



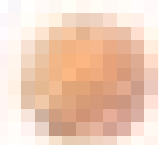
高职高专“十一五”规划教材
计算机系列·计算机软件技术专业

Visual C++ 程序设计教程

李永波 主 编

炎士涛 许利军 副主编

国防科技大学出版社



清华大学出版社
Tsinghua University Press

Visual C++ 程序设计教程

清华大学出版社
清华大学出版社

清华大学出版社

高职高专“十一五”规划教材

计算机系列·计算机软件技术专业

Visual C++ 程序设计教程

李永波 主 编

炎士涛 许利军 副主编

国防科技大学出版社

【内容简介】本教材是为高职高专计算机及相关专业编写的教材。

本书较系统、全面地介绍了 Visual C++ 的相关概念和技术,主要包括 Visual C++ 集成开发环境、C++ 语言基础和面向对象技术、MFC 中的资源与界面设计、对话框设计和 Windows 常用控件、菜单和工具栏设计、属性单和属性页、框架窗口、文档、视图、图形设备接口以及数据编程方面的知识。

本教材适合高职高专学生使用,也可作为学习 Visual C++ 程序设计的自学教程。

图书在版编目(CIP)数据

Visual C++ 程序设计教程/李永波主编. —长沙:国防科技大学出版社,2009.3

(高职高专“十一五”规划教材·计算机系列)

ISBN 978-7-81099-532-0

I. V… II. 李… III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 180371 号

出版发行:国防科技大学出版社

网 址: <http://www.gfkdcbs.com>

责任编辑:耿 筠 特约编辑:李志敏

印 刷 者:三河市骏杰印刷厂

开 本:787mm×1 092mm 1/16

印 张:17.5

字 数:437 千字

印 次:2009 年 2 月第 1 次印刷

定 价:27.00 元

高职高专“十一五”规划教材·计算机系列

编审委员会

- 顾问** 郑启华 清华大学教授
计算机教育资深专家
- 主任** 黄维通 清华大学计算机科学与技术系
全国计算机基础教育研究会副秘书长
- 副主任** 李俊 清华大学信息科学技术学院
骆海峰 北京大学软件与微电子学院
梁振方 上海交通大学电子信息与电气工程学院
- 委员** (以姓氏笔画为序)
- | | | | | |
|-----|-----|-----|-----|-----|
| 卫世浩 | 王玉芬 | 王军号 | 王建平 | 卢云宏 |
| 付俊辉 | 朱广丽 | 刘庆杰 | 刘春霞 | 江枫 |
| 李永波 | 李光杰 | 李克东 | 李学勇 | 张春飞 |
| 张岩 | 郑义 | 姚海军 | 高国红 | 徐桂保 |
| 殷晓波 | 程华安 | 谢广彬 | 詹林 | |
- 课程审定** 张歆 清华大学信息科学技术学院
战扬 北京大学软件与微电子学院
- 内容审定** 倪铭辰 清华大学信息科学技术学院
谢力军 北京大学软件与微电子学院
李振华 北京航空航天大学计算机学院

出版说明

高职高专教育作为我国高等教育的重要组成部分,承担着培养高素质技术、技能型人才的重任。近年来,在国家和社会的支持下,我国的高职高专教育取得了不小的成就,但随着我国经济的腾飞,高技能人才的缺乏越来越成为影响我国经济进一步快速健康发展的瓶颈。这一现状对于我国高职高专教育的改革和发展而言,既是挑战,更是机遇。

要加快高职高专教育改革的步伐,就必须对课程体系和教学模式等问题进行探索。在这个过程中,教材的建设与改革无疑起着至关重要的基础性作用,高质量的教材是培养高素质人才的保证。高职高专教材作为体现高职高专教育特色的知识载体和教学的基本工具,直接关系到高职高专教育能否为社会培养并输送符合要求的高技能人才。

