



“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定



高等院校教学改革创新示范教材 软件开发系列

Visual C++ 实用教程（第5版）

◎ 丁有和 主编



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

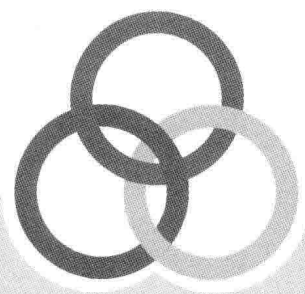


“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定

高等职业院校教学改革创新示范教材·软件开发系列

Visual C++ 实用教程 第5版

主 编 丁有和



電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是根据高等职业教育的特点, 兼顾 C++ 等级考试内容, 以应用为目的, 以必需够用为度, 以方便教和学为宗旨而编写。第 1~8 章为 C++ 内容, 兼顾 C++ 等级考试。第 9~14 章为 Visual C++ 内容, 包括 MFC 应用程序建立、窗口和对话框、常用控件、基本界面元素、数据文档和视图、图形和数据库应用等。每章以“实际需要应用”为主线, 内容之后通常都是实例, 一般先提出为什么, 再提做什么, 然后给出示例演示怎么做, 最后包括“常见问题解答”、“实验实训”以及“思考和练习”等内容。综合应用通过附录中的两个大作业(学生成绩管理(C++版)和 MFC)实现, 用于比较和教学。

本书可作为高等职业教育相关课程教材, 也可供广大 Visual C++ 应用开发人员参考。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

Visual C++ 实用教程 / 丁有和主编. —5 版. —北京: 电子工业出版社, 2014.11
“十二五”职业教育国家规划教材

ISBN 978-7-121-23926-7

I. ①V… II. ①丁… III. ①C 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2014) 第 172859 号

策划编辑: 程超群

责任编辑: 郝黎明

印 刷: 北京京师印务有限公司

装 订: 北京京师印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 19.75 字数: 506 千字

版 次: 2000 年 8 月第 1 版

2014 年 11 月第 5 版

印 次: 2014 年 11 月第 1 次印刷

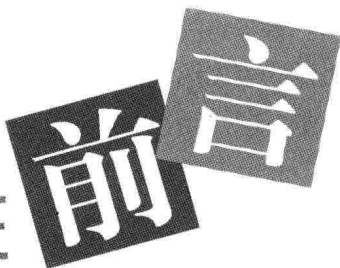
定 价: 39.90 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

PREFACE



Visual C++ (简称 VC) 是 Microsoft 公司推出的目前使用极为广泛的基于 Windows 平台的 C++ 可视化开发环境, 我国高等院校的计算机专业和有些非计算机专业已开设 C++ 和 VC 应用程序设计课程。为了方便教学, 2000 年, 我们兼顾本专科教学需要, 编写 Visual C++ 实用教程。出版后, 得到高校教师、学生和读者的广泛认同。2003~2012 年, 我们相继推出了 Visual C++ 实用教程的第 2 版、第 3 版和第 4 版, 教程和实验的内容和分工进一步优化, 一直保持着市场较高的认同度。先后被评为江苏省优秀教学成果二等奖, 普通高等教育“十一五”国家级规划教材。

本书在“Visual C++ 实用教程”的基础上, 专门根据高等职业教育和应用型本科的特点, 兼顾 C++ 等级考试内容, 以应用为目的, 以必需够用为度, 以方便教和学为宗旨而编写。具体包含了以下几个方面:

(1) 每章以“以实际需要应用”为主线, 内容之后通常都是实例, 对于较晦涩的理论, 一般先提出为什么, 再提做什么, 然后给出示例告诉怎么做! 最后包括“常见问题解答”、“实验实训”和“思考和练习”等内容。

(2) 第 1~8 章为 C++ 内容, 为了兼顾 C++ 等级考试, 内容做了一些优化, 对于一些难点则是通过“常见问题解答”来解决。实际上, 如果没有 C 语言的基础, 仍然可以使用本书。因为本书在 C++ 这部分章节中还大量介绍了结构化程序设计的算法, 包括排序和链表。由于本书重在基础, 以必需够用为度, 所以对于 C++ 模板、异常处理机制等内容并没有涉及。

(3) 第 9~14 章是 Visual C++ 的内容, 包括 MFC 应用程序建立、窗口和对话框、常用控件、基本界面元素、数据文档和视图、图形和数据库应用等。MFC 的推出实际上是减轻 SDK 编程的代码量和烦冗的 Windows 程序设计, 所以本书对于较底层控制的编程没有介绍, 转而专注于界面设计的操作层面。这样一来, 对于初学者, 更容易入门且能掌握基本技能。

总之, 本书不仅适合于教学, 也非常适合于用 Visual C++ 编程和开发应用程序的用户学习和参考。只要阅读本书, 结合操作示例进行练习, 就能在较短的时间内基本掌握 Visual C++ 及其应用技术。

