



赠送光盘
中附有完
整的案例
源代码

Visual C++ 课程设计 案例精编

夏崇镠 任海军 余 健 编著

- GPA计算器 • 聊天程序 • 通讯录 •
- 餐厅管理系统 • 商品库存管理系统 •
- 旅馆管理系统 • 图书馆管理系统 •
- 本科生信息管理系统 • 出租车管理系统 •



清华大学出版社



高等院校课程设计方案精编

Visual C++课程设计方案精编

夏崇锴 任海军 余 健 编著

图书在版编目(CIP)数据

Visual C++课程设计方案精编/夏崇锴,任海军,余健编著. —北京:清华大学出版社,2008.11

ISBN 978-7-302-18612-0

Ⅰ. V… Ⅱ. 夏… Ⅲ. C… Ⅳ. TP312

1. Visual C++课程设计方案精编 Ⅱ. 夏崇锴,任海军,余健 Ⅲ. C++ Ⅳ. TP312

2. 计算机—教材 3. 程序设计—教材 4. 数据库—教材 5. 网络—教材 6. 图形—教材 7. 动画—教材 8. 多媒体—教材 9. 虚拟现实—教材 10. 人工智能—教材 11. 专家系统—教材 12. 模糊逻辑—教材 13. 神经网络—教材 14. 遗传算法—教材 15. 粒子群优化—教材 16. 蚁群算法—教材 17. 遗传编程—教材 18. 进化计算—教材 19. 模糊推理—教材 20. 专家推理—教材 21. 模糊控制—教材 22. 模糊决策—教材 23. 模糊聚类—教材 24. 模糊识别—教材 25. 模糊诊断—教材 26. 模糊预测—教材 27. 模糊控制—教材 28. 模糊决策—教材 29. 模糊聚类—教材 30. 模糊识别—教材 31. 模糊诊断—教材 32. 模糊预测—教材 33. 模糊控制—教材 34. 模糊决策—教材 35. 模糊聚类—教材 36. 模糊识别—教材 37. 模糊诊断—教材 38. 模糊预测—教材 39. 模糊控制—教材 40. 模糊决策—教材 41. 模糊聚类—教材 42. 模糊识别—教材 43. 模糊诊断—教材 44. 模糊预测—教材 45. 模糊控制—教材 46. 模糊决策—教材 47. 模糊聚类—教材 48. 模糊识别—教材 49. 模糊诊断—教材 50. 模糊预测—教材 51. 模糊控制—教材 52. 模糊决策—教材 53. 模糊聚类—教材 54. 模糊识别—教材 55. 模糊诊断—教材 56. 模糊预测—教材 57. 模糊控制—教材 58. 模糊决策—教材 59. 模糊聚类—教材 60. 模糊识别—教材 61. 模糊诊断—教材 62. 模糊预测—教材 63. 模糊控制—教材 64. 模糊决策—教材 65. 模糊聚类—教材 66. 模糊识别—教材 67. 模糊诊断—教材 68. 模糊预测—教材 69. 模糊控制—教材 70. 模糊决策—教材 71. 模糊聚类—教材 72. 模糊识别—教材 73. 模糊诊断—教材 74. 模糊预测—教材 75. 模糊控制—教材 76. 模糊决策—教材 77. 模糊聚类—教材 78. 模糊识别—教材 79. 模糊诊断—教材 80. 模糊预测—教材 81. 模糊控制—教材 82. 模糊决策—教材 83. 模糊聚类—教材 84. 模糊识别—教材 85. 模糊诊断—教材 86. 模糊预测—教材 87. 模糊控制—教材 88. 模糊决策—教材 89. 模糊聚类—教材 90. 模糊识别—教材 91. 模糊诊断—教材 92. 模糊预测—教材 93. 模糊控制—教材 94. 模糊决策—教材 95. 模糊聚类—教材 96. 模糊识别—教材 97. 模糊诊断—教材 98. 模糊预测—教材 99. 模糊控制—教材 100. 模糊决策—教材

清华大学出版社 地址:北京清华大学学研大厦A座 邮编:100084 电话:010-62770175 网址:www.tup.com.cn

1 0 0 0 8 4

北京100084信箱57

清华大学出版社

北京

邮政编码: □ □ □ □ □ □

内 容 简 介

Visual C++是 Microsoft 公司出品的运行于 Windows 平台上的交互式可视化集成开发工具。本书以案例的形式深入浅出地全面介绍了如何利用 Visual C++进行课程设计和软件制作。

本书共有 11 章。第 1、2 章介绍了 VC 编程需要掌握的知识和技巧。第 3~11 章介绍了课程设计案例,各个案例相对独立,覆盖了 MFC 编程的绝大部分,深入浅出地说明了 MFC 编程中最典型和最有用处的程序设计方法。考虑到数据库在实际课程设计中的重要性,本书有意增多这方面的典型案例,以使读者从中快速找到能解决当前问题的技术方法。本书的应用性极强,案例全部可以运行,读者可以根据这些案例进行研究、修改和扩展,使其符合自己的要求。

本书适合具有 C++基础知识的读者阅读,可作为课程设计的参考用书,也适合中高级用户进行开发设计时参考。

为方便读者使用,本书附带一张含有程序源代码的光盘。

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

Visual C++课程设计案例精编/夏崇锴,任海军,余健编著. —北京:清华大学出版社,2008.11
(高等院校课程设计案例精编)

