



普通高等教育“十三五”规划教材
高等院校计算机系列教材

Visual C++面向对象程序设计

李康满 李 浪 ◎ 主编



华中科技大学出版社

<http://www.hustp.com>

普通高等教育“十二五”规划教材
高等院校计算机系列教材

Visual C++面向对象程序设计

主 编 李康满 李 浪
副主编 王 樱 田小梅 刘新宇
郑光勇 尹友明

华中科技大学出版社
中国·武汉

内 容 提 要

本书是结合多年的教学和实践经验、参考国内外有关著作及文献编写而成的。本书针对初学者的特点,由浅入深、系统地介绍 Visual C++ 面向对象编程的基本原理和方法,主要包括 C++ 面向对象程序设计、可视化编程环境、Windows 编程基础、MFC 编程方法和 Visual C++ 高级编程等内容。本书共分 9 章。第 1 章介绍 Visual C++ 集成开发环境;第 2 章介绍 C++ 语言面向对象编程基础;第 3 章介绍 Windows 应用程序开发的基本原理;第 4 章介绍 MFC 原理与方法;第 5 章介绍文档/视图结构;第 6 章介绍对话框的原理与应用,包括通用对话框的应用;第 7 章详细介绍常用控件;第 8 章介绍图形处理的原理与方法;第 9 章介绍 ODBC 及 Socket 编程的 Visual C++ 的高级应用。为了帮助学生掌握知识的应用,每章均有拓展案例,以指导学生对本章知识进行综合应用;每章后面均配有习题及上机编程题,以帮助学生巩固知识。

本书内容详实,重点、难点突出,所选案例具有较强的代表性,有助于学生举一反三。本书注重理论性和实用性的结合,收集的例题与习题大多是一些应用型的实例。本书内容安排循序渐进,重点突出,实例典型,文字精练。本书特别适合作为大中专院校、各类职业院校及计算机培训学校相关专业课程的教材,还可作为 Visual C++ 应用开发人员的自学读本或参考工具书。

图书在版编目(CIP)数据

Visual C++ 面向对象程序设计/李康满,李浪主编. —武汉:华中科技大学出版社,2019.1

ISBN 978-7-5680-4930-6

I. ①V… II. ①李… ②李… III. ①C++ 语言-程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 008029 号

Visual C++ 面向对象程序设计

李康满 李浪 主编

Visual C++ Mianxiang Duixiang Chengxu Sheji

策划编辑:范莹

责任编辑:朱建丽

封面设计:原色设计

责任校对:刘竣

责任监印:赵月

出版发行:华中科技大学出版社(中国·武汉) 电话:(027)81321913

武汉市东湖新技术开发区华工科技园 邮编:430223

录排:佳思漫艺术设计中心

印刷:武汉市洪林印务有限公司

开本:787mm×1092mm 1/16

印张:21

字数:493 千字

版次:2019 年 1 月第 1 版第 1 次印刷

定价:49.80 元



华中出版

本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

前 言

面向对象的方法日趋完善,其倡导的封装性、继承性和多态性等特性在应用中不断被人们所领悟,并得以提升和推广。尽管新的编程技术和工具不断涌现,但无论程序设计技术如何发展,面向对象程序设计方法仍是当前编程技术的根本和基础,以 MFC 为主的 Visual C++ 技术在桌面应用程序开发方面仍然具有很大的优势。

本书是软件工程学习的基础教程,考虑到计算机及其相关专业的现状,从面向对象的基本概念出发,讲述面向对象程序设计的思想与方法,既有原理性的讲解,也有实例说明和分步骤的编程实现,深入浅出地引导读者思维和实践,既注重学生对基本原理的学习,也注重学生实际应用软件开发能力的培养。本书力求通过实例让学生较好地掌握“面向对象与可视化程序设计”的思路、开发技巧与体系;同时还增加了案例拓展,以培养学生的应用能力。本书内容循序渐进,易于讲解,结构清晰,方便教师组织教学内容。