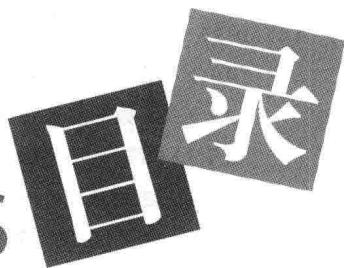
本书由南京师范大学丁有和老师担任主编。参加本书编写的还有徐文胜、刘启芬、殷红先、曹弋、陈瀚、陈冬霞、邓拼搏、高茜、刘博宇、彭作民、钱晓军、孙德荣、陶卫冬、吴明祥、王志瑞、徐斌、俞琰、严大牛、郑进、张为民、周何骏、周怡君、于金彬、马骏、周怡明、顾韵华、姜乃松、梁敬东等。此外，还有许多同志对本书的编写提供了很多帮助，在此一并表示感谢！

由于作者水平有限，不当之处在所难免，恳请读者批评指正。本书可免费下载教学课件、实例文件、综合应用实习源文件，网址为 <http://www.hxedu.com.cn>。

意见建议邮箱：easybooks@163.com

编 者

CONTENTS



第 1 章 Visual C++概述1	
1.1 C++概述.....1	
1.1.1 C++程序创建.....1	
1.1.2 C++代码结构.....3	
1.2 使用 Visual C++.....4	
1.2.1 控制台应用程序向导.....4	
1.2.2 认识开发环境布局.....6	
1.2.3 认识工具栏.....7	
1.2.4 操作工具栏.....8	
1.2.5 项目工作区窗口.....9	
1.2.6 退出 Visual C++ 6.0.....10	
1.3 代码编辑和错误修正.....10	
1.3.1 代码编辑.....10	
1.3.2 修正语法错误.....11	
1.4 常见问题解答.....12	
1.5 实验实训.....13	
思考与练习.....14	
第 2 章 常量、变量和运算15	
2.1 常量和变量.....15	
2.1.1 基本数据类型.....15	
2.1.2 字面常量.....17	
2.1.3 变量及其命名规则.....18	
2.1.4 标识符常量.....20	
2.2 运算.....22	
2.2.1 算术运算.....22	
2.2.2 赋值运算.....23	
2.2.3 数据类型转换.....24	
2.2.4 自增和自减.....25	
2.2.5 sizeof 运算符.....25	
2.2.6 逗号运算符.....26	
2.2.7 位运算符.....26	
2.2.8 优先级和结合性.....27	
2.3 基本输入输出.....28	
2.3.1 输入流(cin).....29	
2.3.2 输出流(cout).....29	
2.3.3 使用格式算子 oct、dec 和 hex.....30	
2.4 常见问题解答.....30	
2.5 实验实训.....31	
思考与练习.....32	
第 3 章 顺序、选择和循环34	
3.1 顺序结构和块.....34	
3.1.1 说明语句.....34	
3.1.2 表达式语句.....34	
3.1.3 块语句.....35	
3.2 选择结构.....35	
3.2.1 条件的构成.....36	
3.2.2 if 语句.....37	
3.2.3 ?:运算符.....38	
3.2.4 switch 语句.....39	
3.3 循环结构.....40	
3.3.1 while 语句.....40	
3.3.2 do...while 语句.....41	
3.3.3 for 语句.....42	
3.3.4 break 和 continue.....43	
3.4 结构化程序设计应用.....44	
3.4.1 算法和流程图.....44	
3.4.2 自动出题器.....46	
3.4.3 打印图案.....47	
3.5 常见问题解答.....48	
3.6 实验实训.....49	

思考与练习	50
第4章 函数、作用域和编译预处理	53
4.1 函数	53
4.1.1 函数的定义和调用	53
4.1.2 函数的参数传递	56
4.1.3 带默认形参值的函数	56
4.1.4 函数重载	57
4.1.5 函数的递归调用	58
4.2 作用域和存储类型	60
4.2.1 作用域	60
4.2.2 存储类型	62
4.3 编译预处理	64
4.3.1 宏定义	64
4.3.2 文件包含命令	65
4.3.3 条件编译命令	65
4.3.4 包含重复处理	67
4.4 常见问题解答	68
4.5 实验实训	68
思考与练习	68
第5章 数组、指针和引用	71
5.1 数组	71
5.1.1 一维数组	71
5.1.2 二维数组	73
5.1.3 数组与函数	74
5.1.4 排序	75
5.2 指针和引用	78
5.2.1 地址和指针	78
5.2.2 指针定义和引用	79
5.2.3 &和*运算符	79
5.2.4 指针运算	80
5.2.5 指针和数组	81
5.2.6 指针和函数	83
5.2.7 new 和 delete	83
5.2.8 引用和引用传递	84
5.3 字符串及其操作	85
5.3.1 字符数组	86
5.3.2 字符指针	86
5.3.3 带参数的 main 函数	87
5.3.4 字符串处理函数	88
5.4 常见问题解答	90
5.5 实验实训	91
思考与练习	92

