        |
| 该运行了 .....                           | (32)        |
| 发生了什么,什么用了这么长时间 .....                | (32)        |
| 如果出了错该怎么办 .....                      | (34)        |
| 如果它总是在编译该怎么办 .....                   | (34)        |
| <b>第四章 被 ClassWizard 所陶醉 .....</b>   | <b>(35)</b> |
| 准备好一个 Wizard .....                   | (35)        |
| Wizards 始终在类的顶层 .....                | (36)        |
| 改变 About 框 .....                     | (36)        |
| 在对话框中增加新内容 .....                     | (38)        |
| 一个新的对话框 .....                        | (39)        |
| 给对话框指定一个类 .....                      | (40)        |
| 增加新的菜单项目 .....                       | (42)        |
| 最后会晤:挂接新对话框 .....                    | (43)        |

|                                |             |
|--------------------------------|-------------|
| 再运行一次 .....                    | (45)        |
| 自定义程序 .....                    | (46)        |
| <b>第五章 项目 .....</b>            | <b>(47)</b> |
| 为什么感到麻烦 .....                  | (47)        |
| 项目文件 .....                     | (48)        |
| 工作区和项目 .....                   | (48)        |
| 生成一个新的项目 .....                 | (48)        |
| 指定有关新项目的细节 .....               | (49)        |
| 给项目加入文件 .....                  | (50)        |
| 在 Workspace 窗口可做的普通工作 .....    | (51)        |
| 找出谁在依靠你 .....                  | (51)        |
| 这样设计看上去不错 .....                | (52)        |
| <b>第六章 编辑器 .....</b>           | <b>(53)</b> |
| 适合编辑的所有代码 .....                | (53)        |
| 编辑过程的主角 .....                  | (55)        |
| 充满活力的色彩显示 .....                | (56)        |
| 再单击一次鼠标 .....                  | (57)        |
| 查找代码块 .....                    | (58)        |
| 不要小看规则表达式匹配 .....              | (60)        |
| <b>第七章 编译 .....</b>            | <b>(61)</b> |
| 只编译一次就通过是比较难的 .....            | (61)        |
| 语法错误 .....                     | (62)        |
| 警告 .....                       | (63)        |
| 如果编译器知道有错误,为什么不能自己改正它 .....    | (64)        |
| 如果一开始没有成功,就再编译、建立、重建 .....     | (64)        |
| <b>第八章 调试器 .....</b>           | <b>(65)</b> |
| 语法错误与逻辑错误的区别是什么 .....          | (65)        |
| 调试过程的概略介绍 .....                | (66)        |
| 你叫什么名字? 你的调试器是什么? 它长得象我吗 ..... | (67)        |
| 以逻辑错误的名义停下来 .....              | (67)        |
| 单步运行 .....                     | (68)        |
| 当心出错 .....                     | (69)        |
| 可以改变错误的变量值 .....               | (70)        |
| 快速查看 .....                     | (70)        |
| 如果需要更快呢 .....                  | (71)        |

|                              |             |
|------------------------------|-------------|
| 纠正逻辑错误 .....                 | (71)        |
| 开始调试 .....                   | (72)        |
| 一大堆逻辑错误 .....                | (73)        |
| 程序员,停下来,每个人都看到这里有很多错误 .....  | (73)        |
| 我们还在那儿吗 .....                | (76)        |
| 还是有错误 .....                  | (76)        |
| 再来一次 .....                   | (77)        |
| 可以运行了 .....                  | (77)        |
| 清除一个断点和一个观察窗口 .....          | (77)        |
| 来一曲调试蓝调音乐 .....              | (78)        |
| 调试器的续篇 .....                 | (79)        |
| <br>                         |             |
| <b>第九章 购买? 还是浏览 .....</b>    | <b>(81)</b> |
| 浏览一个项目 .....                 | (81)        |
| 定义和引用:不需字典 .....             | (83)        |
| 这些秘密文字是什么 .....              | (84)        |
| 浏览器种类 .....                  | (85)        |
| 将浏览器放在可以得到的地方 .....          | (85)        |
| 使用源代码 .....                  | (86)        |
| 浏览家族树 .....                  | (86)        |
| 缩小查找范围 .....                 | (87)        |
| 别疯狂地走开——离开就行了 .....          | (88)        |
| 有时会不运行了 .....                | (88)        |
| <br>                         |             |
| <b>第十章 考虑所有的选项 .....</b>     | <b>(89)</b> |
| 选项 .....                     | (89)        |
| 一次改变一个文件 .....               | (91)        |
| 改变一两个选项 .....                | (92)        |
| 友好地对待环境 .....                | (93)        |
| 目录 .....                     | (93)        |
| 格式 .....                     | (93)        |
| 自定义工具条 .....                 | (94)        |
| <br>                         |             |
| <b>第二部分 关于 C++ 的使用 .....</b> | <b>(97)</b> |
| <br>                         |             |
| <b>第十一章 了解程序 .....</b>       | <b>(99)</b> |
| 程序的开始 .....                  | (100)       |
| 利用库文件 .....                  | (101)       |
| 如何利用程序解决问题 .....             | (101)       |
| 阅读自动电唱机程序 .....              | (102)       |