ISBN 978-7-302-18612-0

I. V… II. ①夏… ②任… ③余… III. C 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 144788 号

责任编辑:闫光龙 宋延清

封面设计:山鹰工作室

版式设计:杨玉兰

责任印制:孟凡玉

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

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

印 刷 者:北京密云胶印厂

装 订 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185×260 印 张:23.75 字 数:571 千字

版 次:2008 年 11 月第 1 版 印 次:2008 年 11 月第 1 次印刷

附光盘 1 张

印 数:1~4000

定 价:42.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:028449-01

前言

提到在 Windows 下进行 32 位的应用程序开发,就不能不重点关注 Visual C++。相比其他的编程工具而言,Visual C++在提供了可视化编程方法的同时,也提供了编写系统底层操作的功能,其生成代码的质量是其他开发工具所无法比拟的。

然而,复杂的 MFC 结构往往给广大的在校学生、VC 初学者带来很大的困难,在使用 VC 工具进行应用系统开发时,更是困难重重。

本书通过生动有趣的、完整的案例开发过程,向读者介绍可视化编程的技术和软件开发的思维方式,并使读者能够领悟一些编程技巧。考虑到数据库在实际课程设计以及实际项目中应用的广泛性,本书有意增多这方面的典型案例,以使读者能够通过本书快速找到这方面的技术方法,应用到实际编程中。

本书共有 11 章,第 1、2 章介绍 VC 编程需要掌握的知识和技巧。第 3~11 章介绍课程设计案例,各个案例相对独立,覆盖了 MFC 编程的绝大部分。前面的案例便于迅速模仿实现,适合读者初学;后面的案例功能逐步丰富,并采用难度更大和功能更强的技术,让读者进一步掌握实际系统的设计与开发。各章具体内容如下。

- **第 1 章 Visual C++概述。**该章介绍 Visual C++的概况,以及如何使用 Visual C++提供的各种工具、有关工程创建的一般方法、调试技巧,及发布程序等必要的基础知识。限于篇幅,该章讲解时点到即止,读者应多练多想,才能完全掌握其内容,为以后应用程序的编制打下良好的基础。
- **第 2 章 MFC 基础。**一个程序员能不能成功地学会 VC 编程,在很大程度上依赖于他对 MFC 的理解和熟练运用程度。该章从应用的角度依次介绍 MFC 的层次、DDX、序列化、消息映射等 MFC 编程必须了解的知识,使读者能在短时间内对这些知识有个初步的印象,然后在后面的案例学习中进一步深化理解。
- **第 3 章 GPA 计算器。**GPA 计算器是所有案例中难度最小的一个应用程序,但俗话说“麻雀虽小,五脏俱全”,通过该程序可以学到对话框应用程序的编程方法、部分常用控件的使用规律、菜单的使用等极为普通但却很重要的内容。
- **第 4 章 聊天程序。**VC 的实际编程往往需要加入多种技术,如网络技术、数据库技术、串口技术等。上一章的 GPA 计算器属于纯对话框编程,而此章是在对话框编程的基础上加入了网络技术。通过本章的学习可以进一步深入理解对话框编程,同时也可以体会到 VC 编程的难点不仅仅在于复杂的 VC 框架,更在于这种编程需要融合多种技术,对这些技术我们需要一一地领会。
- **第 5 章 通讯录。**通讯录是文件信息系统类型程序的一个例子,该章为学生学习 VC++的单文档编程及其文档序列化操作提供了示范。作为第一个单文档应用程序,该章对项目建立的整个过程进行详细的跟踪指导,以便让学生掌握方法。
- **第 6 章 餐厅管理系统。**该章通过一个餐厅管理系统的完整开发过程,向读者介绍相对简单易用的 MFC ODBC 对数据库的访问技术。在界面设计方面使用选项卡分页管理。数据库采用 Microsoft Access 2003。

- **第7章** 商品库存管理系统。该章通过一个商品库存管理系统的完整开发过程，向读者介绍目前流行的 ADO 对象操作技术。在界面设计方面，由登录界面引导，主界面采用按钮分类。日志管理也是不少信息系统所必备的，这里也进行了详细介绍。数据库采用 Microsoft Access 2003。
- **第8章** 旅馆管理系统。该章通过一个旅馆管理系统的完整开发过程，向读者介绍目前流行的 ADO 对象操作技术。以单文档结构为设计基础，在界面设计方面，采用视图切分和树状内容分布以及快捷菜单操纵。数据库采用 Microsoft SQL Server 2000。
- **第9章** 图书馆管理系统。该章通过一个图书馆管理系统的完整开发过程，向读者介绍相对简单易用的 MFC ODBC 对数据库的访问技术。在界面设计方面使用按钮(提供操作标签提示功能)。数据库采用 Microsoft Access 2003。
- **第10章** 本科生信息管理系统。该章将向读者介绍一个本科生信息管理系统的完整开发过程，其中主要涉及到 Outlook 式界面的编写，应用 ADO 对象操纵数据库的技术，并在 ADO 的技术中灵活运用了 OLE DB 的数据库链接操纵方法、多条 SQL 查询语句、控件 Combo Box 与 ActiveX 控件 ADO Data Control 和 DataGrid Control。
- **第11章** 出租车管理系统。该章通过一个出租车管理系统的完整开发过程，向读者介绍目前流行的 ADO 对象操作技术。在界面设计方面，采用视图切分和树状功能项操纵，列表视图内容显示。数据库采用 Microsoft SQL Server 2000。