第6章 结构和链表	95
6.1 结构体	95
6.1.1 结构体变量	95
6.1.2 结构数组	97
6.1.3 结构与函数	98
6.1.4 结构指针	99
6.2 共用体和自定义	100
6.2.1 共同体	100
6.2.2 使用 typedef	101
6.3 简单链表	102
6.3.1 链表概述	102
6.3.2 链表的创建和遍历	103
6.3.3 链表的基本操作	105
6.3.4 求解 josephus 问题	108
6.4 常见问题解答	110
6.5 实验实训	110
思考与练习	111
第7章 类、对象和成员	112
7.1 类和对象定义	112
7.1.1 类的声明和定义	112
7.1.2 对象定义和成员访问	114
7.1.3 类作用域和成员访问权限	114
7.1.4 构造函数和析构函数	115
7.2 对象的使用	117
7.2.1 对象赋值和拷贝	117
7.2.2 浅拷贝和深拷贝	118
7.2.3 对象成员的初始化	120
7.3 成员特性	121
7.3.1 静态成员	121
7.3.2 常类型	123
7.3.3 this 指针	125
7.4 常见问题解答	126
7.5 实验实训	127
思考与练习	128
第8章 继承、多态和流	130
8.1 继承和派生	130
8.1.1 单继承	130
8.1.2 派生类的构造和析构	133
8.1.3 多继承	134
8.2 多态和虚函数	134
8.2.1 虚函数	135
8.2.2 纯虚函数和抽象类	136

8.3 运算符重载	137	10.2.2 创建对话框应用程序	181
8.3.1 运算符重载概述	137	10.2.3 添加对话框资源	182
8.3.2 赋值符重载	139	10.2.4 设置对话框属性	183
8.3.3 转换函数	140	10.2.5 添加和布局控件	184
8.4 输入输出流	142	10.3 使用对话框	186
8.4.1 流类和流对象	142	10.3.1 创建对话框类	186
8.4.2 输入输出成员函数	143	10.3.2 映射 WM_INITDIALOG 消息	187
8.4.3 文件流及其处理	145	10.3.3 在程序中调用对话框	188
8.4.4 顺序和随机文件操作	147	10.3.4 模式和无模式对话框	189
8.5 常见问题解答	150	10.3.5 通用对话框	189
8.6 实验实训	152	10.4 常见问题解答	190
思考与练习	152	10.5 实验实训	190
第9章 MFC 应用程序基础	154	思考与练习	192
9.1 MFC 编程	154	第11章 常用控件	193
9.1.1 MFC 概述	154	11.1 创建和使用控件	193
9.1.2 一个 MFC 程序	155	11.1.1 控件的通用属性和消息	193
9.1.3 程序运行机制	157	11.1.2 控件类和控件变量	194
9.2 应用程序向导	157	11.1.3 DDX 和 DDV	195
9.2.1 MFC AppWizard 概述	157	11.1.4 示例：使用控件变量	195
9.2.2 创建文档应用程序	158	11.2 静态控件和按钮	197
9.2.3 项目文件和配置	161	11.2.1 静态控件	197
9.2.4 项目工作区	162	11.2.2 按钮	198
9.2.5 应用程序类框架	163	11.2.3 示例：制作问卷调查	199
9.3 消息映射和类向导	164	11.3 编辑框和旋转按钮	202
9.3.1 消息映射机制	164	11.3.1 编辑框	202
9.3.2 使用类向导	165	11.3.2 旋转按钮	203
9.3.3 常见消息	167	11.3.3 示例：学生成绩输入	204
9.3.4 消息对话框	169	11.4 列表框	207
9.4 Visual C++ 常用操作	170	11.4.1 列表框样式和消息	207
9.4.1 成员的添加和删除	170	11.4.2 列表框基本操作	208
9.4.2 文件打开和成员定位	171	11.4.3 示例：城市邮政编码	209
9.4.3 使用向导工具栏	172	11.5 组合框	211
9.5 常见问题解答	172	11.5.1 组合框样式和消息	212
9.6 实验实训	173	11.5.2 组合框常见操作	213
思考与练习	174	11.5.3 示例：城市邮编和区号	214
第10章 窗口和对话框	175	11.6 进展条和日历控件	216
10.1 框架窗口	175	11.6.1 进展条	217
10.1.1 主窗口和文档窗口	175	11.6.2 日历控件	217
10.1.2 窗口样式的设置	176	11.6.3 计时器	218
10.1.3 窗口状态的改变	178	11.6.4 示例：自动时间显示	219
10.2 创建和布局对话框	180	11.7 滚动条和滑动条	219
10.2.1 资源和资源标识	180		

11.7.1 滚动条	220	13.2.1 一般视图框架	248
11.7.2 滑动条	221	13.2.2 图像列表	249
11.7.3 示例: 调整对话框背景颜色	222	13.2.3 列表视图框架	250
11.8 常见问题解答	224	13.2.4 树视图框架	254
11.9 实验实训	224	13.3 文档视图结构	259
思考与练习	225	13.3.1 作用函数	259
第 12 章 基本界面元素	226	13.3.2 切分窗口	260
12.1 图标和光标	226	13.3.3 一档多视	261
12.1.1 图像编辑器	226	13.4 常见问题解答	266
12.1.2 图标及其使用	227	13.5 实验实训	267
12.1.3 光标及其使用	230	思考与练习	267
12.2 菜单	231	第 14 章 图形和数据库	268
12.2.1 用编辑器设计菜单	232	14.1 图形和文本	268
12.2.2 使用键盘快捷键	234	14.1.1 图形绘制	268
12.3 工具栏	235	14.1.2 GDI 位图	271
12.3.1 使用工具栏编辑器	235	14.1.3 字体与文字	272
12.3.2 工具按钮和菜单项相结合	236	14.2 MFC ODBC 数据库	276
12.4 状态栏	237	14.2.1 MFC ODBC 向导过程	276
12.4.1 状态栏的定义	237	14.2.2 数据表绑定更新	279
12.4.2 状态栏的常用操作	238	14.2.3 显示记录总数和当前记录号	280
12.4.3 改变状态栏的风格	239	14.2.4 查询记录	281
12.5 常见问题解答	240	14.2.5 编辑记录	282
12.6 实验实训	241	14.3 常见问题解答	284
思考与练习	241	14.4 实验实训	285
第 13 章 数据、文档和视图	242	思考与练习	288
13.1 数据和文档序列化	242	附录 A 程序简单调试	289
13.1.1 字串资源	242	附录 B 大作业一: 学生成绩管理	292
13.1.2 文档序列化	243	(C++版)	
13.1.3 使用 CArchive 类	245	附录 C 大作业二: 学生成绩管理	299
13.1.4 简单数组集合类	246	(MFC 版)	
13.1.5 示例: 文档读取并显示	247	附录 D 创建 Access 数据库和数据表	305
13.2 视图及应用框架	248		