参加编写工作的有衡阳师范学院的李康满、李浪、王樱、田小梅、刘新宇、郑光勇和衡阳技师学院的尹友明等老师。值得一提的是,本书得到湖南省普通高校教学改革研究项目(湘教通[2018]436号 NO:538)、教育部产学合作协同育人项目(No.201701048037,201702071007)、湖南省普通高校实践教学建设项目、衡阳师范学院“十三五”转型发展项目、“十三五”专业综合改革项目、校校合作项目等的支持,也得到了合作公司及合作院校的实训案例支持,使得本书的案例具有很强的实用性和代表性。其中第1、2章由王樱和尹友明编写,第3章由李浪编写,第4、5章由李康满编写,第6、7章由田小梅编写,第8章由刘新宇编写,第9章由郑光勇编写。本书由李康满和李浪统稿和审稿。在编写期间,参与讨论和代码调试工作的还有梁小满、赵辉煌、焦铭、陈石义、封山河、张琴艳、龙大奇、罗恒辉、周瑛等老师。本书的作者都是多年从事面向对象编程教学和科研的老师,在编写的过程中,参考了国内外大量文献资料,为此向文献的作者表示忠心感谢。尽管我们再三校对,书中可能还存在错误和不足,恳请专家和广大读者指正和谅解。

本书不仅可以作为大中专院校、各类职业院校及计算机培训学校相关专业课程的教材,还可作为软件开发的参考用书。同时,本书已整理好书中实例代码并开发好相应的教学课件,有教学需要的老师可以在华中科技大学出版社的网页上下载,也可发邮件向我们索取,我们的联系方式:lkm@hynu.edu.cn;lilang911@126.com。

编 者

2019年1月

目 录

第 1 章 Visual C++ 集成开发环境	1
1.1 Visual C++ 概述	1
1.2 Visual Studio 集成开发环境	1
1.2.1 Visual Studio 2010 界面	2
1.2.2 项目与解决方案	3
1.3 创建应用程序	5
1.3.1 创建 Win32 控制台应用程序	6
1.3.2 创建 MFC 应用程序	9
1.4 习题	10
第 2 章 C++ 语言面向对象编程基础	11
2.1 一个简单的 C++ 语言程序	11
2.2 C++ 语言程序基本要素	12
2.2.1 关键字和标识符	12
2.2.2 基本数据类型	13
2.2.3 常量与变量	14
2.2.4 运算符与表达式	16
2.3 类与对象	19
2.3.1 面向对象程序设计的基本思想和特点	19
2.3.2 类的定义与实现	20
2.3.3 构造函数和析构函数	22
2.3.4 this 指针	27
2.4 组合类	28
2.4.1 组合类的定义	28
2.4.2 组合类的构造函数	29
2.5 友元	31
2.5.1 友元函数	31
2.5.2 友元类	32
2.6 类的静态成员	34
2.6.1 静态数据成员	34
2.6.2 静态成员函数	36

2.7 继承与派生	37
2.7.1 继承与派生的概念	37
2.7.2 派生类的定义	38
2.7.3 继承的方式	40
2.7.4 派生类的构造函数与析构函数	44
2.7.5 虚基类	48
2.8 重载	52
2.8.1 函数重载	52
2.8.2 运算符重载	55
2.9 多态与虚函数	60
2.9.1 多态的概念	60
2.9.2 虚函数	60
2.9.3 虚析构函数	62
2.9.4 纯虚函数与抽象类	63
2.10 流	64
2.11 拓展案例	65
2.12 习题	68
第3章 Windows 应用程序	74
3.1 Windows 编程基础知识	74
3.1.1 窗口	74
3.1.2 事件驱动	74
3.1.3 句柄	75
3.1.4 Windows 消息	75
3.2 Windows 应用程序常用消息	76
3.3 Windows 中的事件驱动程序设计	77
3.3.1 程序入口函数	77
3.3.2 定义窗口类	78
3.3.3 注册窗口类	78
3.3.4 创建窗口	78
3.3.5 显示窗口	79
3.3.6 消息循环	79
3.3.7 窗口函数	80
3.4 拓展案例	83
3.5 习题	84

