



21世纪高等学校电子信息类专业规划教材

Visual C++ 程序设计案例教程

主编 申闫春



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社

<http://press.bjtu.edu.cn>

21 世纪高等学校电子信息类专业规划教材

Visual C++ 程序设计案例教程

申闫春 主 编

清华大学出版社
北京交通大学出版社

· 北京 ·

内 容 简 介

本书以使初学者能熟练掌握 Visual C++ 应用程序开发为目的,以浅显易懂的语言和基本实例引领读者迈上编程殿堂。

全书共 22 章:前 3 章作为基础章必不可少,重点复习了 C 语言知识,讲述 C++ 语言在 C 上的扩充及面向对象的概念,教授了 Win32 API 函数的简单使用,详解演示如何通过 MSDN 学习 API 函数;第 4~10 章作为 MFC 的进阶章节,让读者全面掌握 Windows 窗口程序的建立、窗口绘图、对话框、键盘和鼠标及计时器事件的响应、菜单和工具栏的建立、文档/视图和框架的相互关系,使读者能从大局上把握 MFC 的必备知识;第 11~22 章为高级应用主题,以完整编程代码,讲解位图图像、控制打印、封装 ActiveX 控件、封装 DLL、进程和多线程、修改注册表、获取系统信息、网络通信、数据库、水晶报表、Office 和多媒体等编程技术。

本书适合于 Visual C++ 的初学者,特别是大中专院校刚学完 C 语言的学生,有 C++ 面向对象知识则更佳。

本书提供配套的案例源代码,所有源代码均在 Visual C++ 2005 下调试通过。读者可通过出版社网站免费下载,网址为 <http://press.bjtu.edu.cn>。

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 13501256678 13801310933

图书在版编目(CIP)数据

Visual C++ 程序设计案例教程/申闰春主编. —北京:清华大学出版社;北京交通大学出版社, 2009. 12

(21 世纪高等学校电子信息类专业规划教材)

ISBN 978-7-81123-961-4

I. ①V… II. ①申… III. ①C 语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2009)第 214623 号

责任编辑:郭东青

出版发行:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>
北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京瑞达方舟印务有限公司

经 销:全国新华书店

开 本:185×260 印张:23.75 字数:593 千字

版 次:2010 年 4 月第 1 版 2010 年 4 月第 1 次印刷

书 号:ISBN 978-7-81123-961-4/TP·559

印 数:1~4000 册 定价:36.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。
投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

前 言

长久以来，在编程界流传着一个说法，即：“聪明的程序员用 Delphi，真正的程序员用 Visual C++”。从中可以看出，Visual C++ 就像“独孤九剑”，强大但难以掌握，很多怀着编程爱好的人，在 Visual C++ 的高门槛前无法跨越，抱憾而退，以至于有“学 Visual C 有捷径吗？如果有的话，也只是态度、方法和汗水！”的感叹，让不少初学者望而却步。

在互联网已经普及的今天，各种高级开发语言日臻完善，相应工具简单易用，如 C# 和 Java 等，以其高效快速开发，适合浏览器方式吸引了不少程序员，那么，Visual C++ 还有用处吗？

类似的问题在网上争论不休，然而，不可否认的是，微软的 Windows 操作系统在桌面应用中仍占据霸主地位，Visual C++ 作为 Visual Studio 的主要集成开发工具之一，有着方便实现底层编程的强大功能，其应用范围还在不断扩展，如：安全、游戏、外挂、木马、病毒、远程控制、视频、3G 智能手机、驱动程序开发等，它仍是程序员的神兵利器。

笔者作为计算机教育战线的资深人员，深明技术学习无指导之苦，从对读者负责角度出发，特推出“买教程送培训”活动，聘请专业开发人员在网指导，凡购买本书并获得开心学网站 www.kaixinx.com 会员资格的读者，可加入“开心学 Visual C++ 群 (103756638)”，接受软件工程师的答疑解惑。