Visual C++概述

Visual C++是 Microsoft 公司推出的目前使用极为广泛的基于 Windows 平台的可视化编程环境。Visual C++ 6.0 是在以往版本不断更新的基础上形成的,由于其功能强大、灵活性好、完全可扩展以及具有强有力的 Internet 支持,在各种 C++语言开发工具中脱颖而出,成为目前最为流行的 C++语言集成开发环境。

Visual C++ 6.0 分为标准版、专业版和企业版三种,但其基本功能是相同的。Visual C++ 6.05 中文版是在 Visual C++ 6.0 基础上进行汉化的一个版本,本书以此版本作为编程环境。为统一起见,本教程仍称之为 Visual C++ 6.0,并以 Windows XP 作为操作系统平台。

要学习和应用 Visual C++, C++语言是基础。

1.1 C++概述

C++是在 20 世纪 80 年代初期由贝尔实验室设计的一种在 C 语言的基础上增加了对面向对象程序设计支持,是目前应用最为广泛的编程语言。

1.1.1 C++程序创建

使用 C++高级语言编写的程序称为源程序。由于计算机只能识别和执行的是由 0 和 1 组成的二进制指令,称为机器代码,因而 C++源程序是不能被计算机直接执行的,必须转换成机器代码才能被计算机执行。这个转换过程就是编译系统对源代码进行编译和连接的过程,简称“编连”过程。如图 1.1 所示。

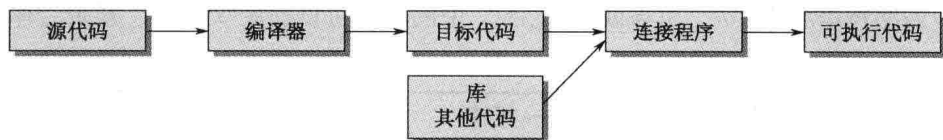


图 1.1 C++程序创建过程

事实上,对于 C++程序的源代码编写、编译和连接的步骤,许多 C++编程工具软件商都提供了各自的 C++集成开发环境(Integrated Development Environment, IDE)用于程序上述步骤的一体化操作。常见的有: Microsoft Visual C++、各种版本的 Borland C++(如 Turbo C++、C++ Builder 等)、IBM Visual Age C++和 bloodshed 免费的 Dev-C++等。不过,尽管 C++ Builder 6.0、Dev-C++和 Visual C++ 6.0 对 ANSI/ISO C++支持都比较好,但 Visual C++ 6.0 在项目文件管理、调试以及操作的亲和力等方面上都略胜一筹,故本书以 Visual C++ 6.05 SP6 中文企业版作为编程环境。

下面以一个简单的 C++ 程序为例来说明在 Visual C++ 中创建和运行的一般过程。

需要说明的是，由于 Visual C++ 对应用程序是采用文件夹的方式来管理的，即一个程序项目的源代码、编译的中间代码、连接的可执行文件等内容均放置在与程序项目名同名的文件夹中及其 debug（调试）或 release（发行）子文件夹中。因此，在用 Visual C++ 进行应用程序开发时，一般先要创建一个工作文件夹，以便于集中管理和查找。

1. 创建工作文件夹

创建 Visual C++ 6.0 的工作文件夹“D:\Visual C++ 程序”，以后所有创建的应用程序项目都在此文件夹下。在该文件夹下再创建一个子文件夹“第 1 章”，这样本小程序均在该文件夹下。下一章就在文件夹下创建的子文件夹“第 2 章”中，依此类推。

2. 启动 Visual C++ 6.0

选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”，运行 Visual C++ 6.0。

第一次运行时，将显示如图 1.2 的“每日提示”对话框。单击 **下一条(N)** 按钮，可看到有关各种操作的提示。如果在“启动时显示提示”复选框中单击鼠标，去除复选框的选中标记“✓”，那么下一次运行 Visual C++ 6.0，将不再出现此对话框。单击 **关闭(C)** 按钮关闭此对话框，进入 Visual C++ 6.0 开发环境，如图 1.3 所示。