|                            |       |
|----------------------------|-------|
| <b>第十二章 面向对象编程介绍</b> ..... | (105) |
| 面向对象编程是如何进行的.....          | (105) |
| 这就是全部优点吗.....              | (106) |
| 封装.....                    | (108) |
| 继承.....                    | (108) |
| 多态性.....                   | (108) |
| <b>第十三章 编程基本知识</b> .....   | (111) |
| 在源文件中放入什么.....             | (112) |
| 从最基本的程序入手.....             | (112) |
| 如何用 C++ 实现显示输出 .....       | (114) |
| 换行表示法.....                 | (115) |
| 关于 cin .....               | (115) |
| 彩色注释.....                  | (116) |
| 从库里借一个函数.....              | (116) |
| Hello World 程序 .....       | (117) |
| 用 Visual C++ 设计程序 .....    | (118) |
| <b>第十四章 数据类型</b> .....     | (121) |
| 声明变量类型.....                | (121) |
| 基本数据类型.....                | (122) |
| 其他数据类型.....                | (123) |
| 类型安全性.....                 | (123) |
| 常量.....                    | (125) |
| 音乐实例.....                  | (125) |
| <b>第十五章 变量</b> .....       | (127) |
| 命名变量.....                  | (127) |
| 定义变量.....                  | (129) |
| 变量初始化.....                 | (130) |
| <b>第十六章 结构</b> .....       | (131) |
| 声明结构.....                  | (131) |
| 结构成员的使用.....               | (133) |
| 结构嵌套.....                  | (134) |
| 结构在程序中的应用.....             | (135) |
| 结构是不是很酷.....               | (136) |

|                                |       |
|--------------------------------|-------|
| <b>第十七章 表达式</b> .....          | (137) |
| 运算符.....                       | (138) |
| 高级运算符.....                     | (138) |
| ++ 运算符 .....                   | (139) |
| >> 运算符 .....                   | (139) |
| << 运算符 .....                   | (140) |
| 布尔表达式.....                     | (140) |
| 赋值.....                        | (141) |
| 复杂表达式.....                     | (142) |
| 运算符的独立性.....                   | (143) |
| 位.....                         | (144) |
| if 条件语句.....                   | (146) |
| 运算符的运算次序.....                  | (146) |
| 运算符实例.....                     | (147) |
| <br>                           |       |
| <b>第十八章 C++ 的流程</b> .....      | (149) |
| if、for 和 while 三大关键词 .....     | (150) |
| if 语句 .....                    | (150) |
| for 语句.....                    | (153) |
| While 关键字 .....                | (154) |
| 求数的阶乘.....                     | (155) |
| 其他流程控制关键词.....                 | (156) |
| switch、case 和 break 关键词.....   | (156) |
| do 关键词 .....                   | (157) |
| goto 关键词 .....                 | (158) |
| <br>                           |       |
| <b>第十九章 更加完善的自动电唱机应用</b> ..... | (159) |
| 基本原理.....                      | (159) |
| 自动电唱机程序清单.....                 | (161) |
| <br>                           |       |
| <b>第二十章 函数</b> .....           | (163) |
| 开放式语句.....                     | (163) |
| 函数的定义和调用.....                  | (164) |
| 形参.....                        | (165) |
| 函数返回值.....                     | (166) |
| Factorial 程序 .....             | (168) |
| 阅读含有函数的程序.....                 | (170) |
| 变量和名称作用域.....                  | (170) |
| 内联扩展.....                      | (170) |

