



北京市高等教育精品教材立项项目



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

Visual

# ++ 面向对象编程教程

王育坚 编著



清华大学出版社

# Visual C++ 面向对象编程教程

王育坚 编著

清华大学出版社  
北京

## 内 容 简 介

本书是按照北京市高等教育精品教材立项项目的总体要求而编写的。

本书系统地介绍了 Visual C++ 面向对象编程的基础知识和基本方法。全书分为 9 章, 主要内容包括 C++ 程序设计、Visual C++ 编程基础和 Visual C++ 高级编程三个部分, 按照 Visual C++ 编程环境→C++ 语言→应用程序向导→文档/视图结构→MFC→对话框→图形处理→编程深入的顺序安排学习内容。为了配合教学, 本书还提供了非常丰富的习题。

全书内容安排循序渐进, 重点突出, 实例典型, 文字精练。本书适合作为高等学校相关课程的教材或参考书, 也可作为 Visual C++ 初、中级用户的自学读本或培训教材。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

### 图书在版编目(CIP)数据

Visual C++ 面向对象编程教程/王育坚编著. —北京: 清华大学出版社, 2003

(北京市高等教育精品教材立项项目)

ISBN 7-302-06914-X

I. V… II. 王… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 059050 号

出 版 者: 清华大学出版社

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

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

责任编辑: 张瑞庆

封面设计: 孟繁聪

版式设计: 肖 米

印 刷 者: 北京牛山世兴印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×230 印张: 27.5 字数: 561 千字

版 次: 2003 年 9 月第 1 版 2003 年 9 月第 1 次印刷

书 号: ISBN 7-302-06914-X/TP·5118

印 数: 1~6000

定 价: 35.00 元



# 序

我一直对程序库/框架 (libraries/frameworks) 之类的产品和技术有着深厚的兴趣。在我的技术探索地图中, 这两者一直是最大板块。MFC (Microsoft Foundation Classes), 这个曾经最重要、最为全世界广泛使用的最早先趋框架产品之一, 一直是我关注的焦点。由于 MFC 附含于 Microsoft Visual C++ 之中, 导致我们常把学习 MFC 和学习 Visual C++ 并谈。名称其实无所谓, 学习 Visual C++ 主要就是学习如何运用 MFC。Visual C++ 本身只是个编译器, 并不需要太多学习 (当然它丰富的集成环境需要多加熟练, 但这种学习很简单)。这里头真正要学习的, 是 C++ 语言, 以及以 C++ 语言写就的庞大复杂的应用框架 MFC。

所谓框架 (framework), 是一种以 classes 集体力量为用户完成工作, 并允许用户在特定协议下注入新血完成扩张的大型程序库 (library)。个别 class 有自己明确的目标, 集体行动的 classes 当然也必须有一致的明确目标, 因此各式各样的框架有聚焦于数据结构/算法者 (如 STL), 有聚焦于网络通信者 (如 ACE), 有聚焦于设计模式者 (如 Loki)。其中聚焦于应用程序骨干 (通常包含 GUI 图形界面、文档视图、打印、预视、数据交换能力……) 者, 技术位阶极高, 我们特别称为应用框架 (application framework)。MFC 就是一种“应用框架”, 帮助你塑造、建构出你的 Windows 应用程序骨干。

除非我等为框架开发者或研究者, 否则学习框架是为了应用, 不是为了其中展现的技术, 更不是为了摸清楚其中每一个环节和每一个流动。是的, 除非为了学术目的与纯粹技术钻研, 否则学习的目的是为了应用。但是正因为我们对事物的运用娴熟度随着我们对事物原理的更多理解而增强, 又因为 MFC 如此庞大繁复 (框架无不如此), 管中窥豹、盲人摸象的情况时有所闻, 失控失准的现实例子屡见不鲜, 所

以我们才需要《Dissecting MFC》、《Internal MFC》之类的书籍，由它们以各自的定位为读者进行一场外科手术，剖析 MFC 的肌理与神经。

理解 MFC 框架本身的同时，为了流畅地运用它，我们还需要大量的实例和良好的解说，并且最好能够与 MFC 的肌理和神经关联起来。这方面，在 MFC 问世多年的今天，也有了很好好书，例如《Inside Visual C++》和《Programming Windows with MFC》。