本书以学生课程设计和应用系统制作为背景，面向初、中级读者。要求读者已经掌握 C++知识和 Visual C++ 6.0 的最基本语法。

本书由夏崇锴、任海军、余健主要执笔。参与本书编写和程序开发工作的还有蒋宏彪、郭巍、曾峰、韩俊杰、沈海峰、李彬、何佳安、隆左林、蒋兴旺、王维、赵奇、王凯、陈航伟、徐川等。

由于作者水平有限，书中难免有错误，欢迎广大读者提出宝贵的意见。

目 录

第 1 章 Visual C++概述.....	1
1.1 Visual C++简介.....	1
1.2 Visual C++的使用.....	2
1.2.1 创建工程.....	2
1.2.2 代码编写中用到的工具.....	6
1.2.3 调试技巧.....	9
1.2.4 发布程序.....	10
1.3 获得帮助.....	11
1.3.1 MSDN 简介.....	11
1.3.2 MSDN 的使用.....	12
1.4 本章小结.....	14
第 2 章 MFC 基础.....	15
2.1 MFC 编程概述.....	15
2.1.1 什么是 MFC.....	15
2.1.2 MFC 设计原理.....	15
2.1.3 MFC 的层次结构.....	18
2.2 数据交换和校验.....	21
2.2.1 什么是 DDX.....	21
2.2.2 创建 DDX 变量.....	21
2.2.3 DoDataExchange 和 UpdateData 函数.....	22
2.3 读文档和写文档.....	25
2.3.1 什么是序列化.....	25
2.3.2 类的序列化.....	25
2.4 消息映射.....	26
2.4.1 什么是消息映射.....	26
2.4.2 MFC 消息映射的实现方法.....	27
2.4.3 消息的处理.....	28
2.5 本章小结.....	29
第 3 章 GPA 计算器.....	30
3.1 功能描述.....	30
3.2 系统分析与设计.....	30
3.3 关键技术与算法.....	30

3.3.1 CFile 和 CArchive.....	30
3.3.2 公共对话框.....	31
3.4 程序实现.....	31
3.4.1 创建项目.....	31
3.4.2 界面设计.....	32
3.4.3 代码编写.....	34
3.5 运行结果.....	41
3.6 本章小结.....	42
第 4 章 聊天程序.....	43
4.1 功能描述.....	43
4.2 系统分析与设计.....	43
4.3 关键技术与算法.....	44
4.4 程序实现.....	44
4.4.1 项目创建.....	44
4.4.2 界面设计.....	45
4.4.3 代码编写.....	46
4.5 运行结果.....	63
4.6 本章小结.....	64
第 5 章 通讯录.....	65
5.1 功能描述.....	65
5.2 系统分析与设计.....	65
5.3 关键技术与算法.....	66
5.4 程序实现.....	67
5.4.1 项目创建.....	67
5.4.2 界面设计.....	67
5.4.3 代码实现.....	68
5.5 运行结果.....	78
5.6 本章小结.....	79
第 6 章 餐厅管理系统.....	80
6.1 功能描述.....	80
6.2 系统分析与设计.....	80
6.2.1 系统分析.....	80
6.2.2 数据库设计.....	81

6.3 程序实现	84	8.3.3 数据库操纵	167
6.3.1 准备工作	84	8.3.4 树视图的实现	169
6.3.2 创建项目	85	8.3.5 切换视图的实现	172
6.3.3 菜谱管理模块	86	8.3.6 功能对话框的实现	178
6.3.4 餐桌管理模块	92	8.3.7 快捷菜单的实现	200
6.3.5 订菜管理模块	98	8.3.8 切换功能的实现	202
6.3.6 结算管理模块	104	8.4 运行结果	203
6.3.7 主界面模块	109	8.5 本章小结	206
6.4 运行结果	111	第9章 图书馆管理系统	207
6.5 本章小结	113	9.1 功能描述	207
第7章 商品库存管理系统	114	9.2 系统分析与数据库设计	207
7.1 功能描述	114	9.2.1 系统分析	207
7.2 系统分析与设计	114	9.2.2 数据库设计	208
7.2.1 系统分析	114	9.3 程序实现	211
7.2.2 数据库设计	115	9.3.1 准备工作	211
7.3 程序实现	118	9.3.2 创建项目	212
7.3.1 创建项目	118	9.3.3 登录模块	213
7.3.2 数据库操纵	119	9.3.4 图书信息模块	216
7.3.3 登录对话框	120	9.3.5 读者信息模块	226
7.3.4 主界面的实现	123	9.3.6 借书服务模块	235
7.3.5 商品操作对话框	127	9.3.7 还书服务模块	238
7.3.6 供货商家对话框	131	9.3.8 主界面模块	244
7.3.7 入库对话框	136	9.4 运行结果	246
7.3.8 出库对话框	140	9.5 本章小结	248
7.3.9 信息显示对话框	144	第10章 本科生信息管理系统	249
7.3.10 预警对话框	152	10.1 功能描述	249
7.3.11 日志管理类	155	10.2 系统分析与设计	249
7.4 运行结果	156	10.2.1 系统分析	249
7.5 本章小结	159	10.2.2 数据库设计	250
第8章 旅馆管理系统	160	10.3 程序实现	252
8.1 功能描述	160	10.3.1 项目创建	252
8.2 系统分析与设计	160	10.3.2 学生信息模块	253
8.2.1 系统分析	160	10.3.3 课程信息模块	268
8.2.2 数据库设计	161	10.3.4 选课信息模块	282
8.3 程序实现	164	10.3.5 成绩查询模块	297
8.3.1 项目创建	164	10.3.6 主界面模块	302
8.3.2 主框架的设计	164	10.4 运行结果	306