|                           |              |
|---------------------------|--------------|
| 信息的存储                     | (170)        |
| 在 RMA 中存储信息               | (170)        |
| 在 CPU 寄存器中存储信息            | (170)        |
| 在指令高速缓存器中存储信息             | (171)        |
| 信息存储方式与函数调用的关系            | (171)        |
| 内联扩展                      | (171)        |
| 递归函数                      | (172)        |
| 省略号在函数中的使用                | (174)        |
| 缺省初始值                     | (174)        |
| <b>第二十一章 指针</b>           | <b>(175)</b> |
| 指针的含义                     | (176)        |
| 指针的用途                     | (177)        |
| 值和地址                      | (177)        |
| 未命名的内存                    | (177)        |
| 使用指针的具体例子                 | (178)        |
| C++ 中如何使用指针               | (180)        |
| 程序的地址簿                    | (180)        |
| 复引用指针                     | (181)        |
| 复引用指针实例                   | (181)        |
| 程序清单                      | (181)        |
| 改变指针所指的值                  | (182)        |
| 改变结构中的值                   | (182)        |
| 指针的简明表示法                  | (183)        |
| new 的用法                   | (183)        |
| 不要丢失内存中的地址                | (184)        |
| 链表实例                      | (184)        |
| 工作原理                      | (185)        |
| 程序清单                      | (186)        |
| 释放内存                      | (188)        |
| 删除后的指针                    | (188)        |
| GP 错误                     | (188)        |
| 字符串与指针                    | (189)        |
| 避免使用 void *               | (191)        |
| 关于指针的提示                   | (192)        |
| 指针小结                      | (192)        |
| <b>第二十二章 用指针编写自动电唱机应用</b> | <b>(193)</b> |
| 如何加入指针                    | (193)        |

|                                 |       |
|---------------------------------|-------|
| 改进后的新代码·····                    | (195) |
| <b>第二十三章 数组</b> ·····           | (197) |
| 如何使用枚举·····                     | (197) |
| 使用枚举时的安全性·····                  | (198) |
| cin 与枚举·····                    | (198) |
| 数组·····                         | (199) |
| 数组元素·····                       | (200) |
| 数组初始化·····                      | (200) |
| 字符串数组·····                      | (201) |
| 多维数组·····                       | (202) |
| 声明多维数组·····                     | (202) |
| 关于多维数组的使用方法·····                | (203) |
| <b>第二十四章 用数组编写自动电唱机应用</b> ····· | (205) |
| 使用数组后的变化·····                   | (205) |
| 程序中的其他变化·····                   | (206) |
| 代码·····                         | (207) |
| <b>第二十五章 作用域和变量</b> ·····       | (209) |
| 作用域的安全性·····                    | (210) |
| 作用域规则·····                      | (211) |
| 作用域的使用·····                     | (212) |
| 堆栈·····                         | (213) |
| <b>第三部分 完全地面向对象</b> ·····       | (215) |
| <b>第二十六章 类的剖析</b> ·····         | (217) |
| 欢迎来到 101 类空间·····               | (218) |
| 数据成员·····                       | (218) |
| 成员函数·····                       | (218) |
| 类的声明·····                       | (219) |
| 访问限制·····                       | (219) |
| 受保护的访问·····                     | (220) |
| 如何生成只读变量·····                   | (220) |
| 成员函数的定义·····                    | (220) |
| 类的使用·····                       | (221) |
| 类成员的访问·····                     | (221) |
| 从成员函数访问成员·····                  | (222) |
| 类的头部·····                       | (222) |