编 者

2010 年 3 月 10 日

目 录

第1章 第一个C语言程序	1
1.1 C语言的背景与历史	1
1.2 C语言特点	1
1.3 使用 Visual C++ 2005 编写第一个C语言程序	2
1.4 C语言的基本概念	4
1.4.1 关键字	4
1.4.2 控制语句	5
1.4.3 变量	5
1.4.4 常量	6
1.4.5 函数	8
1.4.6 素数的判断	9
1.5 C语言的指针、数组、结构、联合与枚举	10
1.5.1 指针	10
1.5.2 数组	12
1.5.3 结构	14
1.5.4 联合	15
1.5.5 枚举	17
1.6 总结：C语言的编程风格与学习方法	19
习题1	20
第2章 第一个C++程序	21
2.1 C++概述	21
2.2 C++发展历史	21
2.3 C和C++的关系	21
2.4 第一个C++程序	22
2.5 C++对C的扩充	24
2.5.1 C++的输入输出	24
2.5.2 字符串变量	25
2.5.3 有默认参数的函数	25
2.5.4 动态分配/撤销内存的运算符 new 和 delete	26
2.5.5 内联函数	27
2.5.6 函数的重载	27
2.5.7 函数模板	28
2.6 C++中的面向对象机制	30



2.6.1 类与结构	30
2.6.2 类	31
2.6.3 类的构造函数与析构函数	33
2.7 改造 monitor 类	36
2.8 C++ 小结	38
习题 2	39
第 3 章 创建 Win32 API 应用程序	40
3.1 Windows 编程基础	40
3.1.1 什么是 Win32	40
3.1.2 什么是 Windows API 函数和 Win32 API	40
3.1.3 什么是 SDK	40
3.1.4 窗口与句柄	41
3.1.5 ANSI 编码与 Unicode 编码	41
3.2 利用 Windows API 函数编写一个消息框	41
3.3 消息框的实现详解	42
3.3.1 Windows.h 头文件	42
3.3.2 域操作符::	43
3.3.3 MessageBox 函数	44
3.3.4 通过 MSDN 学习 API 函数	44
3.3.5 常用的 Windows 数据类型	45
3.3.6 API 函数参数的选择	46
习题 3	48
第 4 章 编写第一个 MFC 窗口程序	49
4.1 窗口	49
4.1.1 三种窗口类型	49
4.1.2 客户区和非客户区	49
4.1.3 什么是 MFC	50
4.2 编写第一个 MFC 窗口程序	50
4.2.1 建立一个单文档的程序	50
4.2.2 MFC AppWizard 产生的程序分析	53
4.2.3 MFC 的基类 CObject	54
4.2.4 字符串类 CString	55
4.3 利用设备上下文实现显示输出	58
4.3.1 什么是设备上下文	58
4.3.2 CDC 类与资源泄漏	58
4.4 Windows 消息机制	60
4.4.1 消息的组成	60
4.4.2 消息的处理	60
4.4.3 消息传递机制	61
4.4.4 编写响应鼠标单击的窗口程序	61
4.4.5 消息映射机制的实现	63

4.4.6 客户区的重绘	64
4.5 Win32 函数和 MFC 函数的差别	65
习题 4	66
第 5 章 窗口绘图	67
5.1 绘图相关知识与应用	67
5.1.1 Windows 窗口绘图的坐标系统	67
5.1.2 与绘图相关的几个几何对象的类	67
5.1.3 创建一个绘图程序	68
5.2 Windows 绘图的类与函数	81
5.2.1 CDC 类	81
5.2.2 CPen 类	83
5.2.3 CBrush 类	84
5.2.4 绘图模式	85
5.2.5 Invalidate 函数	86
习题 5	86
第 6 章 对话框	88
6.1 什么是对话框	88
6.2 对话框的分类	88
6.3 如何设计对话框	88
6.3.1 设计对话框模版	89
6.3.2 对话框类的设计	90
6.4 对话框的数据交换机制 (DDX)	96
6.5 模态对话框和非模态对话框	96
6.6 对话框实例	97
习题 6	102
第 7 章 鼠标键盘定时器	103
7.1 鼠标键盘定时器的相关事件及编程例子	103
7.1.1 如何对鼠标事件编程	103
7.1.2 如何对键盘事件编程	103
7.1.3 如何对定时器编程	103
7.1.4 创建鼠标键盘定时器应用程序	104
7.2 键盘鼠标的事件及编码	113
7.2.1 Windows 中的键盘事件	113
7.2.2 扫描码和虚拟键码	113
7.2.3 鼠标事件	115
7.3 定时器的使用	116
7.3.1 如何创建定时器	116
7.3.2 取消定时器	118
习题 7	118