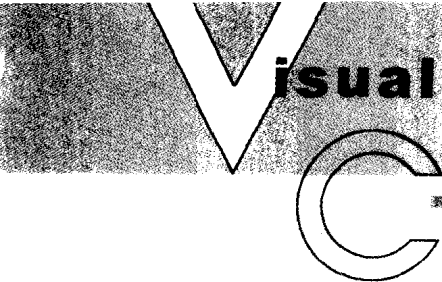
王育坚老师，以十分认真严谨的态度，完成了这本《Visual C++ 面向对象编程教程》。我结识王育坚老师于网络虚拟空间，迄今未曾见面，却很荣幸有机会在本书出版之前阅读定稿。从这些稿件中，我体会到本书在选题、内容上的优越，以及作者的认真、用心与自信。任何技术都有明日黄花的一天，MFC 不可能例外，但是从一本优秀书籍中所学到的技术、知识乃至态度，是可以延续的。

中国的信息产业非常需要坚实并富实际作用的教材。这需要一批对技术、对教育、对写作都有足够实力与热情的人参与，也需要出版大环境提供更优渥的吸引力。技术书籍谈的是技术，技术书籍的写作本身则就是一门极不容易的技术。基于我对本书稿件的印象以及对王育坚老师的认识，我很乐意向读者推荐这本书。

侯 捷

中国台湾·新竹

2003.07.01



# 前言

当本书准备交付出版社之际，编写过程中的烦躁和劳累全被拂去。我以前曾写过书，深知写书过程中的痛苦。因此，当着手写这本书时，面对 Visual C++ 类书籍汗牛充栋的局面，真有些惶惶不安。甚至在正式动笔撰稿时，还有过放弃的念头，但我最终还是坚持下来了。近两年的辛勤耕耘，最终瓜熟蒂落。就像中国台湾 Windows 编程著作家和专栏评论家侯捷先生在他的《深入浅出 MFC》一书中所说，“学习过往的艰险，模糊而明亮，是学成冠冕上闪亮的宝石。过程愈艰辛，宝石愈璀璨”。艰辛永远是和幸福愉悦相伴随的。

本书是按照北京市高等教育精品教材立项项目的要求而编写，目的是为相关课程的教学提供一本精品教材。毋庸置疑，高等教育改革的一个重要方向就是使教学真正面向社会。作者的深刻感受就是当前大学计算机课程的教学落后于计算机技术的飞速发展，知识的讲授大都停留在理论基础上，培养的人才无法很快地适应实际的软件开发工作。教学实践证明，Visual C++ 作为应用非常广泛的 Windows 程序开发工具，在高校相关专业开设其编程课程是非常必要且有效的。

市面上介绍 Visual C++ 的书籍很多，但要找到真正适合大学专业课教学的教材却不易。本书的特色是：读者如果真正读懂了本书，就能够成为一名合格的 Visual C++ 程序员，因为本书浓缩了作者多年来软件开发和教学实践的经验和体会，通过多次讲授 Visual C++ 编程，作者能够深刻理解 Visual C++ 编程的基本学习要求。本书主要面向 Visual C++ 的初、中级用户，读者阅读本书前最好具有 C 语言基础。

比较 Visual Basic、C++ Builder 和 Delphi 等编程工具，用 Visual C++ 编写 Windows 应用程序最富于挑战性和艰巨性。有些人学习了很长时间仍在 Visual C++ 的门口徘徊。究其根源，是由于他们采取一种

蜻蜓点水的学习方式，而没有系统地学习。要想在使用 Visual C++ 编程时游刃有余，在学习过程中必须解决所面临的四个主要困难：C++ 语言的面向对象机制、MFC 类库功能的庞大、Windows 编程的复杂性以及向导所建立的应用程序框架的透明性。

Visual C++ 编程的内容广泛并相互交织，如何编排见仁见智。本书在编写过程中充分考虑了 Visual C++ 的学习特点和教学过程中的基本思路，将重点放在 Visual C++ 基础和实用的知识点上，力求让读者按照循序渐进的过程进行学习，提高学习的效率。

为了突出教学重点，对书中用到的实例进行了一些简化，这样既节约了篇幅，又避免了喧宾夺主。区别于一般的技术读物，本书提供的习题非常丰富，并且都是围绕书中例题而展开的，使读者能够马上学有所用。