第4章 MFC 原理与方法	85
4.1 MFC 的本质	85
4.2 MFC 的组织结构	85
4.2.1 根类	86
4.2.2 应用程序体系结构类	86
4.2.3 可视对象类	87
4.2.4 通用类	89
4.2.5 OLE 类	90
4.2.6 ODBC 类	90
4.2.7 Internet 和网络类	91
4.2.8 调试支持类和异常类	91
4.3 Visual C++ 工程类型	92
4.3.1 ATL Project	92
4.3.2 CLR 项目	92
4.3.3 常规	92
4.3.4 MFC 项目	93
4.3.5 测试	93
4.3.6 Win32	93
4.4 MFC 应用程序向导	94
4.5 MFC 应用程序向导生成的文件	103
4.6 MFC 消息管理	103
4.6.1 MFC 消息映射机制	103
4.6.2 消息类别	104
4.6.3 消息处理	105
4.6.4 添加消息处理函数	106
4.7 MFC 宏和指令	107
4.7.1 #include 指令	108
4.7.2 #define 指令	108
4.7.3 #typedef 指令	108
4.7.4 #ifdef_#else_#endif 指令	108
4.7.5 #Pragma 指令	108
4.7.6 宏:_LINE_和_FILE_	110
4.7.7 调试宏:ASSERT()、VERIFY()、TRACE()	110
4.7.8 消息处理宏	111
4.7.9 消息映射宏	111

4.7.10	宏 <code>DEBUG_NEW</code>	112
4.7.11	异常宏	112
4.7.12	宏 <code>DECLARE_DYNAMIC()</code> 、宏 <code>IMPLEMENT_DYNAMIC()</code>	114
4.7.13	宏 <code>DECLARE_DYNCREATE()</code> 、宏 <code>IMPLEMENT_DYNCREATE()</code>	114
4.7.14	宏 <code>DECLARE_SERIAL()</code> 、宏 <code>IMPLEMENT_SERIAL()</code>	115
4.7.15	宏 <code>RUNTIME_CLASS()</code>	115
4.8	常用的 MFC	115
4.8.1	<code>CRuntimeClass</code> 结构	115
4.8.2	派生类 <code>CObject</code>	116
4.8.3	类 <code>CCmdTarget</code>	116
4.8.4	类 <code>CWinThread</code>	116
4.8.5	类 <code>CWinApp</code>	116
4.8.6	类 <code>CWnd</code>	117
4.8.7	类 <code>CFrameWnd</code>	117
4.8.8	类 <code>CDocument</code> 和类 <code>CView</code>	118
4.9	拓展案例	119
4.10	习题	120
第 5 章	文档/视图结构	122
5.1	文档/视图类	122
5.1.1	文档类	122
5.1.2	单文档和多文档	122
5.1.3	视图类	123
5.1.4	文档/视图/框架结构	124
5.2	文档/视图结构的操作	125
5.2.1	各类间的相互操作	125
5.2.2	视图类 <code>CView</code> 的操作	126
5.2.3	文档类 <code>CDocument</code> 的操作	127
5.3	鼠标和键盘消息处理	128
5.3.1	鼠标事件和鼠标消息	128
5.3.2	鼠标消息处理	129
5.3.3	键盘消息处理	132
5.4	菜单设计	135
5.4.1	菜单资源	136
5.4.2	菜单快捷键	139
5.4.3	弹出式菜单	140

5.5 工具栏和状态栏	142
5.5.1 工具栏	142
5.5.2 状态栏	143
5.6 文档序列化	145
5.6.1 类 CFile	145
5.6.2 类 CArchive	148
5.6.3 序列化函数	148
5.6.4 自定义序列化	150
5.7 拓展案例	154
5.8 习题	158
第6章 对话框	160
6.1 对话框概述	160
6.1.1 对话框资源	160
6.1.2 对话框对象	160
6.1.3 文档对象	160
6.2 基于对话框的应用程序	161
6.3 对话框的使用	167
6.3.1 设计对话框资源	168
6.3.2 生成对话框类	173
6.3.3 运行对话框	182
6.4 对话框类 CDialog 和类 CDialogEx	182
6.4.1 类 CDialog	183
6.4.2 类 CDialogEx	186
6.5 模式对话框和非模式对话框	188
6.5.1 模式对话框	188
6.5.2 非模式对话框	190
6.6 消息对话框	194
6.6.1 消息对话框的弹出	194
6.6.2 消息对话框的返回值	196
6.7 通用对话框	198
6.7.1 文件对话框	199
6.7.2 字体对话框	202
6.7.3 颜色对话框	206
6.8 拓展案例	208
6.9 习题	210