第8章 菜单、工具栏、快捷键	119
8.1 菜单	119
8.1.1 什么是菜单	119
8.1.2 建立菜单	119
8.2 工具栏	128
8.2.1 工具栏的作用	128
8.2.2 建立工具栏	128
8.3 快捷键	129
8.3.1 什么是快捷键	129
8.3.2 添加快捷键	130
8.4 菜单、工具栏和快捷键相关知识	132
8.4.1 菜单的消息与函数	132
8.4.2 工具栏的属性与函数	136
8.4.3 快捷键的相关属性	136
习题8	138
第9章 文档、视图、框架窗口	139
9.1 文档—视图结构	139
9.1.1 什么是文档—视图模型	139
9.1.2 什么是SDI和MDI	139
9.1.3 什么是文档模板	140
9.2 文档与视图的联系	140
9.3 文档/视图架构中消息流动机制	143
9.4 持久性和序列化	146
9.4.1 什么是持久性	146
9.4.2 什么是序列化	146
9.4.3 文档的存盘和读取	146
9.4.4 OnDraw (CDC*/* pDC*/) 简介	151
9.4.5 保存时自定义后缀的实现	151
9.4.6 使类具备序列化功能	152
习题9	153
第10章 控件	154
10.1 控件的分类	154
10.2 控件的共性	154
10.2.1 所有的控件都是子窗口	154
10.2.2 控件的创建方法	155
10.2.3 手工创建控件	155
10.2.4 控件的访问方法	157
10.2.5 控件及控件对象的删除	159
10.3 MFC通用控件的实例	159
10.3.1 列表控件 (CListCtrl)	159
10.3.2 树控件 (CTreeCtrl)	161

10.4 列表控件和树控件举例	163
10.5 设计新的控件类	166
习题 10	166
第 11 章 位图	167
11.1 位图编程基础	167
11.1.1 什么是位图	167
11.1.2 调色板	168
11.1.3 调色板管理器机制	168
11.1.4 DDB 和 DIB	169
11.1.5 位图文件结构	169
11.2 将客户区图像保存为位图文件	171
11.2.1 将位图显示在客户区中	171
11.2.2 CreateCompatibleDC 函数与 BitBlt 函数	173
11.2.3 将位图显示在客户区中	174
11.3 软件背景图和空心字的显示	180
11.3.1 软件背景图的拉伸显示	180
11.3.2 StretchBlt 和 SetStretchBltMode 函数	181
11.3.3 空心字的显示	182
11.3.4 路径函数 BeginPath 和 EndPath	184
习题 11	185
第 12 章 打印	186
12.1 打印编程基础	186
12.1.1 显示分辨率	186
12.1.2 打印分辨率	186
12.1.3 映射模式	187
12.1.4 视口和窗口	187
12.1.5 需要重载的 CView 类成员函数	188
12.2 编写一个所见即所得的程序	189
12.3 打印详解	193
12.3.1 设置打印机状态	193
12.3.2 通过 CView 打印	193
12.4 编写一个基于对话框的打印程序	196
12.4.1 打印的实现	196
12.4.2 打印对话框	199
12.4.3 DrawText 函数	199
12.4.4 StartDoc 函数和 EndDoc 函数	200
12.4.5 inflateRect 函数和 deflateRect 函数	201
习题 12	201
第 13 章 ActiveX 控件	202
13.1 ActiveX 简介与开发	202