10.5 本章小结	308	11.3.5 切换视图的实现	324
第 11 章 出租车管理系统	309	11.3.6 功能对话框的实现	338
11.1 功能描述	309	11.3.7 登录对话框	359
11.2 系统分析与数据库设计	309	11.4 运行结果	362
11.2.1 系统分析	309	11.5 本章小结	364
11.2.2 数据库设计	310	附录 A VC++ 6.0 常用的调试方法	365
11.3 程序实现	314	A.1 VC++ 6.0 调试环境	365
11.3.1 创建项目	314	A.2 VC++ 6.0 调试步骤	367
11.3.2 主框架的设计	314	A.3 VC++ 6.0 调试技巧	367
11.3.3 数据库操纵	319		
11.3.4 树视图的实现	320		

第 1 章 Visual C++概述

Visual C++是在 Windows 操作系统平台中进行 32 位应用程序开发的重要工具。与其他的编程工具相比, Visual C++在提供了可视化编程方法的同时, 也提供了编写系统底层操作的功能, 其生成代码的质量是其他开发工具所无法比拟的。

本章将介绍如何使用 Visual C++提供的各种便利工具来创建应用程序、编辑和调试程序, 以及发布程序。

1.1 Visual C++简介

标准的 C++是从 C 演化而来的。而 C 语言是从两种早期的程序设计语言 BCPL 和 B 发展而来。

Martin Richards 于 1967 年开发了 BCPL 语言, 他用 BCPL 来编写操作系统软件和编译器。Ken Thompson 在他的 B 语言中模拟了 BCPL 的许多特性, 并于 1970 年在贝尔实验室的 DEC PDP-7 计算机上用 B 语言开发了 Unix 操作系统的早期版本。BCPL 和 B 是“无类型”的语言。例如, 每一个数据在内存中占有一个“字”的空间, 而数据是作为整体处理还是作为实数处理则是程序员的责任。

C 语言是由贝尔实验室的 Dennis Ritchie 开发的。1972 年, C 语言在 DEC PDP-11 计算机上开发完成。C 语言采用了 BCPL 和 B 的许多重要概念, 并增加了数据类型和其他特性。20 世纪 70 年代末, C 语言发展成了现在的“传统的 C”、“经典的 C”或“Kernighan 和 Ritchie C”。1978 年, Prentice-Hall 出版了 Kernighan 与 Ritchie 合著的《The C Programming Language》一书。这本书引起了人们对 C 语言的普遍关注。

C 语言最初是作为 Unix 操作系统的开发语言而被人们所广泛认识的。C 语言在多种计算机(硬件平台)上的广泛使用导致出现了它的各种相似变种。这些变种语言经常互不兼容。这给需要开发在不同平台上使用的可移植程序的程序员带来了很大的麻烦。很明显, 一个标准的 C 语言版本是必需的。1983 年, 美国国家标准协会(American National Standards Institute, ANSI)成立了关于计算机和信息处理的 X3J11 技术委员会, 该委员会致力于制定一个明确的、与设备无关的语言定义。1989 年, 该标准被通过。后来 ANSI 与国际标准化组织(ISO)在 C 语言的标准化问题上展开了合作; 1990 年, 联合标准文件公开发布, 该文件称为 ANSI/ISO 9899:1990。1988 年, Kernighan 和 Ritchie 在《The C Programming Language》第二版中介绍了这个 C 语言版本, 称为 ANSI C。现在, ANSI C 已成为世界各地广为使用的语言。

C++是 C 语言的扩展, 它是由贝尔实验室的 Bjarne Stroustrup 在 20 世纪 80 年代初开发的。C++提供了许多“修饰”C 语言的特性。C++最重要的特点是具有面向对象的程序设计能力。

在当时那个时代, 虽然对更强大的新软件有很高的需求, 但是快速、正确、经济地开

发软件仍是一个可望而不可及的目标。对象实质上是可重用软件组件，它可用于模拟现实世界的事务。软件开发者们发现，利用模块化、面向对象的程序设计和开发方法可以大大提高软件开发团队的效率。这是以往的程序开发技术(如结构化程序设计)所无法做到的。面向对象的程序更容易理解、纠正和修改。

Visual C++开发于 20 世纪 90 年代初。它是 Microsoft 公司的 C++实现工具，其中包含了 Microsoft 公司对于 C++语言所做的专用扩展。在过去的 10 年中，Microsoft 公司发布了 Visual C++的多个版本。Visual C++是一种可视化编程语言——开发者用图形工具来创建应用程序。

本书在使用 Visual C++进行图形用户界面(Graphical User Interface, GUI)程序设计时，是通过调用 Microsoft 基础类库(Microsoft Foundation Class, MFC)来实现的。Microsoft 基础类库是微软开发的 C++类的集合，用来帮助程序员编写功能强大的 Windows 应用程序。它相当彻底地封装了 Win32 软件开发工具包(Software Development Kit, SDK)中的结构和函数。而市面上新推出的 Visual C++ .NET 除了支持 MFC 编程外，还可以使用 Microsoft 公司提供的另外一种库(.NET 的 FCL)来实现 GUI、图形、网络、多线程和其他功能。这个库可用于 .NET 兼容的语言，如 Visual C++ .NET、Visual Basic .NET 和 Microsoft 的新语言 C#。