第 7 章 常用控件	212
7.1 控件概述	212
7.1.1 控件的分类	212
7.1.2 控件的组织	214
7.1.3 控件的属性	215
7.1.4 控件的创建	215
7.1.5 控件通知消息	216
7.2 静态控件和按钮控件	217
7.2.1 静态控件	217
7.2.2 按钮控件	220
7.3 编辑框控件和旋转按钮控件	235
7.3.1 编辑框控件	235
7.3.2 旋转按钮控件	239
7.4 列表框控件、组合框控件和滚动条控件	240
7.4.1 列表框控件	240
7.4.2 组合框控件	248
7.4.3 滚动条控件	257
7.5 滑块控件和进度条控件	261
7.5.1 滑块控件	261
7.5.2 进度条控件	262
7.6 日期时间控件和标签控件	263
7.6.1 日期时间控件	263
7.6.2 标签控件	264
7.7 拓展案例	270
7.8 习题	277
第 8 章 图形处理	279
8.1 MFC 绘图基础类 CDC	279
8.1.1 类 CClientDC	280
8.1.2 类 CWindowDC	280
8.1.3 类 CPaintDC	280
8.1.4 类 CMetaFileDC	280
8.2 画笔和画刷	281
8.2.1 用画笔绘制基本几何图形	282
8.2.2 用画刷绘制基本几何图形	285
8.2.3 绘制复杂几何图形	287

8.3 文本和字体	289
8.3.1 获取及设置默认字体	290
8.3.2 输出字体设置	290
8.4 位图、图标和光标	290
8.4.1 位图	291
8.4.2 图标	291
8.4.3 光标	292
8.5 拓展案例	293
8.6 习题	296
第9章 Visual C++的高级应用	297
9.1 Visual C++数据库编程	297
9.1.1 ODBC 的基本概念	297
9.1.2 ODBC 数据库配置过程	298
9.1.3 ODBC 编程实现	299
9.2 Visual C++ Socket 编程	304
9.2.1 面向连接的网络通信	304
9.2.2 Socket 服务器端编程	307
9.2.3 Socket 客户端编程	314
9.3 拓展案例	318
9.4 习题	322
参考文献	323

第 1 章 Visual C++ 集成开发环境

Visual C++ 简称 VC++，是由微软公司提供的 C++ 语言开发工具，它与 C++ 语言的根本区别在于 C++ 是语言，而 Visual C++ 是用 C++ 语言编写的工具平台。Visual C++ 不仅是一个编译器，更是一个集成开发环境，包括编辑器、调试器和编译器等，一般它包含在 Visual Studio 中。Visual Studio 还包含 VB、VC++、C# 等编译环境。

1.1 Visual C++ 概述

Visual C++ 是微软公司推出的基于 Win 32 的开发环境，是面向对象的可视化集成编程系统。它不但具有程序框架自动生成、灵活方便的类管理、代码编写和界面设计集成交互操作、可开发多种程序等优点，而且通过简单的设置就可使其生成的程序框架支持数据库接口、OLE2、WinSock 网络、三维控制界面。它以拥有“语法高亮”、自动编译及高级除错功能而著称，如允许用户进行远程调试、单步执行等。它还允许用户在调试期间重新编译被修改的代码，而不必重新启动正在调试的程序，其编译及建置系统以预编译头文件、最小重建功能及累加连接著称。这些特征明显缩短了程序编辑、编译及链接的时间花费，在大型软件计划上尤其显著。

Visual C++ 的特点如下。

(1) 源程序可以采用标准 C++ 语言和扩展 C++/CLI 语言编写，支持面向对象设计方法，使用功能强大的微软基础类库 (Microsoft Foundation Classes, MFC)。

(2) 开发出来的软件稳定性好、可移植性强。

(3) 可以编制各种各样 Windows 应用程序，包括对话框程序、文档/视图程序和组合界面程序。

(4) 作为 Visual Studio 可视化组件家族中最重要的成员，可与其他组件 Visual Basic、NET、Visual J#、Visual C# 及 Windows Forms 紧集成，为开发人员提供了相关的工具和框架支持，可进行不同类型和综合软件项目的开发，适用于开发专业的 Windows、Web 和企业级应用程序。