13.1.1	ActiveX 简介	202
13.1.2	创建一个 ActiveX 控件	203
13.1.3	Visual C++ 2005 中使用 ActiveX 控件	215
13.2	ActiveX 控件详解	220
13.2.1	不需要注册 ActiveX 控件	220
13.2.2	ActiveX 控件测试容器	220
13.3.3	MFC ActiveX 的属性、方法、事件	220
习题 13		221
第 14 章	动态链接库	222
14.1	动态链接库基础	222
14.1.1	什么是动态链接库	222
14.1.2	静态链接和动态链接	222
14.1.3	使用动态链接库的好处	223
14.1.4	应用程序和动态链接库的区别	223
14.2	动态链接库的封装和加载	223
14.2.1	动态链接库的封装	223
14.2.2	Dumpbin 工具查看 DLL 的导出函数	224
14.2.3	从 DLL 导出函数	226
14.2.4	动态链接库的隐式加载	227
14.2.5	动态链接库的显式加载	229
14.2.6	易犯的名字改编错误	230
14.2.7	什么是 DLL 劫持	231
14.3	MFC DLL 简介	231
习题 14		233
第 15 章	进程、线程	234
15.1	进程与线程	234
15.1.1	什么是进程与线程	234
15.1.2	为什么要使用多线程	234
15.1.3	如何创建和结束进程	235
15.1.4	如何创建和结束线程	235
15.1.5	进程线程实例	236
15.2	进程与线程相关方法	245
15.2.1	如何使用 CreateProcess() 创建进程	245
15.2.2	如何使用 ExitProcess() 结束进程	246
15.2.3	如何使用 TerminateProcess() 结束进程	246
15.2.4	如何使用 AfxBeginThread() 创建线程	247
15.2.5	如何使用 AfxEndThread() 结束线程	248
15.2.6	如何使用 TerminateThread() 结束线程	248
习题 15		248

第 16 章 注册表	249
16.1 注册表知识	249
16.1.1 什么是注册表	249
16.1.2 什么是 INI 文件	249
16.1.3 注册表的层次结构	250
16.2 编写程序启动自运行实例	251
16.3 编程操作注册表	253
16.3.1 注册表启动项	253
16.3.2 操作注册表的 API 函数	254
16.3.3 路径分解函数_wsplitpath	255
16.4 宽字符编程中易犯的错误	255
习题 16	258
第 17 章 系统管理	259
17.1 系统信息的获取与显示	259
17.1.1 如何获取各种系统信息	259
17.1.2 如何获取系统进程信息	259
17.1.3 如何获取内存使用情况	259
17.1.4 如何获取系统驱动器信息	259
17.1.5 如何获取系统设备信息	260
17.1.6 创建系统管理应用程序	260
17.2 系统信息相关函数的使用	276
17.2.1 进程相关的几个函数是如何使用的	276
17.2.2 如何获取系统物理内存与虚拟内存的使用情况	279
17.2.3 获取系统驱动器信息的几个函数是如何使用的	279
17.2.4 获取系统设备的几个函数是如何使用的	281
习题 17	283
第 18 章 网络通信	284
18.1 socket 编程	284
18.1.1 什么是 socket	284
18.1.2 socket 编程步骤	284
18.1.3 创建聊天系统的服务器端	285
18.1.4 创建聊天程序的客户端	296
18.2 网络编程中 socket 的使用	309
18.2.1 如何使用 socket 相关的函数	309
18.2.2 SOCK_DGRAM 类型的套接字如何使用	312
18.2.3 为什么本章例子程序中多次使用线程	312
习题 18	312
第 19 章 数据库编程	313
19.1 ADO 简介	313



19.1.1	ODBC 与 ADO	313
19.1.2	使用 ADO 的基本流程	313
19.1.3	如何引入和初始化 ADO 的 COM 库	313
19.1.4	ADO 提供了什么接口	314
19.1.5	使用 ADO 操作数据库	315
19.2	ADO 对象与方法详解	332
19.2.1	关于数据类型转换 (_bstr_t 和 CString)	332
19.2.2	使用 ADO 连接数据库	332
19.2.3	执行不返回结果的 SQL 命令	333
19.2.4	执行 SQL 命令获取记录集	334
习题 19		335
第 20 章	水晶报表	337
20.1	为何使用报表工具	337
20.2	什么是水晶报表	337
20.2.1	安装水晶报表	338
20.2.2	Crystal Reports for Visual Studio 2005	338
20.3	水晶报表使用实例	338
20.4	在 MFC 中使用水晶报表	343
习题 20		344
第 21 章	Office 编程	345
21.1	如何进行 Office 编程	345
21.1.1	Office 的基本编程模型——对象	345
21.1.2	Excel 编程模型中的对象	345
21.1.3	在 Visual C++ 2005 中操作 Excel 对象	346
21.1.4	Excel 编程实例	347
21.2	Excel 编程中方法与参数的详解	356
21.2.1	打开一个 Workbook 的 open 方法	356
21.2.2	保存 Excel 文档的 SaveAs 方法	358
21.2.3	设置字体或单元格等颜色的 ColorIndex 参数	359
习题 21		360
第 22 章	多媒体编程	361
22.1	Windows 平台下的多媒体编程	361
22.2	Windows 下的屏幕保护程序	361
22.3	制作一个简单的屏幕保护程序	362
22.4	屏保程序的调用机制	365
习题 22		366