如果将本书作为教材使用，建议将课程的教学分为课堂讲授、教师指导上机和学生自己上机三个层次。课堂讲授和教师指导上机的教学课时为 72~90 学时，教师可根据不同教学对象或教学大纲的要求安排学时数和教学内容。上机编程是本课程的重要环节，建议教师指导学生上机的课时为每周 2 学时。此外，学生必须花费大量时间自己上机，学生课余上机时间应不少于每周 3 学时。每章后面都提供了专门用于上机的习题，教师可根据具体情况酌情选择。

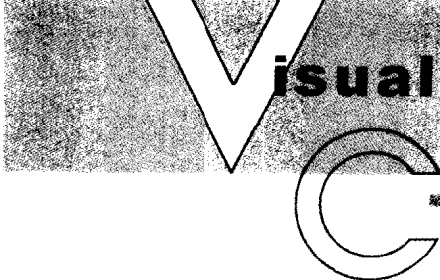
本书凝聚了很多同仁的劳动和智慧，李启隆、宋一中、陈贻昆和王琦等老师参加了部分章节的编写。中国台湾的侯捷先生仔细阅读了本书，并义务为本书作序。在此，对他们表示衷心的感谢！

感谢阅读本书的读者！感谢将本书作为教材的老师！恳请读者惠予批评指正。

王育坚

vcplus2@yahoo.com.cn

2003 年 7 月



# 目 录

|                                |    |
|--------------------------------|----|
| <b>第 1 章 Visual C++ 集成开发环境</b> | 1  |
| 1.1 Visual C++ 概述              | 1  |
| 1.1.1 Visual C++ 的特点           | 1  |
| 1.1.2 Visual C++ 的安装           | 2  |
| 1.1.3 集成开发环境窗口                 | 3  |
| 1.1.4 MSDN 帮助系统                | 5  |
| 1.2 项目和项目工作区                   | 7  |
| 1.2.1 项目                       | 7  |
| 1.2.2 项目工作区                    | 7  |
| 1.2.3 项目设置                     | 10 |
| 1.3 编辑器                        | 11 |
| 1.3.1 源代码编辑器                   | 11 |
| 1.3.2 资源编辑器                    | 12 |
| 1.4 菜单栏和工具栏                    | 17 |
| 1.4.1 菜单栏                      | 17 |
| 1.4.2 工具栏                      | 18 |
| 1.4.3 Standard 工具栏             | 19 |
| 1.4.4 Build MiniBar 工具栏        | 20 |
| 1.4.5 WizardBar 工具栏            | 21 |
| 习题                             | 22 |
| <b>第 2 章 C++ 语言基础</b>          | 23 |
| 2.1 C++ 概述                     | 23 |
| 2.1.1 C++ 的主要特点                | 23 |
| 2.1.2 C++ 程序与 C 程序             | 24 |

|       |                               |    |
|-------|-------------------------------|----|
| 2.1.3 | C++ 对 C 的一般扩充 .....           | 25 |
| 2.1.4 | 利用 Visual C++ 编制 C++ 程序 ..... | 26 |
| 2.2   | C++ 程序基本要素 .....              | 27 |
| 2.2.1 | 标识符 .....                     | 27 |
| 2.2.2 | 关键字 .....                     | 27 |
| 2.2.3 | 常量和变量 .....                   | 28 |
| 2.2.4 | 运算符和表达式 .....                 | 28 |
| 2.3   | C++ 数据类型 .....                | 29 |
| 2.3.1 | 基本数据类型 .....                  | 29 |
| 2.3.2 | const 常量类型 .....              | 31 |
| 2.3.3 | 数组 .....                      | 32 |
| 2.3.4 | 指针 .....                      | 33 |
| 2.3.5 | 字符串 .....                     | 35 |
| 2.3.6 | 结构 .....                      | 36 |
| 2.3.7 | 枚举 .....                      | 38 |
| 2.3.8 | 用 typedef 定义类型 .....          | 38 |
| 2.4   | 变量的作用域与存储类型 .....             | 39 |
| 2.4.1 | 作用域和生存期 .....                 | 39 |
| 2.4.2 | 存储类型 .....                    | 41 |
| 2.4.3 | 作用域限定符 .....                  | 43 |
| 2.4.4 | 动态内存分配 .....                  | 43 |
| 2.5   | 控制语句 .....                    | 45 |
| 2.5.1 | 条件语句 .....                    | 45 |
| 2.5.2 | 循环语句 .....                    | 47 |
| 2.5.3 | 转移语句 .....                    | 49 |
| 2.6   | 函数 .....                      | 50 |
| 2.6.1 | 函数定义、函数调用和函数声明 .....          | 50 |
| 2.6.2 | 函数参数的默认值 .....                | 54 |
| 2.6.3 | 内联函数 .....                    | 55 |
| 2.6.4 | 引用 .....                      | 55 |
| 2.7   | 编译预处理指令 .....                 | 57 |
| 2.7.1 | #include 文件包含指令 .....         | 57 |
| 2.7.2 | #define 宏定义指令 .....           | 57 |
| 2.7.3 | 条件编译指令 .....                  | 59 |