因为本书是围绕如何用 MFC 编写各种应用程序的主题来编写的，并不涉及 .NET 编程，所以在选择 Visual C++版本时采用的是在 MFC 编程方面比较成熟的 Visual C++ 6.0。

1.2 Visual C++的使用

Visual C++是一个可视化编程工具，它为程序员在编程的各个环节以工具的形式使用 MFC 提供了便利。只有熟练地掌握这些工具的使用，才能顺利地进行编程。本节将按照编程的步骤依次介绍创建工程、编写代码、调试程序、发布程序等方面的知识。

1.2.1 创建工程

在编写 MFC 应用程序时，通常使用 AppWizard 来生成应用程序框架，然后在此框架的基础上添加应用程序特定功能的代码。

在 Visual C++中，是以工作空间(Workspace)和工程(Project)来组织文件和进行工作的。工作空间位于这个结构的最顶层，因此，我们首先要创建一个工作空间。

(1) 从菜单栏中选择 File→New 命令，如图 1-1 所示。

(2) 在 New 对话框中选择 Projects 选项卡，如图 1-2 所示。在 Project name 文本框中输入工程名(工程的命名在一定程度上是任意的)，在 Location 文本框中输入保存工程的文件夹路径，然后在左边的列表中确认选择了 MFC AppWizard(exe)，在 Platforms 列表中确认选择了 Win32。

(3) 完成之后，单击 OK 按钮进入下一步。随后弹出如图 1-3 所示的对话框。根据实际需要选择合适的应用程序类型。

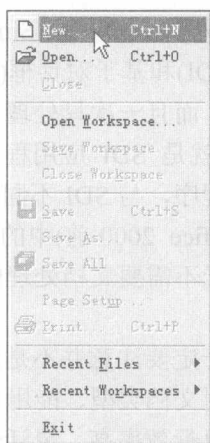


图 1-1 文件菜单中的 New 命令

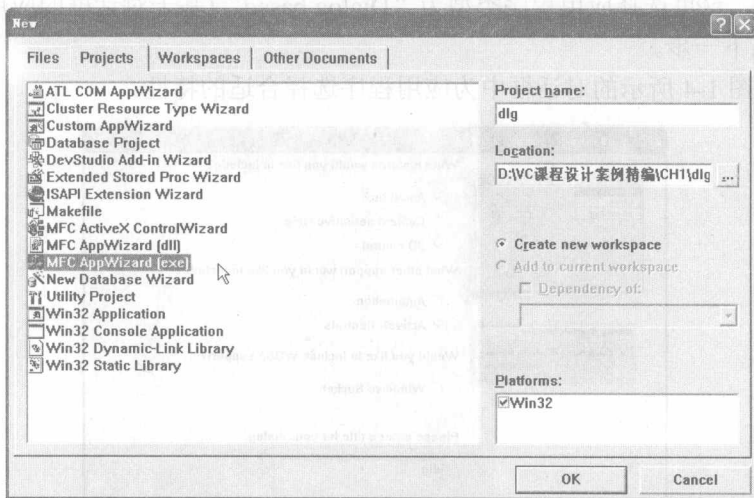


图 1-2 新建工程对话框

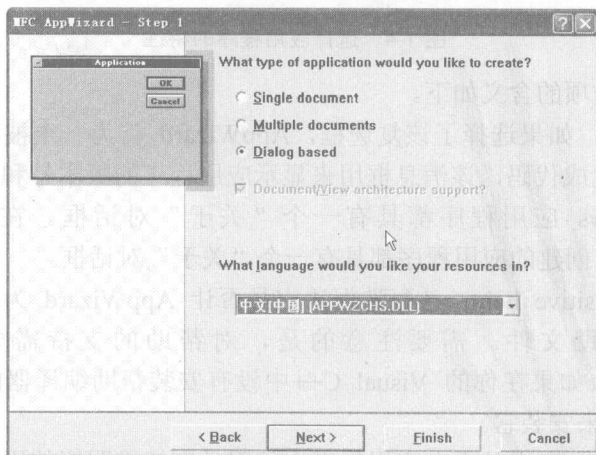


图 1-3 选择工程类型

MFC 支持三种不同类型的应用程序：单文档(Single Document Interface, SDI)、多文档(Multiply Document Interface, MDI)和基于对话框(Dialog based)的程序。“单文档”适用于需要处理文档对象的应用程序，而且一次只处理一个文档对象。我们经常在 Windows 中使用的记事本(Windows Notepad)就是 SDI 应用程序的一个例子。“多文档”适用的对象同样是需要处理文档对象的应用程序，与 SDI 不同的是，它可以一次处理任意数量的文档。Microsoft Office 的早期版本(Office 2000 前)中的 Microsoft Word 就是 MDI 应用程序的一个例子。“基于对话框”适用于不需要文档处理的大多数应用程序。Windows 中的计算器就是基于对话框的应用程序。