第 1 章 第一个 C 语言程序

C 语言是国际上广泛流行的一种结构化程序设计语言，既可用来开发系统软件，也可用来开发应用软件。

1.1 C 语言的背景与历史

C 语言是国际上广泛流行的高级语言。C 语言是在 B 语言的基础上发展起来的。B (BCPL)语言是 1970 年由美国贝尔实验室设计的，并用于编写了第一个 UNIX 操作系统，在 PDP7 上实现。优点是精练、接近硬件，缺点则过于简单、数据无类型。

1973 年贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言，对 B 语言取长补短，并用之改写了原来用汇编编写的 UNIX (即 UNIX 第 5 版)，但仅在贝尔实验室使用。

1975 年 UNIX 第 6 版发布，C 语言优点突出引起关注。

1977 年出现了《可移植 C 语言编译程序》，推动了 UNIX 在各种机器上实现，C 语言也得到推广，与其发展相辅相成。

1978 年影响深远的名著 *The C Programming Language* 由 Brian W. Kernighan 和 Dennis M. Ritchie 合著，被称为标准 C 语言。

之后，C 语言先后移植到大、中、小、微型计算机上，已独立于 UNIX 和 PDP，风靡世界，成为使用最广泛的几种计算机语言之一。

1983 年，美国国家标准化协会 (ANSI) 根据 C 语言各种版本对 C 的发展和扩充，制定了新的标准 ANSI C，比标准 C 有了很大的发展。

1987 年，ANSI 公布了新标准——87 ANSI C。

1988 年 K&R 按照 ANSI C 修改了他们的 *The C Programming Language*。

1990 年，国际标准化组织接受了 87 ANSI C 为 ISO C 的标准 (ISO 9899—1990)。

1994 年，ISO 又修订了 C 语言标准。

目前流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的。

1.2 C 语言特点

C 语言具有以下特点。

(1) 语言简洁、紧凑，使用方便、灵活。32 个关键字、9 种控制语句，程序形式自由。



- (2) 运算符丰富，34 种运算符。
- (3) 数据类型丰富，具有现代语言的各种数据结构。
- (4) 具有结构化的控制语句，是完全模块化和结构化的语言。
- (5) 语法限制不太严格，程序设计自由度大。
- (6) 允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可直接对硬件进行操作，兼有高级和低级语言的特点。
- (7) 目标代码质量高，程序执行效率高。只比汇编程序生成的目标代码效率低 10%—20%。
- (8) 程序可移植性好（与汇编语言相比），基本上不做修改就能用于各种型号的计算机和各种操作系统。

1.3 使用 Visual C++ 2005 编写第一个 C 语言程序

- (1) 启动 Microsoft Visual Studio 2005。
- (2) 新建项目。

执行“文件”|“新建”|“项目”（或在起始页上单击创建：项目）|“Visual C++”|“Win32 控制台应用程序”命令，在名称里输入“Ex1_1”，在“位置”里输入“D:\”。如图 1-1 所示。