(5) 联机帮助系统功能强大。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信息，又能脱离集成开发环境而独立地运行。

1.2 Visual Studio 集成开发环境

Visual Studio 是微软公司推出的开发环境，是目前流行的 Windows 平台应用程序开发环境。从 Visual Studio 的第一个版本开始，微软公司就将提高开发人员的工作效率和灵活性作为研发目标。Visual Studio 2010 于 2010 年 4 月 12 日上市，其集成开发环境 (IDE, In-

egrated Development Environment)的界面被重新设计和组织,变得更加简单明了。Visual Studio 2010 同时带来了 .NET Framework 4.0、Microsoft Visual Studio 2010 CTP(Community Technology Preview),并且支持开发面向 Windows 7 的应用程序。除了支持 SQL Server 外,Visual Studio 2010 还支持 IBM DB2 和 Oracle 数据库。开发人员使用具有更多新特性的版本 Visual Studio 2010,即可非常有效地在新一代应用平台上为客户创造令人惊奇的解决方案。

Visual Studio 2010 的 9 个新功能如下。

- (1) C# 4.0 中的动态类型和动态编程。
- (2) 多显示器支持。
- (3) 使用 Visual Studio 2010 的特性支持测试驱动开发(TDD)。
- (4) 支持 Office。
- (5) Quick Search 特性。
- (6) C++ 0x 新特性。
- (7) IDE 增强。
- (8) 使用 Visual C++ 2010 创建 Ribbon 界面。
- (9) 新增基于 .NET 平台的语言 F#。

Visual C++ 被整合在 Visual Studio 之中,但仍可单独安装使用。目前整合了 Visual C++ 的 Visual Studio 2010 有四种版本: Visual Studio 2010 Professional、Visual Studio 2010 Premium、Visual Studio 2010 Ultimate 和 Visual Studio Test Professional 2010。

Visual Studio 2010 Professional 是供开发人员执行基本开发任务的重要工具,可简化在各种平台(SharePoint 和云)上创建、调试和开发应用程序的过程。Visual Studio 2010 Professional 自带对测试驱动开发的集成支持及调试工具,以帮助确保提供高质量的解决方案。

1.2.1 Visual Studio 2010 界面

Visual Studio 2010 是基于 Microsoft Windows 的应用程序集成开发环境,它使开发人员能够快速地创建高质量和丰富用户体验的应用程序。Visual Studio 2010 能让各种规模的组织都能快速地创建更安全、更易于管理且更可靠的应用程序。

在安装完 Visual Studio 2010 后,第一次打开使用时会出现一个对话框,在对话框中选择 Visual C++ 选项即可。Visual Studio 2010 集成开发环境由若干元素组成:菜单栏、标准工具栏,以及停靠或自动隐藏在左侧、右侧、底部和编辑器空间中的各种窗口,如图 1-1 所示。可用的工具窗口、菜单栏和工具栏取决于所处理的项目或文件类型。

1. 菜单栏

IDE 顶部的菜单栏将命令分成不同的类别。例如,“项目”菜单包含与开发人员正在处理的项目相关的命令。在“工具”菜单上,开发人员可通过选择“选项”自定义 IDE,或选择“获取工具和功能”向安装程序添加功能。