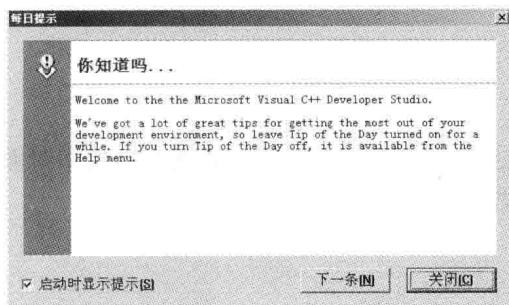


图 1.2 “每日提示”对话框

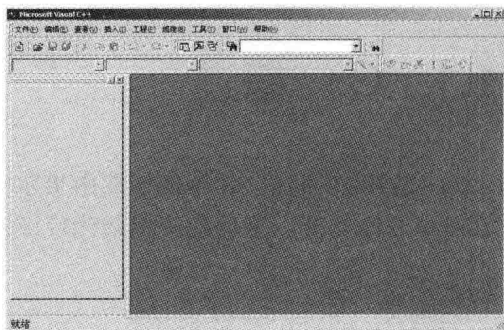


图 1.3 无项目的 Visual C++ 界面

3. 添加 C++ 程序

(1) 单击标准工具栏  上的“新建”(N)按钮，打开一个新的文档窗口，在这个窗口中输入下列 C++ 代码。注意：在输入字符和汉字时，要切换到相应的输入方式中；除了双引号中的字符串和由“//”引导或“/*...*/”中的注释可以使用汉字外，其余一律用英文字符输入。

【例 Ex_Simple1】 一个简单的 C++ 程序

```
/* 第一个简单的 C++ 程序 */
#include <iostream.h>
int main()
{
    double r, area; // 定义变量 r, area 双精度实数类型
    cout<<"输入圆的半径: "; // 显示提示信息
    cin>>r; // 从键盘上输入的值存放到 r 中
    area = 3.14159 * r * r; // 计算圆面积, 结果存放到 area 中
    cout<<"圆的面积为: "<<area<<"\n";
    return 0; // 指定返回值
}
```

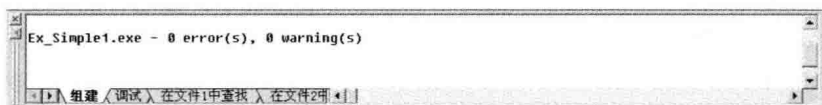
本书约定：书中凡是需要用户添加或修改的代码均用填充底纹来标明。

(2) 选择“文件”→“保存”菜单或按快捷键 Ctrl+S 或单击标准工具栏的“”按钮，弹出“保存为”文件对话框。将文件定位到“D:\Visual C++程序\第1章”文件夹中，文件名指定为“Ex_Simple1.cpp”（注意扩展名.cpp 不能省略，cpp 是 C Plus Plus 的缩写）。

此时在文档窗口中所有代码的颜色都发生改变，这是 Visual C++ 6.0 的文本编辑器所具有的语法颜色功能，绿色表示注释（例如//...），蓝色表示关键词（例如 double）等。

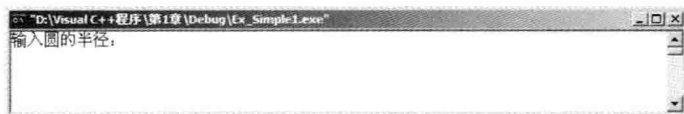
4. 编连和运行

(1) 单击编译工具条     上的生成工具按钮“”或直接按快捷键 F7，系统弹出一个对话框，询问是否为该程序创建默认的活动工作区文件夹，单击  按钮，系统开始对 Ex_Simple1 进行编译、连接，同时在输出窗口中显示编连的有关信息，当出现：



表示 Ex_Simple1.exe 可执行文件已经正确无误地生成了。

(2) 单击编译工具条     上的运行工具按钮“”或直接按快捷键 Ctrl+F5，就可以运行刚刚生成的 Ex_Simple1.exe 了，结果弹出这样的窗口（其背景属性已被修改过），它称之为控制台窗口，是一种与传统 DOS 环境相兼容的窗口：



此时等待用户输入一个数。当输入 10 并按 Enter 键后，控制台窗口显示为：



其中，“Press any key to continue”是 Visual C++ 自动加上去的，表示 Ex_Simple1 运行后，按一个任意键将返回到 Visual C++ 开发环境，这就是 C++ 程序的创建、编连和运行过程。

本书约定：在以后的 C++ 程序运行结果中，本书不再完整显示其控制台窗口，也不再显示“Press any key to continue”，仅将控制台窗口中运行结果列出。

1.1.2 C++ 代码结构

从上面的程序可以看出，一个 C++ 程序由编译预处理指令、数据或数据结构定义和若干个函数组成。在 C++ 中，一个程序可以存放在一个或多个文件中，这样的文件称为源程序文件。为了与其它文件相区别，每一个 C++ 源程序文件通常是以 .cpp 为扩展名。这里再以【例 Ex_Simple1】的程序代码来分析 C++ 程序的组成和结构。

1. main 函数

(1) 主函数。一个 C++ 语言程序可以包含若干个函数，其中 main 表示主函数，由于每一个程序执行时都必须从 main 开始，而不管该函数在整个程序中的具体位置，因此每一个 C++ 程序必须包含一个且只有一个 main 函数。

(2) 函数体。每一个函数由函数头和函数体组成。在 main 函数中，“int main()”称为 main 函数的函数头。函数头下面用一对花括号“{”和“}”括起来的部分就是函数体。函数体中