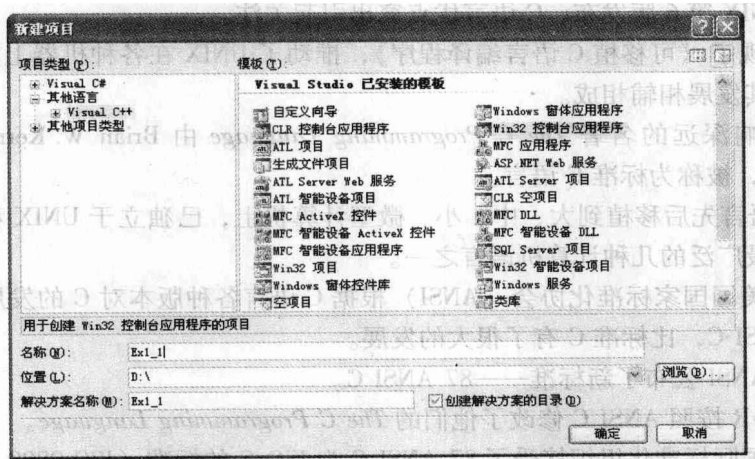


图 1-1 新建控制台应用程序

单击“确定”按钮，进入 Win32 应用程序向导，可以选择直接单击“完成”按钮，但这里选择单击“下一步”按钮，进入应用程序设置。如图 1-2 所示。

如图 1-3 所示，在应用程序设置选项中，默认情况下，选中控制台应用程序，单击“完成”按钮，进入 Ex1_1 的代码编辑区域。

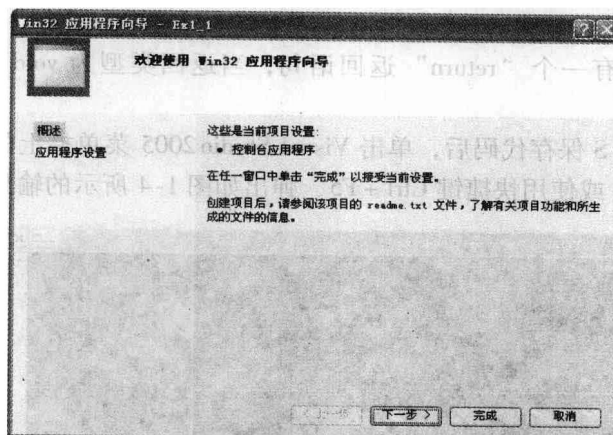


图 1-2 Win32 应用程序向导

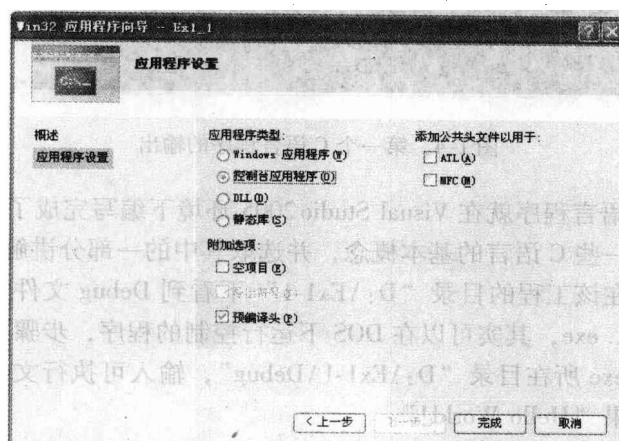


图 1-3 应用程序设置

在代码区域输入以下代码：

```
#include "stdafx.h"
#include "stdio.h"           /* 文件包含 */
int main()                  /* 主函数 */
{                            /* 函数体开始 */
    printf("Hello,World! \n"); /* 输出语句 */
    return 0;               /* 函数返回语句 */
}
```

可以看到，一个完整的C语言程序，至少应该包含以下部分。

- 使用标准库函数时应程序开头一行写：`#include "stdio.h"`
- `main`：主函数名。每个C程序必须有一个主函数 `main`。
- `int`：函数返回类型。
- `{ }`：函数开始和结束的标志，不可省略。



- 每个 C 语句以分号结束。
- 每个函数通常有一个“return”返回语句，当返回类型为 void 时，可以省略 return 语句。

使用快捷键 Ctrl + S 保存代码后，单击 Visual Studio 2005 菜单栏上的“调试”|“开始执行（不调试）”命令，或使用快捷键 Ctrl + F5，弹出如图 1-4 所示的输出窗口。