为促进高职高专教育的发展,加强教材建设,教育部在《关于全面提高高等职业教育教学质量的若干意见》中,提出了“重点建设好3 000种左右国家规划教材”的建议和要求,并对高职高专教材的修订提出了一定的标准。为了顺应当前我国高职高专教育的发展潮流,推动高职高专教材的建设,我们精心组织了一批具有丰富教学和科研经验的人员成立了高职高专“十一五”规划教材编审委员会。

编审委员会依据教育部高教司制定的《高职高专教育基础课程教学基本要求》和《高职高专教育专业人才培养目标及规格》,调研了百余所具有代表性的高等职业技术学院和高等专科学校,广泛而深入地了解了高职高专的专业和课程设置,系统地研究了课程的体系结构,同时充分汲取各院校在探索培养应用型人才方面取得的成功经验,并在教材出版的各个环节设置专业的审定人员进行严格审查,从而确保了整套教材“突出行业需求,突出职业的核心能力”的特色。

本套教材的编写遵循以下原则:

(1) 成立教材编审委员会,由编审委员会进行教材的规划与评审。

(2) 按照人才培养方案以及教学大纲的需要,严格遵循高职高专院校各学科的专业规范,同时最大程度地体现高职高专教育的特点及时代发展的要求。因此,本套教材非常注重培养学生的实践技能,力避传统教材“全而深”的教学模式,将“教、学、做”有机地融为一体,在教给学生知识的同时,强化了对学生实际操作能力的培养。

(3) 教材的定位更加强调“以就业为导向”,因此也更为科学。教育部对我国的高职高专教育提出了“以应用为目的,以必需、够用为度”的原则。根据这一原则,本套教材在编写过程中,力求从实际应用的需要出发,尽量减少枯燥、实用性不强的理论灌输,充分体现出“以行业为导向,以能力为本,以学生为中心”的风格,从而使本套教材更具实用性和前瞻性,与就业市场结合也更为紧密。

(4) 采用“以案例导入教学”的编写模式。本套教材力图突破陈旧的教育理念,在讲解的过程中,援引大量鲜明实用的案例进行分析,紧密结合实际,以达到编写实训教材的

目标。这些精心设计的案例不但可以方便教师授课,同时又可以启发学生思考,加快对学生实践能力的培养,改革人才的培养模式。

本套教材涵盖了公共基础课系列、物流管理系列、计算机系列、财经管理系列、电子信息系列、机械系列和化学化工系列的主要课程。目前已经规划的教材系列名称如下:

公共基础课系列

- 公共基础课

计算机系列

- 公共基础课
- 计算机专业基础课
- 计算机网络技术专业
- 计算机软件技术专业
- 计算机应用技术专业

电子信息系列

- 公共基础课
- 应用电子技术专业
- 通信专业
- 电气自动化专业

化学化工系列

- 化学基础课

物流管理系列

- 物流管理专业

财经管理系列

- 工商管理专业
- 财务会计专业
- 经济贸易专业
- 财政金融专业
- 市场营销专业

机械系列

- 机械基础课
- 机械设计与制造专业
- 数控技术专业
- 模具设计与制造专业
- 机电一体化专业

对于教材出版及使用过程中遇到的各种问题,欢迎您通过电子邮件及时与我们取得联系(联系方式详见“教师服务登记表”)。同时,我们希望有更多经验丰富的教师加入到我们的行列当中,编写出更多符合高职高专教学需要的高质量教材,为我国的高职高专教育做出积极的贡献。

高职高专“十一五”规划教材编审委员会

序

21 世纪是科技和经济高速发展的重要时期。随着我国经济的持续快速健康发展, 各行各业对高技能专业型人才的需求量迅速增加, 对人才素质的要求也越来越高。高职高专教育作为我国高等教育的重要组成部分, 在加快培养高技能专业型人才方面发挥着重要的作用。

与国外相比, 我国高职高专教育起步时间短, 这种状况与我国经济发展对人才大量需求的现状是很不协调的。因此, 必须加快高职高专教育的发展步伐, 提高应用型人才的培养水平。