包括若干条语句，每一条语句都由分号“;”结束。

(3) 函数值。函数返回的值就是函数值，函数头包括函数值类型和函数名。由于 main 函数名的前面有一个 int，它表示 main 函数的类型是整型。函数值就是函数体中的 return 语句返回的值。上面 main 函数体包含“return 0”，所以 main 函数值是 0。如果函数体描述的过程不需要返回函数值，函数就没有值。此时，函数头包括函数值类型就是 void。

2. 预处理指令

#include <iostream.h>称为预处理指令。iostream.h 是 C++编译系统自带的文件，称为 C++库文件，它定义了标准输入/输出流的相关数据及其操作。由于程序用到了输入/输出流对象 cin 和 cout，因而需要用#include 将其合并到程序中。又由于它们总是被放置在源程序文件的起始处，所以这些文件被称为头文件（Header File）。C++编译系统自带了许多这样的头文件，每个头文件都支持一组特定的“功能”，用于实现基本输入输出、数值计算、字符串处理等方面的操作。

需要说明的是，为了能突出 C++与 C 语言本身的不同，对于以往 C 语言的标准头文件(.h)也改用新的文件，去掉了“.h”扩展名。这就是说，【例 Ex_Simple1】代码中头文件 iostream.h 的包含指令应写成下面的新格式：

```
#include <iostream>
```

同时为使 iostream 中的定义对程序有效，还需使用下面名称空间编译指令来指定：

```
using namespace std; // 注意不要漏掉后面的分号
```

using 是一个在代码编译之前处理的指令。namespace 称为名称空间，它是 C++一个新的特性。这样，上述程序应改成：

```
#include <iostream>
using namespace std;
int main()
{
    ...
}
```

3. 注释

程序 Ex_Simple 中的“/*...*/”之间的内容或“//”开始一直到行尾的内容是用来注释的，它的目的只是为了提高程序的可读性，对编译和运行并不起作用。正是因为这一点，所注释的内容既可以用汉字来表示，也可以用英文来说明，只要便于理解就行。

一般来说，注释应在编程的过程中同时进行，不要指望程序编制完成后再补写注释。那样只会多花好几倍的时间，更为严重的是，时间长了以后甚至会读不懂自己写的程序。

4. 缩进

缩进是指程序在书写时不要将程序的每一行都由第一列开始，而且在适当的地方加进一些空格，也是为了提高程序的可读性。通常，在书写代码时，每个“}”花括号占一行，并与使用花括号的语句对齐。花括号内的语句采用缩进书写格式，缩进量为 4 个字符（一个默认的制表符）。

1.2 使用 Visual C++

对于 C++程序开发，除了【例 Ex_Simple1】过程外，还可直接使用 Visual C++应用程序向导来创建，同时还有相应的界面操作及程序语法错误修改的方法，下面就来讨论。

1.2.1 控制台应用程序向导

为了帮助用户快速创建应用程序，Visual C++提供了标准的应用程序框架结构。若创建用于控制台的常用 C++应用程序，则可有下列步骤和过程：

(1) 在 Visual C++ 6.0 开发环境的最上层菜单中, 选择“文件”→“新建”菜单命令, 则打开应用程序向导, 显示出“新建”对话框, 如图 1.4 所示。单击“工程”标签, 从列表框中选中 Win32 Console Application (Win32 控制台应用程序) 项 (图 1.4 中的标记①)。

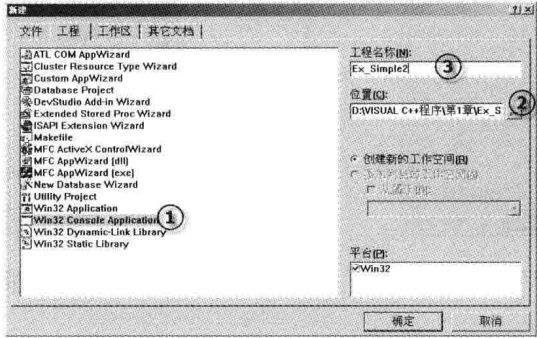


图 1.4 “新建”对话框工程页面

(2) 单击“位置”编辑框右侧的浏览按钮  (图 1.4 中的标记②), 从弹出的“选择目录”对话框指定项目所在的文件夹“D:\Visual C++程序\第 1 章”, 单击  按钮, “选择目录”对话框退出, 回到了“新建”对话框界面。

(3) 在“新建”对话框的“工程名称”编辑框 (图 1.4 中的标记③) 中输入名称 Ex_Simple2, 保留“平台”下“Win32”复选框的默认“选中  ”状态, 单击  按钮进入下一步。

(4) 出现 Win32 Console Application (Win32 控制台应用程序) 向导的步骤 1 (共 1 步) 对话框, 从中可选择要创建的应用程序类型: “一个空工程”、“一个简单的程序”、“一个“Hello, World!”程序”和“一个支持 MFC 的程序”, 如图 1.5 所示。

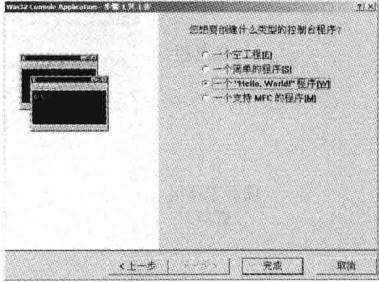


图 1.5 应用程序的向导对话框

这些应用程序类型的区别在于:

- “一个空工程”仅创建控制台应用程序文件框架, 不含任何代码;
- “一个简单的程序”是在“一个空工程”基础上添加了程序框架 (有入口函数、#include 指令等);
- “一个“Hello, World!”程序”在“一个简单的程序”基础上增加了 C 语言的 printf 函数调用, 用来输出“Hello World!”;
- “一个支持 MFC 的程序”则是支持 MFC 的控制台应用程序框架, 由于其相对复杂且一般人也不会使用它, 所以这里不去赘述。

(5) 选中“一个“Hello, World!”程序”, 单击  按钮, 弹出“新建工程信息”对话框, 如图 1.6 所示。单击  按钮, 系统将按前面的选择自动创建此应用程序。

(6) 又回到 Visual C++ 开发环境, 在菜单栏下面可以看见有许多工具条, 单击编译工具条  上的生成工具按钮  或直接按快捷键 F7, 系统开始对 Ex_Simple2 项目工程中的文件进行编译、连接, 同时在输出窗口中观察出现的内容, 当出现

Ex_Simple2.exe - 0 error(s), 0 warning(s)

表示 Ex_Simple2.exe 可执行文件已经正确无误地生成了。单击编译工具条  上的运行工具按钮  或直接按快捷键 Ctrl+F5, 就可以运行刚刚生成的 Ex_Simple2.exe, 结果如图 1.7 所示。事实上, 上述创建的是一个 C 程序。

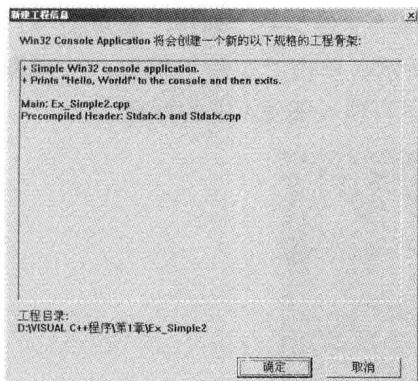


图 1.6 “新建工程信息”对话框

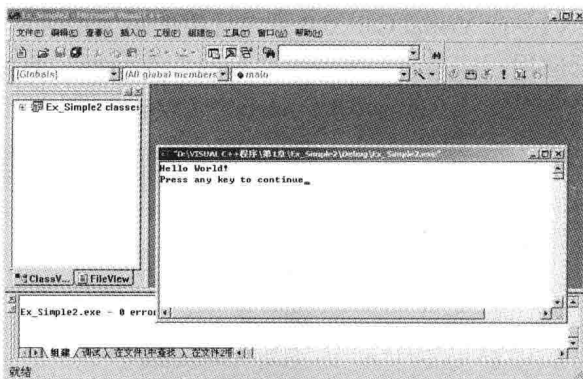


图 1.7 开发环境和运行结果

1.2.2 认识开发环境布局

当项目创建后，Visual C++ 6.0 开发环境如图 1.8 所示。它是由标题栏、菜单栏、工具栏、项目工作区窗口、文档窗口、输出窗口以及状态栏等组成。

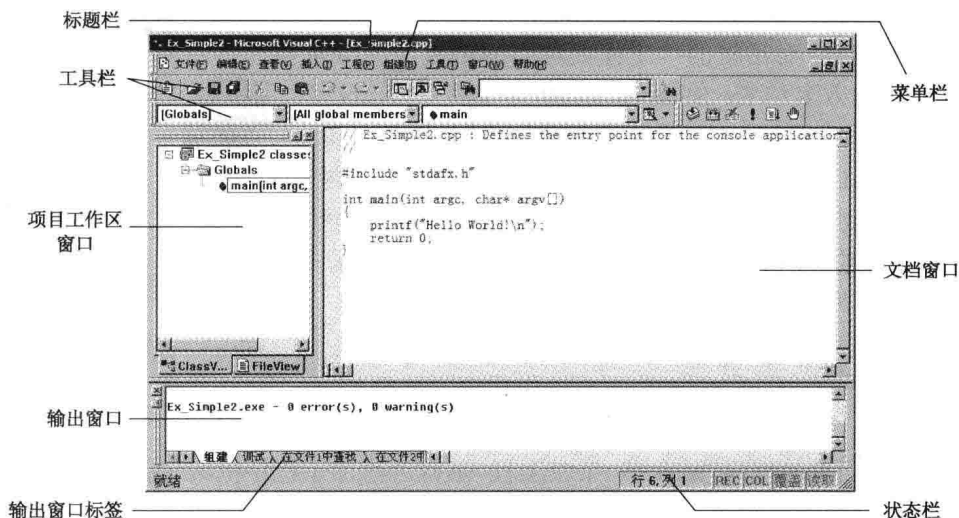


图 1.8 Visual C++ 6.0 开发环境（有项目）

标题栏处在开发环境的最上层，一般有“最小化”（）、“最大化”（）或“还原”（）以及“关闭”（）按钮，单击按钮将退出开发环境。事实上，标题栏上还会显示出当前文档窗口中正在操作的文档的文件名。

标题栏的下面就是菜单栏，它包含了开发环境中几乎所有的命令，提供了文档操作、程序编译、调试、窗口操作等一系列的功能。菜单栏往往和工具栏同处在一个区域，并且菜单中的一些常用命令还被排列在相应的工具栏上，以便更快捷地操作。