|                                      |           |
|--------------------------------------|-----------|
| 习题 .....                             | 60        |
| <b>第3章 C++ 面向对象程序设计 .....</b>        | <b>67</b> |
| 3.1 面向对象程序设计概述 .....                 | 67        |
| 3.1.1 传统的结构化程序设计 .....               | 67        |
| 3.1.2 面向对象程序设计方法及特征 .....            | 68        |
| 3.2 C++ 类 .....                      | 70        |
| 3.2.1 类的定义与实现 .....                  | 70        |
| 3.2.2 构造函数和析构函数 .....                | 72        |
| 3.2.3 this 指针 .....                  | 74        |
| 3.2.4 静态成员 .....                     | 76        |
| 3.2.5 友元 .....                       | 78        |
| 3.3 类的继承 .....                       | 80        |
| 3.3.1 基类和派生类 .....                   | 80        |
| 3.3.2 基类和派生类的构造函数 .....              | 83        |
| 3.3.3 多重继承 .....                     | 85        |
| 3.3.4 虚基类 .....                      | 87        |
| 3.4 多态性和虚函数 .....                    | 89        |
| 3.4.1 用基类指针指向派生类对象 .....             | 89        |
| 3.4.2 虚函数 .....                      | 90        |
| 3.4.3 抽象类和纯虚函数 .....                 | 93        |
| 3.5 重载 .....                         | 94        |
| 3.5.1 函数重载 .....                     | 95        |
| 3.5.2 运算符重载 .....                    | 96        |
| 3.6 C++ 模板 .....                     | 99        |
| 3.6.1 函数模板 .....                     | 99        |
| 3.6.2 类模板 .....                      | 100       |
| 3.7 Microsoft Visual C++ 的语法扩充 ..... | 102       |
| 3.7.1 Visual C++ 自定义数据类型 .....       | 102       |
| 3.7.2 Visual C++ 运行库 .....           | 103       |
| 3.7.3 运行时类型识别 .....                  | 104       |
| 3.7.4 编程规范 .....                     | 106       |
| 习题 .....                             | 107       |

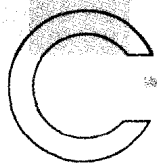
|                         |     |
|-------------------------|-----|
| <b>第 4 章 创建应用程序框架</b>   | 117 |
| 4.1 应用程序向导              | 117 |
| 4.1.1 Visual C++ 中的向导类型 | 117 |
| 4.1.2 使用 MFC AppWizard  | 118 |
| 4.1.3 MFC 应用程序的开发流程     | 128 |
| 4.2 应用程序向导生成的文件         | 130 |
| 4.2.1 应用程序向导生成的文件类型     | 130 |
| 4.2.2 应用程序向导生成的头文件      | 131 |
| 4.2.3 应用程序向导生成的实现文件     | 133 |
| 4.2.4 应用程序向导生成的资源文件     | 139 |
| 4.2.5 应用程序向导生成的其他文件     | 140 |
| 4.3 ClassWizard 类向导     | 141 |
| 4.3.1 ClassWizard 的功能   | 141 |
| 4.3.2 添加消息处理函数          | 142 |
| 4.3.3 为项目添加新类           | 145 |
| 4.4 程序调试                | 146 |
| 4.4.1 查找源程序中的语法错误       | 146 |
| 4.4.2 Debug 调试器         | 147 |
| 4.4.3 跟踪调试可执行程序         | 150 |
| 习题                      | 152 |
| <b>第 5 章 文档与视图</b>      | 155 |
| 5.1 文档与视图结构             | 155 |
| 5.1.1 文档与视图结构概述         | 155 |
| 5.1.2 文档与视图之间的相互作用      | 157 |
| 5.1.3 多文档               | 159 |
| 5.2 菜单设计                | 162 |
| 5.2.1 建立菜单资源            | 162 |
| 5.2.2 添加菜单命令处理函数        | 164 |
| 5.2.3 弹出式菜单             | 168 |
| 5.3 鼠标消息处理              | 170 |
| 5.3.1 鼠标消息              | 170 |
| 5.3.2 一个简单的绘图程序         | 170 |
| 5.4 工具栏和状态栏设计           | 176 |