高职高专教育水平的提高, 离不开课程体系的完善。相关领域人才的培养需要一批兼具前瞻性和实践性的优秀教材。教育部高教司针对高职高专教育人才培养模式提出了“以就业为导向”的指导思想, 这也正是本套高职高专教材的编写宗旨和依据。

如何使高职高专教材既突出行业的需求特点, 又突出职业的核心能力? 这是教材编写的过程中必须首先解决的问题。本系列教材编委会深入研究了高职高专教育的课程和专业设置, 并对以往的教材进行了详细分析和认真考察, 力图在不破坏教材系统性的前提下, 加强教材的创新和实践性内容, 从而确保学生在学习专业知识的同时多动手, 增强自己的实践能力, 以加强“知”与“行”的结合。

同时, 本系列教材在编写过程中还充分重视群体和类别的差异性, 面对不同学校 and 不同专业方向的定位差异, 精心设计了与其相配套的辅助实验指南及相关的习题解答等。通过这些栏目的设计, 使本系列教材内容更加丰富, 条理更为清晰, 为老师的讲授和学生的学习都提供了很大的便利。

经过编委会的辛勤努力, 本套教材终于顺利出版了, 相信本套教材一定能够很好地适应现代高职高专教育的教学需求, 也一定能够在高职高专教育计算机课程的改革中发挥积极的推动作用, 为社会培养更多优秀的应用型人才。

全国计算机基础教育研究会副秘书长



前 言

Visual C++ 是 Microsoft 公司推出的基于 Windows 平台的 C++ 可视化编程环境。在该环境下,程序员可以更好地发挥他们的创造力,快速而有效地开发出功能强大的 Windows 应用程序。此外,利用 Visual C++ 开发数据库、多线程等方面的程序也很容易。因此,Visual C++ 受到很多程序员的青睐,成为众多高校计算机专业的必修课程。

本书突出了针对性和实用性,注重学生基本技能、创新能力和综合应用能力的培养,体现了高等院校非计算机专业学习计算机知识的具体要求。本书既可作为计算机专业课程教材,也可以作为计算机爱好者的自学读本或参考工具书。

全书共分 9 章,主要内容如下:

第 1 章:Visual C++ 集成开发环境,主要让读者了解 Visual C++ 的集成开发环境,并掌握如何使用其进行程序设计。

第 2 章:C++ 语言基础和面向对象,主要讲述了 C++ 语言和面向对象的基础知识,这是学习后续知识的基础。

第 3 章:MFC 中的资源与界面设计,主要介绍 MFC 有关知识以及如何使用 MFC 创建一个好看又实用的界面。

第 4 章:对话框设计和 Windows 常用控件,主要讲述了对话框的创建以及各种常见的 Windows 控件。

第 5 章:菜单和工具栏设计,主要讲述了如何在 Visual C++ 集成开发环境中创建和操控菜单和工具栏。

第 6 章:属性单和属性页,主要讲述了如何创建和使用属性单以及属性页。

第 7 章:框架窗口、文档和视图,分析了应用程序的组成结构,讲述了使用框架窗口、文档和视图的方法和技巧。

第 8 章:图形设备接口,主要介绍有关图形处理的基本原理,并结合实例介绍画笔、画刷、字体、位图、图标和光标等绘图工具的编程方法。

第 9 章:数据库编程,主要介绍数据库的基础知识、关系数据库的基础知识、数据库的 SQL 语言,以及几种访问数据库的方法。

本书由李永波主编,炎士涛、许利军任副主编,参加编写的还有廖文军、陆科、方丽娜、贾海龙、余超、赵春等老师。

本书在编写过程中,得到了相关领导和老师的大力支持,在此向各位领导和老师深表谢意。

由于作者水平有限,加之编写时间仓促,书中难免有疏漏与不足之处,衷心希望各位读者批评指正。

编 者

目 录

第 1 章 Visual C++ 集成开发环境	1
1.1 Visual C++ 概述	1
1.1.1 Visual C++ 的发展史	1
1.1.2 Visual C++ 的特点	1
1.1.3 Visual C++ 6.0 与 MFC	2
1.2 集成环境介绍	3
1.2.1 Developer Studio 的特点	3
1.2.2 Developer Studio 界面	4
1.2.3 菜单	6
1.2.4 工具栏	21
1.2.5 ClassWizard	24
1.3 程序调试	28
1.3.1 查找源程序中的语法错误	28
1.3.2 调试器	29
1.3.3 跟踪调试程序	29
本章小结	30
习题 1	30
第 2 章 C++ 语言基础和面向对象	32
2.1 C++ 语言基础	32
2.1.1 C++ 程序的基本结构	32
2.1.2 数据类型	34
2.1.3 常量和变量	35
2.1.4 运算符	37
2.1.5 C++ 语言的 3 种程序结构	40
2.1.6 函数	42
2.1.7 指针	43
2.1.8 数组	45
2.2 类	47
2.2.1 类及其成员变量与函数的声明和定义	47
2.2.2 构造函数和析构函数	49
2.2.3 友元	52
2.3 面向对象的编程技术	54
2.3.1 面向对象程序设计	54

2.3.2 封装	55
2.3.3 继承	56
2.3.4 多态和虚函数	62
本章小结	66
练习 2	67
第 3 章 MFC 中的资源与界面设计	68
3.1 MFC 基础知识	68
3.1.1 微软基础类库 MFC	68
3.1.2 常用的 MFC 文件	71
3.1.3 MFC 消息管理	73
3.2 了解各类资源	76
3.2.1 资源简介	76
3.2.2 快捷键	76
3.2.3 对话框	77
3.2.4 图标	78
3.2.5 位图	78
3.2.6 菜单	79
3.2.7 字符串	80
3.2.8 工具栏	81
3.3 编程实例——使用资源进行界面设计	81
本章小结	83
练习 3	83
第 4 章 对话框设计和 Windows 常用控件	84
4.1 对话框应用程序介绍	84
4.2 创建对话框程序	85
4.2.1 建立对话框程序框架	85
4.2.2 创建对话框类	87
4.2.3 了解对话框常用函数	90
4.3 使用 Windows 常用控件	91
4.3.1 Windows 常用控件介绍	91
4.3.2 添加 Windows 控件	95
4.3.3 修改 Windows 控件属性	96
4.4 对控件进行排版	97
4.4.1 设置 Windows 控件的大小	98
4.4.2 设置 Windows 控件的对齐模式	99
4.5 模式对话框和非模式对话框	99
4.5.1 模式对话框和非模式对话框概述	99
4.5.2 创建模式对话框	100

4.5.3	创建非模式对话框程序	100
4.6	Windows 通用对话框	101
4.6.1	CFontDialog 对话框	102
4.6.2	CFileDialog 对话框	103
4.6.3	CColorDialog 对话框	104
4.6.4	CPrintDialog 对话框	105
4.6.5	CFindReplaceDialog 对话框	106
4.7	编程实例——对话框和控件使用综合实例	108
	本章小结	115
	练习 4	115
第 5 章	菜单和工具栏设计	116
5.1	创建和编辑菜单	116
5.1.1	定义菜单	116
5.1.2	编写菜单事件处理程序	118
5.1.3	设置菜单的显示效果	121
5.2	快捷键	122
5.2.1	快捷键介绍	123
5.2.2	定义快捷键	123
5.3	动态更改菜单	124
5.3.1	CMenu 类	124
5.3.2	动态更改菜单项	126
5.3.3	动态添加菜单图标	129
5.4	使用上下文菜单	131
5.4.1	建立菜单资源	131
5.4.2	添加上下文菜单资源	131
5.5	工具栏设计	133
5.5.1	创建和初始化工具栏	133
5.5.2	显示和隐藏工具栏	139
5.6	编程实例——编写带有工具栏的画图程序	140
	本章小结	145
	练习 5	146
第 6 章	属性单和属性页	147
6.1	属性单和属性页相关类	147
6.2	创建属性单和属性页程序	150
6.2.1	创建属性页资源	150
6.2.2	创建 CPropertyPage 类	152
6.2.3	创建 CPropertySheet 类	153
6.2.4	显示属性单	155