2. 工具栏

工具栏各项其实在菜单栏都有与它们对应的菜单项,功能是一样的。工具栏上有个下

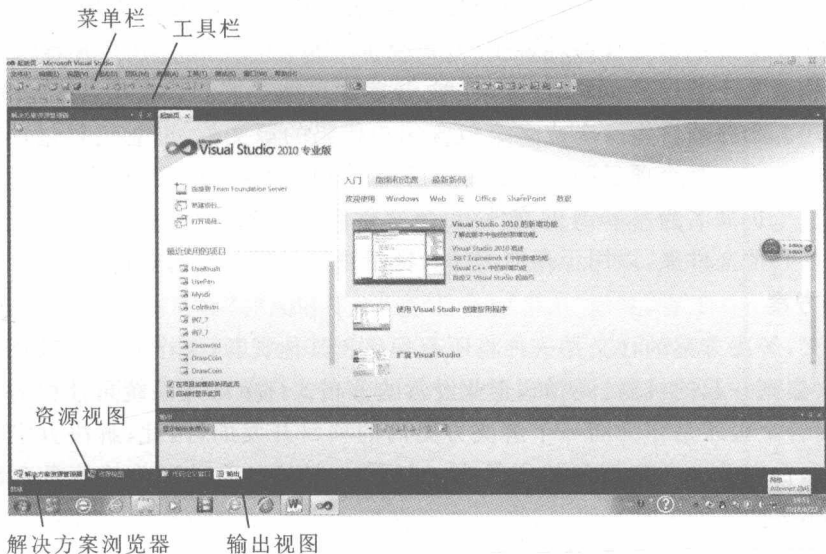


图 1-1 Visual Studio 2010 集成开发环境

列表框,它包含有 Debug、Release 和配置管理器。选择 Debug 时为调试模式,生成的可执行程序中包含调试信息,我们可以调试并清楚地看到变量值;选择 Release 时,生成的可执行程序中不含调试信息,在设置断点后,我们可能看到不准确的变量值。

3. 解决方案浏览器

在解决方案浏览器中,我们可以看到由所有头文件和源文件构成的树,如图 1-2 所示,头文件就是以.h 为扩展名的文件,源文件就是以.cpp 为扩展名的文件。一个解决方案中可以包含多个项目,可以把它理解为多个有关系或没关系的项目的集合,有时把多个项目放到一个解决方案里会使调试很方便。

4. 类视图

类视图显示每个项目的每个类,也是树形结构。

在解决方案浏览器或类视图中双击每一项,其中间区域都会打开相应的文件或到类的位置。

5. 资源视图

资源视图窗口显示当前项目中的一些资源。

6. 输出视图

输出视图,用于输出程序运行信息和一些调试信息。

1.2.2 项目与解决方案

Visual Studio 已经将解决方案的理念融入项目管理中。所以,事实上,我们在创建一个项目时,不是简单地创建一个项目,而是创建一个解决方案。这个解决方案包含项目需要的各种内

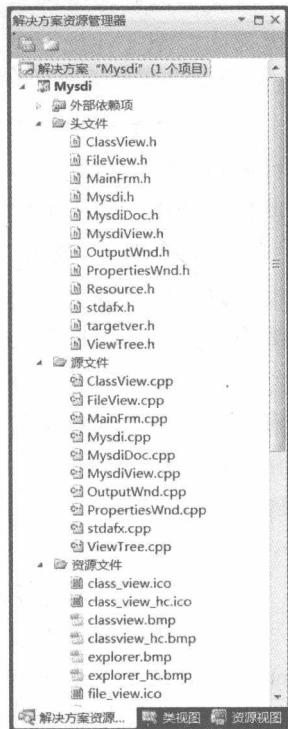


图 1-2 解决方案资源管理器

容,以及将各种内容有条不紊地组织在一起,这可提高开发效率,便于管理。

1. 项目

项目是构成某个程序的全部组件的容器,该程序可能是控制台程序、基于窗口的程序或某种别的程序。程序通常由一个或多个包含用户代码的源文件,可能还要包含其他辅助数据的文件共同组成。某个项目的所有文件都存储在相应的项目文件夹中,关于项目的详细信息存储在一个扩展名为.vcproj的文件中,该文件同样存储在相应的项目文件夹中。项目文件夹还包括其他文件夹,以用于存储编译及链接项目时所产生的输出。

2. 解决方案

顾名思义,解决方案的含义是一种将所有程序和其他资源(某个具体的数据处理问题的解决方案)聚集到一起的机制。例如,企业经营的分分布式订单录入系统可能由若干个不同的程序组成,而各个程序是作为同一个解决方案内的项目开发的,因此,解决方案就是存储与一个或多个项目有关的所有信息的文件夹。与解决方案中项目有关的信息存储在扩展名为.sln和.suo的两个文件中。

我们在 Visual Studio 中新建一个项目时,就隐含地创建了一个解决方案。这样方便以后还可以在解决方案中添加新项目,或加入以后的项目。不过为了管理方便,请把相关的项目放在一个解决方案中,不相关的项目不要放在一个解决方案中。在新建项目时,默认打开了项目管理器。如果在界面中没找到解决方案管理器,那么单击 Visual Studio 主菜单的“视图”→“解决方案资源管理器”选项就可打开它。