在我们选择应用程序类型时，一定要注意并不是选择越高级的类型越好。事实上，正确的做法是能用对话框类型就绝不用文档类型、能用单文档类型就绝不用多文档类型。

基于对话框的应用程序由于其执行效率高，灵活性强，已成为目前使用最为广泛的应用程序类型。

作为示例，这里选择应用程序类型为“Dialog based”(基于对话框的应用程序)，单击 Next 按钮进入下一步。

(4) 在如图 1-4 所示的对话框中为应用程序选择合适的特性。

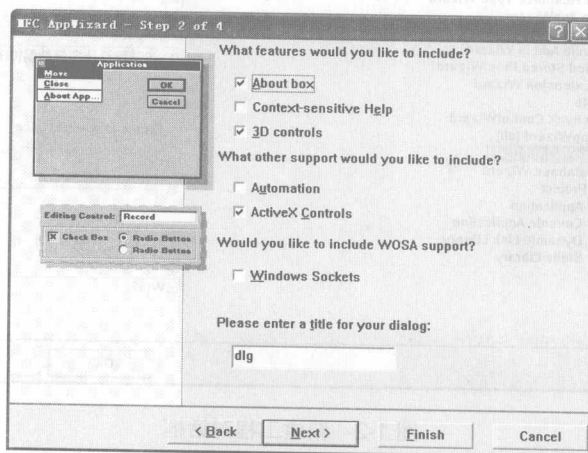


图 1-4 选择应用程序的特性

该对话框中各选项的含义如下。

- **About box:** 如果选择了该复选框，AppWizard 将为一个被称为“关于”对话框的消息框生成代码，该消息框用来显示应用程序的版本号和版权信息等。绝大多数 Windows 应用程序都具有一个“关于”对话框。在默认情况下，使用 AppWizard 创建的应用程序都具有一个“关于”对话框。
- **Context-sensitive help:** 该复选框决定是否让 AppWizard 为应用程序创建上下文相关的帮助文件。需要注意的是，对帮助的支持需要帮助编译器(Help Compiler)，如果在你的 Visual C++ 中没有安装帮助编译器的话，可以重新运行 Setup 程序来安装它。
- **3D controls:** 决定是否在应用程序中使用具有三维阴影的用户界面。使用了三维阴影外观的应用程序界面看起来像那些在 Windows 的早期版本(如 Windows 3.x)

下运行的程序的外观。默认情况下,使用 AppWizard 生成的应用程序都具有三维外观。

- **Automation:** 决定应用程序是否可以操作由其他程序实现的对象,也就是说是否可以将程序作为自动化客户(Automation Client)。
- **ActiveX Controls:** 决定你的应用程序是否使用 ActiveX 控件。如果在创建应用程序框架时没有选择该复选框,则必须在 InitInstance 成员函数中添加对 AfxEnableControlContainer 的调用来向工程插入 ActiveX 控件。在默认情况下,使用 AppWizard 创建的应用程序可以使用 ActiveX 控件。
- **Windows Sockets:** 该复选框决定应用程序是否支持 Windows 套接字。Windows 套接字允许应用程序之间通过基于 TCP/IP 的网络进行通信。

对话框的最下部是对话框所用的标题。在默认情况下 AppWizard 将工程名作为对话框的标题。

设置完成后单击 Next 按钮继续。

(5) 在如图 1-5 所示的对话框中决定是否让 AppWizard 为所创建的源文件加上注释和使用哪一种链接方式。可以使用两种方式链接到 MFC,一种方式是使用动态链接库 DLL,即选择 As a shared DLL;第二种方式是像过去所常用的方式那样,使用 MFC 的静态链接,即选择 As a statically linked library。两种方式各有优缺点,使用 DLL 可以有效地减少应用程序文件的大小,但在运行时必须保证应用程序能够找到所需的动态链接库。使用静态链接的应用程序不需要额外的动态链接库的支持,而且可以在性能上获得少许的提高,但是,使用静态链接的应用程序的可执行文件可能会大很多倍。

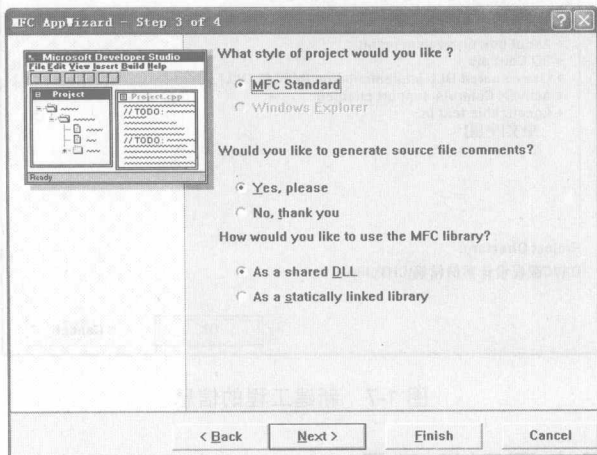


图 1-5 选择链接方式

设置完成后单击 Next 按钮继续。

(6) 在如图 1-6 所示的对话框中更改 AppWizard 创建的类的类名、基类及实现该类的头文件和实现文件。要注意的是,并不是所有类的所有项都可以更改,如果某一项所对应的文本框中内容的显示变为灰色,则表示该项内容不可以修改。

(7) 单击 Finish 按钮。这时,AppWizard 弹出如图 1-7 所示的对话框,该对话框给出了对你在前几步所做的设置的总结,如果一切正确无误的话,单击 OK 按钮接受,这里

AppWizard 根据你的设定定制应用程序的各个代码文件和资源文件。反之,如果在前面的某一步中的设置有误,则可以单击 Cancel 按钮,然后重新运行 AppWizard。