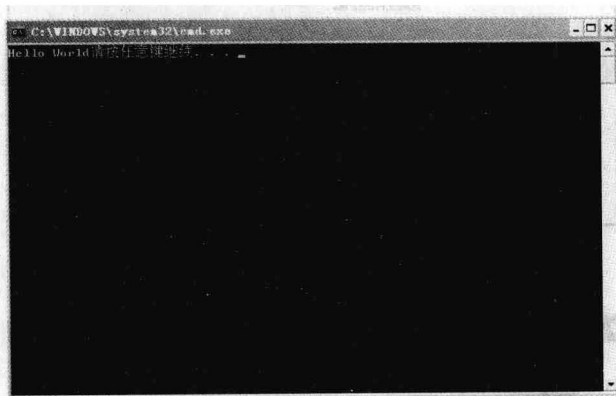


图 1-4 第一个 C 语言程序的输出

这样，第一个 C 语言程序就在 Visual Studio 2005 环境下编写完成了，它只有一个简单的输出语句，下面介绍一些 C 语言的基本概念，并选取其中的一部分讲解。

在调试后，可以在该工程的目录“D:\Ex1-1”下看到 Debug 文件夹，在此文件夹下有一个可执行文件 Ex1-1.exe，其实可以在 DOS 下运行控制的程序，步骤是进入 DOS 状态，进入可执行文件 Ex1-1.exe 所在目录“D:\Ex1-1\Debug”，输入可执行文件名“Ex1-1”，就可以看到程序的输出结果“Hello World!”。

1.4 C 语言的基本概念

1.4.1 关键字

下面列出 ANSI 标准定义的 32 个 C 语言的关键字，这些关键字在后面的学习中基本上都会用到。这些关键字中，有关于变量类型的，如 int、float、double、char、long 等，也有关于程序控制语句的，如 break、continue、for、while 等。关于这些关键字详细的信息，读者可以查询 ANSI 的标准，在后面的章节中也会选择部分作讲解。

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	sizeof	static	struct	switch	typedef	union
unsigned	void	volatile	while			

1.4.2 控制语句

C语言的控制语句一共有以下9种。

- * goto 语句：无条件转向。
- * if 语句：判断语句。
- * while 语句：循环语句。
- * do-while 语句：先执行循环体，然后判断循环条件是否成立，之后继续循环。
- * for 语句：循环语句，可替代 while 语句；只是用法不同。
- * break 语句：跳出本层的循环（只调出包含此语句的循环）。
- * continue 语句：继续。一般放到循环语句里，使程序不再执行它下面的语句，直接跳到判断语句，例如，for 语句，就直接跳到第二个分号处，while 语句，就直接跳到 while() 的括号里。
- * switch 语句：多项选择。
- * return 语句：返回。

这些控制语句在后面的章节中也会加以讲解。

1.4.3 变量

1. 什么是变量

变量是存储数据的值的空间。由于数值的类型有多种，有整数、浮点数、字符等，那么对应的变量就有整型变量、浮点型变量、字符型变量等。变量还有更细的分类，如整型变量还可以分为无符号整型、长整型和短整型。浮点型也可分为单精度型、双精度型和长双精度型。此外根据变量的修饰词，如 static、extern、register、auto 等，还可以将变量分为静态变量、外部变量、寄存器变量和自动存储变量。

变量被声明后，其名字就叫做标识符。标识符的命名是有一定规则的，介绍如下。

- (1) 标识符只能由字母、数字和下画线三类字符组成。
- (2) 第一个字符必须是字母（第一个字符也可以是下画线，但被视为系统自定义的标识符）。
- (3) 大小写敏感，如 A 和 a 是两个不同的标识符。
- (4) 标识符不能是 C 的关键字，在给变量命名时要避开这些关键字。

所有变量在使用前都必须加以声明。变量的声明语句由数据类型和其后的一个或多个变量名组成。变量声明的形式如下：

类型 <变量表>;

其中，变量表是一个或多个标识符名，每个标识符之间用“,”分隔。

2. 整型变量

整型变量是用来存储整数的。

整型变量又可分为几种，最基本的整型变量是用类型说明符 int 声明的符号整型，形式如下：