解决方案资源管理器是一个视图窗口,从本质上说是一个可视化的文档管理系统,选取其内部的一个文件就可显示该文件,在更改了界面后,会将更改存入内部文件,我们可以把它看成是整个项目的大管家,如图 1-2 所示。

在解决方案资源管理器中,有生成解决方案、清理解决方案、添加项目、项目生成顺序、项目依赖和设为启动项目、在资源管理器打开文件夹、属性等快捷菜单。这一级的菜单用于对项目的管理。解决方案以树状结构的形式管理项目,每一个项目都是根节点,多个项目并列。单击每个节点,就可以看到项目的所有文件。项目文件按照项目的类型进行分类。我们也可以自定义分类分组、添加文件、添加类等。例如,想要删除一个文件,只需在解决方案资源管理器中选中这个文件,然后按“Delete”键即可。

Visual Studio 2010 创建项目时,默认的项目文件夹的名称与项目名称相同,该文件夹还将容纳构成该项目定义的文件。如果不修改的话,那么解决方案文件夹具有与项目文件夹相同的名称,其中包含项目文件夹、定义解决方案内容的文件。解决方案文件夹内包含如下四个文件。

(1) 扩展名为.sln的文件,它记录关于解决方案中项目的信息。

(2) 扩展名为.suo的文件,它记录应用于该解决方案的用户选项。

(3) 扩展名为.sdf的文件,它记录与解决方案的 Intellisense 工具有关的数据。Intellisense 工具是在 Editor 窗口输入代码时提供自动完成和提示功能的工具。

(4) 扩展名为.opensdf的文件,它记录关于项目状态的信息,此文件只在项目处于打开状态时才有。

3. 控件工具箱

控件工具箱属于 Visual Studio 的一大特色,它为我们的开发提供许多有用的控件。在 Web 项目的开发中,利用工具箱可以不需要编写任何代码,只使用鼠标“拖曳”的操作方式就能够完成 Web 表单的界面设计,并且这些控件都是跨浏览器和跨设备运行的,如图 1-3 所示。

控件工具箱的内容依赖于当前正在使用的设计器,也同样依赖于当前的项目类型。可以自定义控件工具箱的标签及标签内的项。右击标签顶部,选择“Rename Tab”、“Add Tab”或“Delete Tab”标签,在工具箱的空白处右击,在快捷菜单中单击“Choose Item”选项,就可以添加一个或多个项。同时,还可以把一个项从一个标签拖放到另一个标签内。

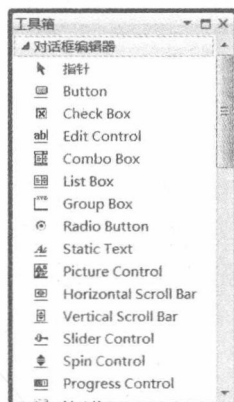


图 1-3 控件工具箱

4. 错误列表

错误列表通过检测有问题的代码而产生错误信息,如当我们写的语句有语法错误时,错误信息会列在错误列表中。错误列表的每一项都由一个文本描述和一个链接组成,这个链接能帮助我们找到项目里面出错程序代码的指定行。作为 Visual Studio 2010 的默认设置,当生成一个有错误的项目时,错误列表会自动出现,也可单击“视图”→“错误列表”选项或按“Ctrl+E”键,打开“错误列表”,如图 1-4 所示。

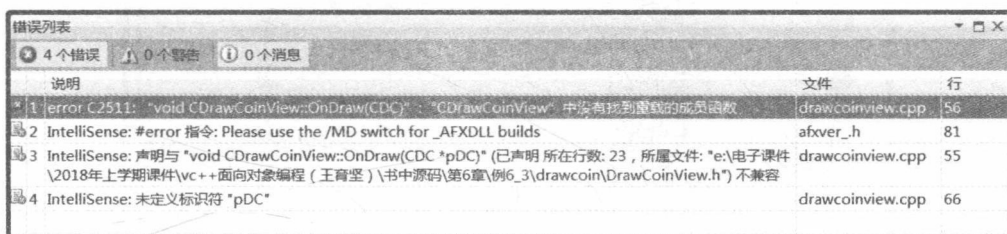


图 1-4 错误列表

由图 1-4 可知,错误列表有“错误”、“警告”和“消息”三个选项卡,以查看不同级别的信息。其中,“错误”表示程序的一些严重性比较高的错误,如果不修改这些错误,那么程序将无法编译成功;“警告”表示软性错误,也可以看成是潜在性错误,如定义了程序里面没有用到的变量、在页面设计时用了不符合标准的 HTML 标签等,这种错误不影响程序的编译,但会带来潜在的错误。