6.3	创建向导属性单应用程序	157
6.4	编程实例——在属性单中设置字体	161
	本章小结	167
	习题 6	167
第 7 章	框架窗口、文档和视图	168
7.1	应用程序组成结构	168
7.1.1	CMainFrame 类和窗口	169
7.1.2	CDocument 类和文档	170
7.1.3	CView 类和视图	171
7.1.4	框架窗口、文档和视图的关系	173
7.2	文档	176
7.2.1	数据成员	176
7.2.2	成员函数	176
7.2.3	文档序列化	180
7.3	视图	184
7.3.1	视图的特点	184
7.3.2	视图的成员函数	185
7.3.3	视图的操作方法	188
7.4	基础实例——单文档应用程序	188
7.5	提高实例——多文档应用程序	194
	本章小结	201
	练习 7	201
第 8 章	图形设备接口	202
8.1	图形设备接口介绍	202
8.1.1	设备环境类	202
8.1.2	GDI 对象	204
8.1.3	映射模式	207
8.1.4	自定坐标系统	209
8.2	文本和字体	210
8.2.1	将对象选入设备	210
8.2.2	字体的属性和创建	211
8.2.3	输出文本	212
8.3	简单图形绘制	214
8.3.1	创建画笔	215
8.3.2	创建画刷	217
8.3.3	简单图形绘制	219
8.4	基础实例——字体应用程序	220
8.5	提高实例——绘图应用程序	224

本章小结	226
练习 8	227
第 9 章 数据库编程	228
9.1 数据库概述	228
9.1.1 数据库和 DBMS	228
9.1.2 结构化查询语言 SQL	229
9.1.3 数据库访问模式	231
9.2 MFC ODBC 数据库访问技术	233
9.2.1 MFC ODBC 访问数据库过程	233
9.2.2 MFC 的 ODBC 类	236
9.3 ADO 数据库访问技术	240
9.3.1 ADO 数据库访问技术介绍	240
9.3.2 ADO 数据库结构及其对象	241
9.3.3 使用 ADO 数据库对象操作数据库	245
9.4 基础实例——ODBC 数据库应用程序	248
9.5 提高实例——ADO 数据库应用程序	251
本章小结	258
练习 9	258
参考文献	259

第 1 章 Visual C++ 集成开发环境

开发环境是程序员同 Visual C++ 交互的界面,程序员通过开发环境可以进行创建工程文件、创建或访问 C++ 源代码、查看或修改资源文件等操作。本章简要介绍 Visual C++ 6.0 的用户界面,并对各种常用的窗口、菜单、按钮的意义和功能有重点地进行讲解。

1.1 Visual C++ 概述

Visual C++ 6.0 是当今最流行的软件开发工具之一,其优势主要表现为 3 个方面。一是其强大的功能能够大大加速程序员的工作,提高程序代码的效率。二是由于 Microsoft 的 PC 操作系统在市场上占垄断地位,使用 Visual C++ 6.0 编程软件能够与 Windows 操作系统最大程度地兼容。三是 Visual C++ 6.0 能够非常方便地使用 MFC 所提供的强大功能。

Visual C++ 6.0 并不只是一个程序代码的编辑器和编译器。要想掌握使用 Visual C++ 6.0 编程的方法,首先需要掌握软件的基本操作,这样可以提高软件的使用效率。

1.1.1 Visual C++ 的发展史

C++ 语言的前身 C 语言问世于 20 世纪 70 年代初,由贝尔实验室发布。C 语言的功能强大,优点诸多,因此逐渐得到广泛应用,成为当代最优秀的程序设计语言之一。