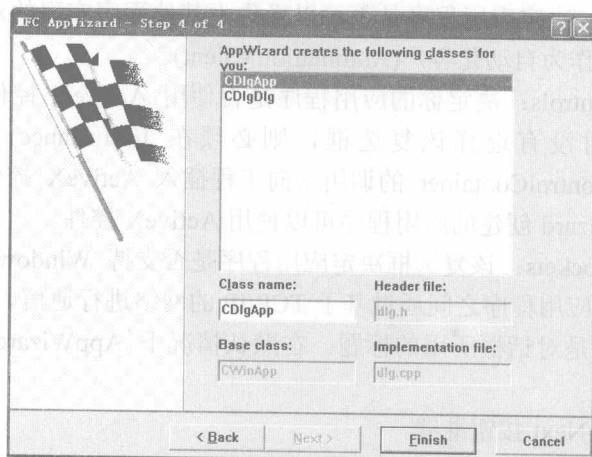


图 1-6 更改生成的类

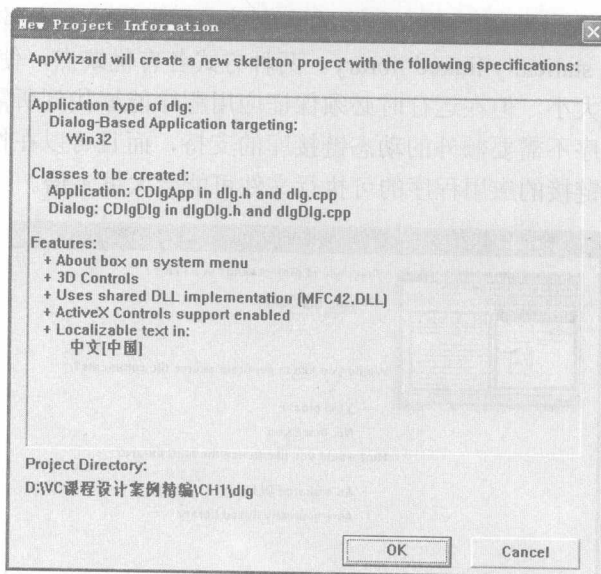


图 1-7 新建工程的信息

1.2.2 代码编写中用到的工具

我们在编辑代码时最频繁使用的恐怕是工作空间(Workspace)和类向导(ClassWizard)了。这两个工具涉及到如何增加消息映射,如何关联成员变量(DDX)等编程技巧,由于本章着重介绍 Visual C++ 这个开发工具的使用,所以这里只简单介绍这些工具的外观及作用。至于涉及到的编程技巧,将在第 2 章中进行详细讲解。

工作空间一般含有三个视图,分别是类视图(ClassView)、资源视图(ResourceView)和文件视图(FileView)。

类视图提供了一个层次清晰的树状视图来展现程序的类结构,如图 1-8 所示。

通过该视图,我们可以很方便地定位到某个类进行编辑,同时关于类、成员变量、重载等操作大多都是通过它来进行的。

资源视图按照资源的不同来展现各种资源,如图 1-9 所示。

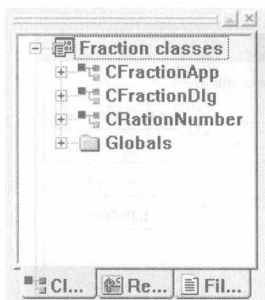


图 1-8 类视图

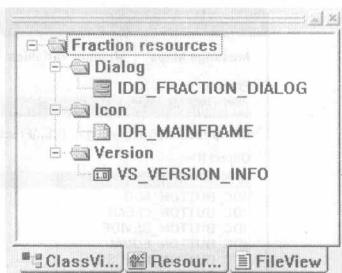


图 1-9 资源视图

通过资源视图,我们不仅可以对相应的资源进行修改,而且可以做一些简单的消息映射。如针对按钮的单击事件,我们可以双击按钮资源来完成消息映射。

文件视图如图 1-10 所示,包含用于创建应用程序的全部文件。

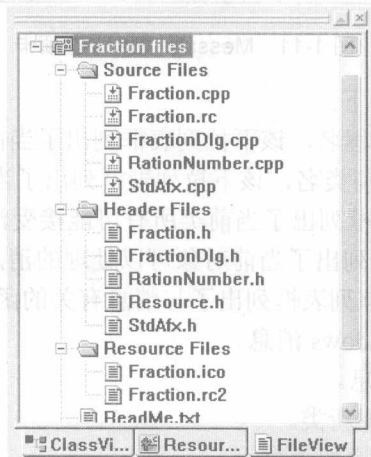


图 1-10 文件视图

通过文件视图,可以清晰地查看程序的实现文件、头文件和资源文件。

类向导是集成开发环境的重要工具。打开或新建一个 MFC 工程后,从菜单栏中选择 View→ClassWizard 命令(或按 Ctrl+W 快捷键),会弹出 MFC Class Wizard 对话框,此对话框包括下面几个选项卡。

- Message Maps: 用来管理消息和成员之间的联系。
- Member Variables: 用来管理与控件相关联的成员变量。
- Automation: 提供各种支持 Automation 的特性。
- ActiveX Events: 提供支持 ActiveX 的各种特性。
- Class Info: 显示类信息。