若要对列表进行排序,请单击任意列标头。若要按其他列对列表进行进一步排序,请按“Shift”键并单击其他列标头。若要选择哪些列显示、哪些列隐藏,请从快捷菜单中单击“显示列”选项。若要更改列的显示顺序,请任意向左或向右拖曳列标头。

1.3 创建应用程序

Visual C++ 不仅是一个 C++ 语言编译器,还是一个基于 Windows 操作系统的可视化集成开发环境。Visual C++ 由许多组件组成,包括编辑器、编译器、调试器及应用程序向

导(AppWizard)、类向导(ClassWizard)等开发工具。这些组件通过一个名为 Developer Studio 的组件集成为一个和谐的开发环境。

1.3.1 创建 Win32 控制台应用程序

Win32 控制台应用程序只是在控制台运行的程序,没有界面,只有命令符,生成的“.exe”文件直接运行即可。Win32 控制台程序初始代码模板以函数 `main()` 为程序入口,默认情况下,只链接 C++ 语言程序运行时库和一些核心的 Win32 库,所以编译出来的程序会有黑色的控制台窗口作为呈现标准输入(stdin)和标准输出(stdout)。

创建一个应用程序,首先要创建一个项目。项目用于管理组成应用程序的所有元素,并由它生成应用程序。

例 1-1 用 Visual C++ 编写一个控制台的“HelloWorld”程序。

(1) 启动 Visual Studio 2010 开发环境,单击菜单“开始”→“文件”→“新建”→“项目”选项,弹出“新建项目”对话框,如图 1-5 所示。展开“Visual C++”树形节点,单击“Win32”→“Win32 控制台应用程序”,在“名称”栏中输入项目名称,“例 1-1”,在“位置”栏中指定项目存储路径,单击“确定”按钮。

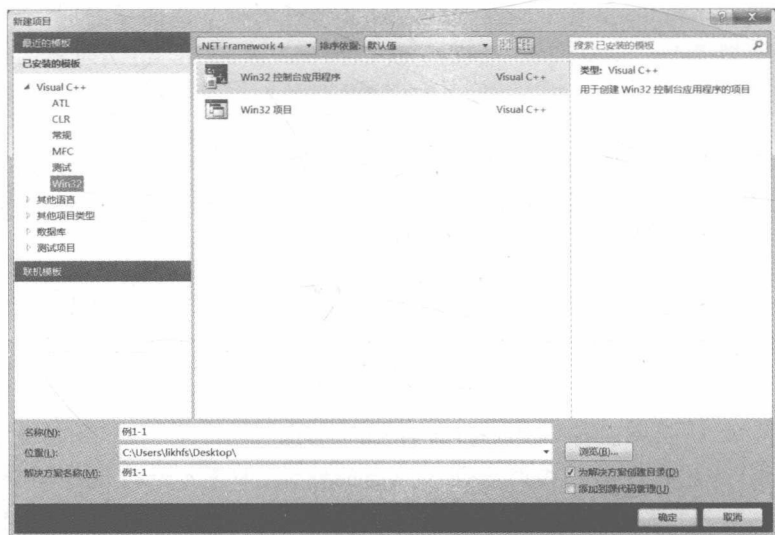


图 1-5 新建 Win32 控制台应用程序项目

(2) 弹出“Win32 应用程序向导”对话框,单击“下一步”按钮或“应用程序设置”标签,如图 1-6 所示。

(3) 选择“控制台应用程序”单选框和“空项目”复选框,如图 1-7 所示,然后单击“完成”按钮。

(4) 单击“调试”→“开始执行”选项,如图 1-8 所示。这时会遇到编译错误,为什么呢? 因为还没有函数 `main()`,对于一个 C++ 语言项目来说,一定要有一个且仅有一个函数 `main()` (Windows 程序需要函数 `WinMain()`),该函数可以是隐式提供的,也可以是显式提供的。