在 C 语言的使用过程中,人们逐渐发现了它的一些缺陷,例如,大规模软件开发过程中的程序结构复杂性的控制、代码的重用以及日益广泛的面向对象分析方法的一致性。为了解决这些难题,1983 年贝尔实验室进一步对 C 语言进行了扩充和完善,使其包含了“类”的结构,特别是它提供了面向对象的程序设计功能,这就是 C++ 语言。

随着 Windows 3.0 的推广,微软公司推出了 Microsoft C++ 7.0,首次采用了微软基础类库(Microsoft Foundation Class Library, MFC)。随后推出的 Microsoft C++ 8.0 改名为 Visual C++,这就是 Visual C++ 1.0。其后推出的各个 Visual C++ 版本都有不同的改进,其中,Visual C++ 6.0 增加了 IE4.0 的控件和类,增强了对 OLE DB 的支持,完善了 DHTML,提供了独立的 MSDN 帮助系统等,同时集成环境内部工具的功能也得到了相应的提升。

1.1.2 Visual C++ 的特点

Visual C++ 6.0 是 Visual Studio 98(Visual Studio 6.0)家族的一个成员,与以前的版本相比,它具有以下新特性:

(1)改进的编译器。包括:延迟的动态库连接输入,各种新的编译参数,可生成更短小的编译程序代码等。

(2)增强的编辑器。包括:关键字提示,更好的初始化控制,新的警告机制,支持操作符 Delete 的自动设置,支持运行时的错误检查等。此外,还能够进行自动声明完成、快速宏记录和支持 IE4.0 的源编辑器。

(3)更方便的跟踪调试工具。包括:新的诊断映射机制,软件形式的错误寄存器,改进的模拟输出,改进的函数指针显示,进程的远程调用,MMX 寄存器显示以及显示动态链接库的对话框等。

(4)新的对象模板。提供了一些新的属性和方法,可以使工程的生成更为灵活。

(5)改进的工程操作。包括:增加了基于命令行的编译方式,动态更新视类,可直接进入对话框编辑器,对 .idl 文件编译的新支持等。

(6)Wizard 工具的新支持。包括:AppWizard 增加了对网络工具条的支持,支持不基于文档/视图结构的 SDI/MDI,新增的对 OLE 的支持,help 文件的客户编译规则,ATL 对象中的数据路径支持,WizardBar 和 ClassView 中直接应用 Delete 命令,支持 IE 控制等。

(7)OLE 数据库模板。

(8)新的 MFC 类库支持。包括:活动的文档内容,对 HTML 控制的动态支持,支持 IE4.0 的一般控制,加入了支持 OLE 数据库的类等。

(9)新的数据库支持。包括:ADO 的数据捆绑对话框工具,ADO 和 OLE 的数据捆绑控制支持,数据工具中的 Oracle 支持。

(10)新的例程。包括:ATL 中添加了 4 个新例程,MFC 中添加了 11 个新例程等。

(11)新的工具。包括:对象管理器,网络 HTML 帮助,更新的安装界面等。

这些新特性使 Visual C++ 6.0 更加适应现在计算机网络化、运行速度加快以及数据传输加强的趋势,成为软件开发的首选工具之一。

1.1.3 Visual C++ 6.0 与 MFC

MFC 是 Microsoft Foundation Classe Library(微软基础类库)的缩写,它提供了面向对象的框架,程序开发人员可以使用这一框架创建 Windows 应用程序。MFC 按照 C++ 类的层次形式组织在一起,几个高层类提供一般功能,而低层类实现更具体的行为。每一个低层类都是从高层类中派生而来的,因此继承了高层类的行为,同时又有自己特殊的功能。

例如,CWnd 类是实现所有 Windows 窗口中的大多数常用功能的高层类,此类具备的功能包括打印文本、绘制图形以及跟踪鼠标指针的移动等。CSplitterWnd 类是从 CWnd 中派生而来的,因此继承了 CWnd 的功能。除此之外,CSplitterWnd 还能实现一种被称为拆分器的特殊窗口,拆分器窗口是至少被拆分为两个部分的窗口,用户可以移动这两部分之间的边框。例如,Windows 资源管理器就是使用拆分器窗口来显示有关计算机资源的信息的。