|              |                  |            |
|--------------|------------------|------------|
| 5.4.1        | 添加工具栏按钮          | 176        |
| 5.4.2        | 定制状态栏            | 177        |
| 5.5          | 文档的读写            | 180        |
| 5.5.1        | 序列化工作原理          | 180        |
| 5.5.2        | MFC 应用程序的序列化     | 182        |
| 5.6          | 使用不同视图           | 184        |
| 5.6.1        | 滚动视图             | 185        |
| 5.6.2        | 多视图              | 186        |
|              | 习题               | 190        |
| <b>第 6 章</b> | <b>MFC 原理与方法</b> | <b>193</b> |
| 6.1          | Windows 编程机制     | 193        |
| 6.1.1        | 关于 API 和 SDK     | 194        |
| 6.1.2        | 句柄               | 198        |
| 6.1.3        | 事件和消息            | 200        |
| 6.2          | MFC 微软基础类        | 202        |
| 6.2.1        | MFC 概述           | 202        |
| 6.2.2        | MFC 体系结构         | 204        |
| 6.2.3        | 学习 MFC 的方法       | 206        |
| 6.3          | MFC 应用程序框架       | 207        |
| 6.3.1        | 应用程序框架中的对象       | 207        |
| 6.3.2        | MFC 应用程序的生与死     | 208        |
| 6.3.3        | 常用的 MFC 文件和库文件   | 211        |
| 6.4          | MFC 消息管理         | 214        |
| 6.4.1        | MFC 消息映射机制       | 214        |
| 6.4.2        | 消息的发送            | 216        |
| 6.4.3        | 手工处理消息映射         | 217        |
| 6.5          | MFC 宏            | 220        |
| 6.5.1        | 有关运行时类型识别的宏      | 221        |
| 6.5.2        | MFC 调试宏          | 223        |
| 6.6          | 常用的 MFC 类        | 224        |
| 6.6.1        | CRuntimeClass 结构 | 225        |
| 6.6.2        | CObject 类        | 225        |
| 6.6.3        | CCmdTarget 类     | 227        |

|              |                           |            |
|--------------|---------------------------|------------|
| 6.6.4        | CWinApp 类 .....           | 228        |
| 6.6.5        | CWnd 类 .....              | 228        |
| 6.6.6        | CFrameWnd 类 .....         | 229        |
| 6.6.7        | CDocument 和 CView 类 ..... | 229        |
| 习题           | .....                     | 230        |
| <b>第 7 章</b> | <b>对话框和控件 .....</b>       | <b>233</b> |
| 7.1          | 对话框概述 .....               | 233        |
| 7.1.1        | 基于对话框的应用程序 .....          | 233        |
| 7.1.2        | 对话框类 CDialog .....        | 235        |
| 7.1.3        | 对话框数据交换 DDX 和验证 DDV ..... | 236        |
| 7.1.4        | 提示信息对话框 .....             | 239        |
| 7.2          | 使用对话框 .....               | 240        |
| 7.2.1        | 一般对话框工作流程 .....           | 241        |
| 7.2.2        | 创建对话框 .....               | 242        |
| 7.2.3        | 添加控件及关联的成员变量 .....        | 245        |
| 7.2.4        | 非模态对话框 .....              | 251        |
| 7.2.5        | 公用对话框 .....               | 256        |
| 7.3          | 标准控件 .....                | 258        |
| 7.3.1        | 控件概述 .....                | 258        |
| 7.3.2        | 控件的组织 .....               | 260        |
| 7.3.3        | 控件共有属性 .....              | 262        |
| 7.3.4        | 静态控件 .....                | 264        |
| 7.3.5        | 编辑框 .....                 | 265        |
| 7.3.6        | 按钮 .....                  | 268        |
| 7.3.7        | 列表框 .....                 | 272        |
| 7.3.8        | 组合框 .....                 | 277        |
| 7.3.9        | 滚动条 .....                 | 282        |
| 7.4          | 公共控件 .....                | 283        |
| 7.4.1        | 旋转按钮 .....                | 283        |
| 7.4.2        | 滑块 .....                  | 286        |
| 7.4.3        | 进度条 .....                 | 289        |
| 7.4.4        | 标签控件和属性表 .....            | 291        |
| 7.4.5        | 图像列表 .....                | 298        |