前面两个选项卡使用得比较多，下面予以详细介绍。

1. Message Maps 选项卡

Message Maps 选项卡可以让用户增加成员函数来控制消息，也可以删除成员函数和编辑成员函数，如图 1-11 所示。

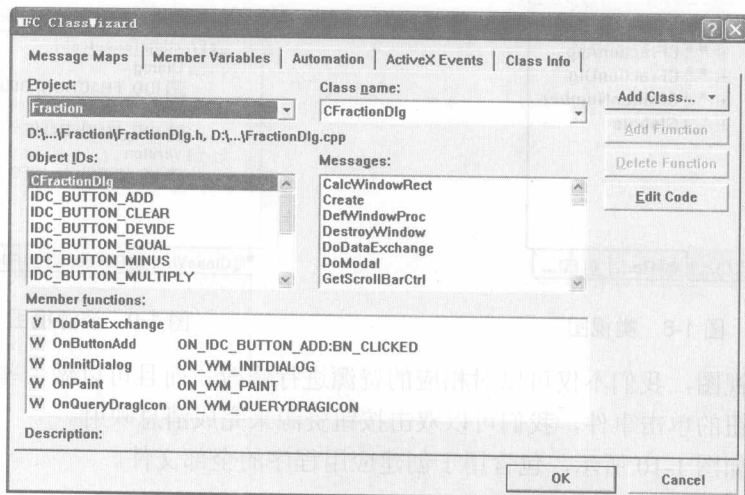


图 1-11 Message Maps 选项卡

界面中各个项目的含义如下。

- Project: 显示当前工程名，该下拉列表框列出了当前工作空间的所有工程。
- Class name: 显示当前类名，该下拉列表框列出了当前工程的所有类。
- Objects IDs: 该列表框列出了当前类所有可能接受消息的对象的 ID。
- Messages: 该列表框列出了当前对象可以处理的消息及可以被重写的虚函数。
- Member functions: 该列表框列出了与消息有关的函数，标注“V”的是虚函数，标注“W”的是 Windows 消息。
- Description: 说明信息。
- Add Class 按钮: 增加新类。
- Add Function 按钮: 增加函数。
- Delete Function 按钮: 删除函数。
- Edit Code 按钮: 编辑函数。

2. Member Variables 选项卡

Member Variables 选项卡允许用户管理与控件相关联的变量，以便程序利用这些成员变量同这些控件进行信息交换，如图 1-12 所示。界面中各个项目的含义如下。

- Project: 选择当前工程名。
- Class name: 选择当前类名。
- Control IDs: 当前类关联的窗口的控件。
- Type: 成员变量的类型。
- Member: 与控件关联的成员变量的名字。

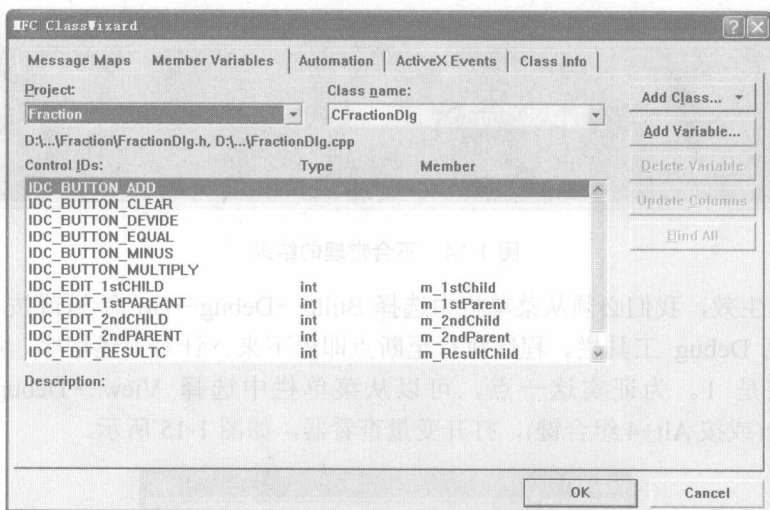


图 1-12 Member Variables 选项卡

1.2.3 调试技巧

Visual C++内含一个精巧的调试器。这个调试器让我们很方便地设定断点(程序执行至此会暂停)、观察变量内容/缓存器内容,并允许在调试过程中改变变量的值。这里将以一个小例子教大家如何使用调试器。

欲使用调试器,首先得保证当前的程序是一个调试的版本。

如图 1-13 所示,在 Build 工具栏中选择“Win32 Debug”模式,然后再进行建造(Building)工作,即可获得调试版本。

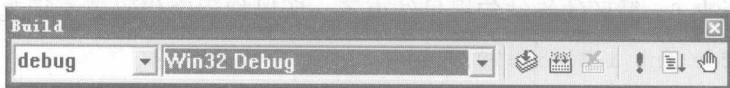


图 1-13 Build 工具栏

现在假设有下面这样一个程序,用来计算 10!的值:

```
#0001 #include <iostream.h>
#0002 int main(int argc, char *argv[])
#0003 {
#0004     int result;
#0005     for(int i=1; i<10; i++)
#0006         result *= i;
#0007     cout << "10!=" << result << endl;
#0008     return 0;
#0009 }
```

程序预期的结果是 $10!=3628800$, 而得到的结果却是如图 1-14 所示的情形。

程序没有语法错误,而运行的结果却不是我们预期的。这时候,我们必须设置断点(Breakpoint);程序暂停之际,我们可以通过观察有嫌疑的变量来排除问题。设置断点的方法是:把光标移到目的行,单击工具栏上的手形按钮(或按 F9 键),于是该行前面出现红点,表示断点被设立。F9 键是一个切换开关,在同一行再按一次 F9 键就能够清除断点。