MFC 可以处理很多与 Windows 相关的常见任务,如在后台做消息处理以及路由选择等。在程序编写过程中,无需在所开发的每一个 Windows 应用程序中编写相同的消息处理循环,MFC 可实现这一消息循环,并提供易学易用的成员函数,如 OnPaint() 函数能插入代码,处理窗口消息。

除类的层次外,MFC 还提供一个应用程序开发模型,此模型被称为文档/视图模型。文档/视图模型是将应用程序数据与用户界面元素分离的一种应用编程方法,它允许这两部分独立存在,这样程序员在更改其中一部分的时候,不会大量更改另一部分。

文档/视图模型有很多优点,绝大多数应用程序都可以使用这一模型。例如,一个字处

理程序或电子表格程序中包含的数据是文档/视图模型中的文档,而数据显示部分是视图。文档类中包含允许视图类与应用程序数据一起运行的成员函数,只要这些函数本身不发生改变,用户界面也就无需改变。事实上,该文档甚至无须是一个字处理文档或电子数据表,它可以是数据库中的记录或图形图像,只要成员函数不发生改变,使用什么类型的信息无关紧要。

MFC 向程序开发人员提供一组服务来协助开发应用程序。在很多情况下,MFC 在幕后执行复杂的操作,而向开发人员提供更简单的编程界面,并可以掩盖 Windows API 的一些缺点。

MFC 面向对象设计将 Windows API 中的相关部分组合到 C++ 类或对象中。例如,CWnd 类封装了大部分 API 编程功能,因此作为编程人员,就不必记忆大量 API 函数的调用,只需要实例化 CWnd 类的一个新实例并调用几个成员函数即可。

1.2 集成环境介绍

单击“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Visual C++ 6.0”菜单项,启动 Visual C++ 集成开发环境,出现集成开发环境的主窗口 Developer Studio。

Visual C++ 的 Developer Studio 是 Microsoft 公司为开发 Windows 应用程序而创建的一种集成开发环境,它不但集成有编辑、编译、连接、调试等程序开发工具,而且提供了全方位的自动编程服务工具——编程向导 Wizard(包括 AppWizard 和 ClassWizard)。

1.2.1 Developer Studio 的特点

Developer Studio 具有如下几个特点:

(1) 程序的开发过程是由应用程序向导 AppWizard 引导完成的。

(2) AppWizard 可以自动生成通用的程序框架代码,并把程序中每个程序都用得到而程序员无须看到的部分(如 main() 函数、WinMain() 函数和 AfxWinMain() 函数等)隐藏起来,只开放程序员需要知道的部分。程序员的工作就是在这个框架中添加自己的代码。

AppWizard 可以建立 3 种应用程序结构:

① MDI(Multiple Document Interface): 多文档界面,允许同时打开多个文档。

② SDI(Single Document Interface): 单文档界面,一次只允许打开一个文档。

③ Dialog based: 基于对话的应用程序,不支持文档/视图结构,仅显示一个简单的对话框,主要用于编写小程序。

(3) 提供了一套功能强大的基础类库。

(4) 引入了文档/视图结构。文档和视图是 MFC 的两类对象:文档(Document)对象就是用户处理的数据单位,用来维护、管理数据,包括数据的读取、存储与修改等;视图(View)对象用来接收并显示数据,并将这些数据交给文档类来处理。

(5) 提供强有力的资源管理功能。资源是 Windows 应用程序的重要组成部分。在创建一个 Windows 应用程序后,当打开该程序时,通过 Workspace 进入 xxx resources(xxx 为应用程序名),就进入了该程序的资源编辑器。

(6) 在 MFC 中用指针来标识对象。MFC 维护着一个查询表集合——句柄映像,用以实