|                             |            |
|-----------------------------|------------|
| 7.4.6 列表视控件                 | 301        |
| 7.4.7 树视控件                  | 308        |
| 习题                          | 312        |
| <b>第8章 图形处理</b>             | <b>317</b> |
| 8.1 图形设备接口                  | 317        |
| 8.1.1 概述                    | 317        |
| 8.1.2 设备环境                  | 318        |
| 8.1.3 GDI 坐标系和映射模式          | 320        |
| 8.1.4 颜色的设置                 | 326        |
| 8.2 画笔和画刷                   | 327        |
| 8.2.1 GDI 对象                | 327        |
| 8.2.2 使用画笔                  | 328        |
| 8.2.3 使用画刷                  | 331        |
| 8.2.4 使用 GDI 堆对象            | 333        |
| 8.2.5 绘制基本图形                | 335        |
| 8.3 文本与字体                   | 337        |
| 8.3.1 绘制文本                  | 337        |
| 8.3.2 字体概述                  | 338        |
| 8.3.3 创建字体                  | 340        |
| 8.4 位图、图标和光标                | 345        |
| 8.4.1 位图                    | 345        |
| 8.4.2 图标                    | 347        |
| 8.4.3 光标                    | 349        |
| 习题                          | 352        |
| <b>第9章 深入 Visual C++ 编程</b> | <b>355</b> |
| 9.1 打印                      | 355        |
| 9.1.1 MFC 的打印流程             | 355        |
| 9.1.2 设置打印坐标系               | 358        |
| 9.1.3 分页处理                  | 361        |
| 9.1.4 停止和取消打印               | 365        |
| 9.2 异常处理                    | 366        |
| 9.2.1 C++ 中的异常处理            | 366        |

|       |                         |     |
|-------|-------------------------|-----|
| 9.2.2 | Win32 异常处理 .....        | 368 |
| 9.2.3 | MFC 异常宏 .....           | 372 |
| 9.2.4 | MFC 中的异常处理 .....        | 373 |
| 9.3   | 动态链接库 .....             | 377 |
| 9.3.1 | 动态链接库概述 .....           | 377 |
| 9.3.2 | 创建动态链接库 .....           | 378 |
| 9.3.3 | 使用动态链接库 .....           | 380 |
| 9.4   | 多线程 .....               | 384 |
| 9.4.1 | 进程和线程 .....             | 384 |
| 9.4.2 | 工作线程 .....              | 385 |
| 9.4.3 | 用户界面线程 .....            | 390 |
| 9.5   | ActiveX 控件 .....        | 392 |
| 9.5.1 | 组件和 ActiveX 控件 .....    | 392 |
| 9.5.2 | 创建 ActiveX 控件 .....     | 394 |
| 9.5.3 | ActiveX 控件的使用 .....     | 399 |
| 9.6   | 托盘程序 .....              | 400 |
| 9.6.1 | 利用钩子截获消息 .....          | 400 |
| 9.6.2 | 实现托盘的数据结构和 API 函数 ..... | 402 |
| 9.6.3 | 一个截获键盘输入的托盘程序 .....     | 403 |
| 9.7   | ODBC 数据库编程 .....        | 408 |
| 9.7.1 | ODBC 概述 .....           | 408 |
| 9.7.2 | MFC ODBC 数据库类 .....     | 409 |
| 9.7.3 | 实现步骤 .....              | 410 |
|       | 习题 .....                | 417 |
|       | 常用术语索引 .....            | 419 |
|       | 参考文献 .....              | 422 |
|       | 读者意见反馈 .....            | 423 |



## Visual C++ 集成开发环境

Visual C++ 是一个功能强大的可视化软件开发工具。自 1993 年 Microsoft 公司推出 Visual C++ 1.0 后,随着其新版本的不断问世,Visual C++ 已成为专业程序员进行软件开发的首选工具。要使用 Visual C++ 编程,首先要了解它的集成开发环境。Visual C++ 集成开发环境中的程序开发工具很多,要真正掌握它们必须结合具体的编程内容。本章只简单介绍 Visual C++ 集成开发环境的特点、安装要求、界面风格和简单工具的使用。通过本章的学习,读者能够对 Visual C++ 有一个感性的认识。