|   |              |
|---|--------------|
| 以面向对象方式思考 Music .....                         | (223)        |
| 类的总结 .....                                    | (224)        |
| <b>第二十七章 构造函数与析构函数 .....</b>                  | <b>(227)</b> |
| 构造函数 .....                                    | (227)        |
| 多构造函数 .....                                   | (228)        |
| 将构造函数置于此处 .....                               | (230)        |
| 构造函数的公共性与私有性 .....                            | (230)        |
| 析构函数 .....                                    | (230)        |
| 清理自己的垃圾 .....                                 | (231)        |
| 还要记得清理动态垃圾 .....                              | (231)        |
| 如果类中有类会怎样呢 .....                              | (232)        |
| 如何阅读面向对象程序 .....                              | (233)        |
| <b>第二十八章 Jukebox ++ .....</b>                 | <b>(235)</b> |
| Jukebox 中的类 .....                             | (235)        |
| 其余部分进展如何呢 .....                               | (238)        |
| 链表是如何工作的 .....                                | (239)        |
| 总体流程 .....                                    | (240)        |
| 这就是 Jukebox! 这就是面向对象! 这就是 Jukebox ++ 代码 ..... | (241)        |
| <b>第二十九章 继承 .....</b>                         | <b>(243)</b> |
| 类的重用是最经济的 .....                               | (244)        |
| 继承的方法 .....                                   | (244)        |
| Public、Private 和 Protected 的影响机制 .....        | (245)        |
| 重载 .....                                      | (245)        |
| 从父代最大限度地获取 .....                              | (245)        |
| 没有人会怀疑继承的作用 .....                             | (246)        |
| 这一点很重要:如何调用基类中的专用构造函数 .....                   | (248)        |
| 调用构造函数与析构函数的次序 .....                          | (248)        |
| 这一点也很重要:指针与派生类 .....                          | (249)        |
| 访问权的继承 .....                                  | (249)        |
| <b>第三十章 虚函数与多态性 .....</b>                     | <b>(251)</b> |
| 如何决定是否需要使用虚函数 .....                           | (252)        |
| 虚函数的声明 .....                                  | (253)        |
| 在 C++ 中虚函数的使用是必要的 .....                       | (254)        |
| 继承的确认 .....                                   | (257)        |



|                          |       |
|--------------------------|-------|
| <b>第三十一章 模板</b> .....    | (259) |
| 加入整洁的模板俱乐部.....          | (259) |
| 模板方面的工作还有许多.....         | (261) |
| 为模板定义成员函数.....           | (261) |
| 使用模板化的类.....             | (262) |
| 模板化类的指针的使用规则 .....       | (262) |
| 消除某些混乱 .....             | (263) |
| 放入头文件 .....              | (263) |
| 模板化的链表程序.....            | (264) |
| 代码.....                  | (266) |
| <br>                     |       |
| <b>第三十二章 音乐声音</b> .....  | (269) |
| 如何阅读多文件程序.....           | (269) |
| 该程序到底在做什么.....           | (270) |
| FancyJukebox 类.....      | (270) |
| PreviewSongList 类.....   | (272) |
| 链表中对象的混合.....            | (273) |
| 存储头文件中的类声明 .....         | (275) |
| 范例程序.....                | (275) |
| <br>                     |       |
| <b>第三十三章 I/O 流</b> ..... | (277) |
| 瞬时文件访问.....              | (277) |
| 还需要了解的内容.....            | (278) |
| 有关文件的五点说明.....           | (278) |
| 读取数字与单词的快速实例.....        | (279) |
| 代码.....                  | (280) |
| 闯入流中的特殊部分.....           | (282) |
| 为整型设置填充和宽度.....          | (282) |
| 让你的小舟随着 I/O 流慢慢行走 .....  | (283) |
| <br>                     |       |
| <b>第三十四章 异常处理</b> .....  | (285) |
| 旧式的错误处理.....             | (285) |
| 新的、改进的错误处理 .....         | (286) |
| 异常处理的“深加工”.....          | (287) |
| 示例 .....                 | (287) |
| 灵活性:异常处理的中间名.....        | (289) |
| 异常的抛出是必要的.....           | (290) |
| 只不过是语法.....              | (291) |
| 易引起混乱的部分.....            | (293) |