虽然微软公司新近推出了 Visual C++ .NET(即 Visual C++ 7.0),但它的应用有很大的局限性,只适用于 Windows 2000、Windows XP 和 Windows NT 4.0。考虑教材特点和实际应用情况,本书仍然以 Visual C++ 6.0 作为教学平台。

### 1.1 Visual C++ 概述

Visual C++ 6.0 不仅是一个 C++ 编译器,而且是一个基于 Windows 操作系统的可视化集成开发环境(integrated development environment, IDE)。Visual C++ 6.0 由许多组件组成,包括编辑器、编译器、调试器以及程序向导 AppWizard、类向导 ClassWizard 等开发工具。这些组件通过一个名为 Developer Studio 的组件集成为和谐的开发环境。

#### 1.1.1 Visual C++ 的特点

Visual C++ 的源程序要求用 C++ 语言编写,它支持面向对象设计方法,并可以使用功能强大的微软基础类库 MFC(Microsoft foundation class),充分体现了 Microsoft 公司的技术精华。由于 Microsoft 公司在操作系统市场上的垄断地位,用 Visual C++ 开发出来的软件稳定性好、可移植性强,而且软件与硬件相互独立。

利用 Visual C++ 可以编制各种类型的 Windows 应用程序,从最简单的单文档和对

话框程序到复杂的多文档和组合界面程序。并且 Visual C++ 作为 Visual Studio 可视化组件家族中最重要的一员,它与其他可视化开发工具如 Visual Basic、Visual J++ 及 Visual C# 紧密地集成在一起,可进行不同类型的程序开发工作,适用于特殊、复杂和综合软件项目的开发以及系统软件的设计。

Visual C++ 6.0 源代码编辑器提供了自动语句完成功能,编辑输入源程序时,它能自动显示当前对象的成员变量和成员函数,并指明函数的参数类型。Visual C++ 6.0 的编译器增加了新的编译参数,改进了对 ANSI C++ 标准的支持,并采用 Microsoft 的代码优化技术,使生成的目标代码更短小,应用程序运行的速度更快。Visual C++ 6.0 程序调试器功能更强大,它提供了诊断映射机制、无须重编译的调试、远程调试和实时调试等功能。

Visual C++ 6.0 的联机帮助系统 MSDN Library (Microsoft developer network library) 采用当今流行的 HTML 格式。它既能与集成开发环境有机地结合在一起,使得用户在编程时随机查询需要的内容,又能脱离集成开发环境而独立地运行。并且用户还可以通过因特网获取实时的帮助信息和实例。

Visual C++ 6.0 通过 Visual Studio 为用户提供了其他一些实用工具,如 Spy++ 查看器、ActiveX Control Test Container 控件测试容器及 Register Control 控件注册程序,扩展了 Visual C++ 的功能。

除了 Visual C++, 其他一些知名公司也推出了自己的 C++ 集成开发环境,如 Borland 的 C++ Builder 和 IBM 的 VisualAge C++。迄今为止,Visual C++ 可以说是一个功能最强大和使用范围最广泛的软件开发工具。当然,由于 Windows 编程的困难和 MFC 类库功能的庞大,加上应用程序向导生成的程序框架结构的复杂性,使得学习 Visual C++ 比学习其他开发工具更困难。用 Visual C++ 编写 Windows 应用程序最具有挑战性,当熟练掌握 Visual C++ 后,就会感受到作为 Visual C++ 程序员的无比优越性。

### 1.1.2 Visual C++ 的安装

计算机硬件技术发展很快,现在一般的机器配置就能支持 Visual C++ 的运行。以 Visual C++ 6.0 为例,要求 CPU 为 Pentium 以上系列,内存为 64MB,所需硬盘空间约为 500MB,操作系统为 Windows 95/98/2000 或 Windows NT。

安装 Visual C++ 6.0 时,首先将第一张光盘放入光驱中,如果计算机启用了自动安装 Autorun 特性,则几秒后安装程序自动运行,否则需要运行光盘根目录下的 Setup.exe 程序。安装向导程序启动后将自动完成 Visual C++ 的安装,用户只需根据对话框的提示输入具体的内容。如输入产品序列号和用户信息,选择安装选项 (Custom、Products、Server Applications)、安装路径和需要安装的组件等。

所选的具体安装选项和组件取决于 Visual Studio 的版本,如果产品是 Visual