在开发环境的左侧是一个带有多个标签页面的项目工作区窗口，它包含当前项目的几乎所有的信息，包括项目中的类（ClassView 页面）信息、文件（FileView 页面）信息以及 Windows 项目的资源（ResourceView 页面）信息等。在这些信息项目中的文字或图标处右击鼠标都会弹出相应的快捷菜单，从中可选择当前状态下的一些快捷操作。

文档窗口一般位于开发环境中的右边，各种程序代码的源文件、资源文件、文档文件等都可以通过文档窗口显示出来。

输出窗口一般出现在开发环境窗口的底部，它包括了编译（Build、组建）、调试（Debug）、在文件中查找（Find in Files）等相关信息的输出。这些输出信息以多页面标签的形式出现在输出窗口中，例如“组建”页面标签显示的是程序在编译和连接时的进度及错误信息。

状态栏一般位于开发环境的最底部，它用来显示当前操作状态、注释、文本光标所在的行列号等信息。

1.2.3 认识工具栏

菜单栏下面是工具栏。工具栏上的按钮通常和一些菜单命令相对应，提供了执行经常使用的命令的一种快捷方式。Visual C++ 6.0 开发环境默认显示的工具栏有：“标准”（Standard）工具栏、“向导”（WizardBar）工具栏及“编译微型条”（Build MiniBar）工具栏。

（1）标准工具栏。如图 1.9 所示，标准工具栏中的工具按钮命令大多数是常用的文档编辑命令，如新建、保存、撤消、恢复、查找等，表 1.1 列出了各个按钮命令的含义。

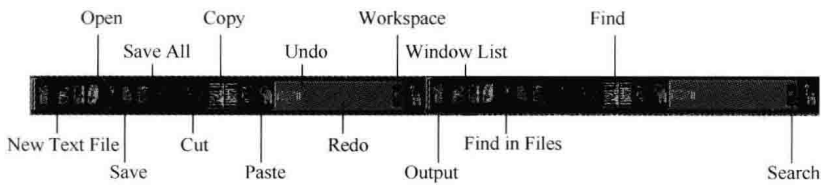


图 1.9 标准工具栏

表 1.1 标准工具栏按钮命令及功能描述

按 钮 命 令	功 能 描 述
New Text File	新建一个文本文件
Open	打开已存在的文件
Save	保存当前文件
Save All	保存所有打开的文件
Cut	将当前选定的内容剪切掉，并移至剪贴板中
Copy	将当前选定的内容复制到剪贴板中
Paste	将剪贴板中的内容粘贴到光标当前位置处
Undo	撤消上一次操作
Redo	恢复被撤消的操作
Workspace	显示或隐藏项目工作区窗口
Output	显示或隐藏输出窗口
Window List	文档窗口操作
Find in Files	在指定的多个文件(夹)中查找字符串
Find	指定要查找的字符串，按 Enter 键进行查找
Search	在当前文件中查找指定的字符串

（2）向导工具栏。向导工具栏是 Visual C++ 6.0 使用频率最高的，它是将 MFC ClassWizard（MFC 类向导，以后还会专门讨论）对话框的功能体现为三个相互关联的组合框和一个 Actions（操作）控制按钮，如图 1.10 所示。



图 1.10 WizardBar

三个组合框分别表示类信息（Class）、选择相应类的过滤器（Filter）和相应类的成员函数（Members）等。单击 Actions 控制按钮可将文本指针移动到指定类成员函数在相应的源文件的定义和声明的位置处，单击 Actions 向下按钮(▼)会弹出一个快捷菜单，从中可以选择要执行的命令。

(3) 编译微型条工具栏。编译微型条工具栏提供了常用的编译、连接操作命令，如图 1.11 所示，前面已使用过。表 1.2 列出了各个按钮命令的含义。

表 1.2 编译微型条工具栏按钮命令及功能描述

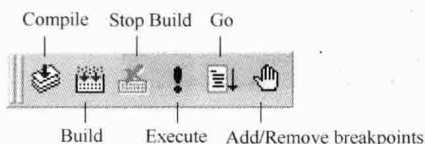


图 1.11 编译微型条工具栏

按钮命令	功能描述
Compile	编译 C 或 C++源代码文件
Build	生成应用程序的 EXE 文件
Stop build	停止编译
Execute	执行应用程序
Go	单步执行
Add/Remove breakpoints	插入或删除断点

需要说明的是，上述工具栏上的按钮有时是处于未激活状态，例如，标准工具栏的复制按钮在没有选定对象前是灰色的，这时无法使用它。

1.2.4 操作工具栏

在 Visual C++中，可以对有“把手”(☞)的工具栏（或菜单栏）进行显示与隐藏、浮动与停泊等操作。

1. 显示和隐藏

显示或隐藏工具栏可以使用“定制”对话框或快捷菜单两种方式进行操作。先来看看“定制”对话框方式：

① 选择“工具”菜单→“定制”命令项。

② 弹出“定制”对话框，如图 1.12 所示；单击“工具栏”页面标签，将显示出所有的工具栏名称，那些显示在开发环境上的工具栏名称前面将带有选中标记(✓)。

如果认为上述操作不够便捷，那么可以在开发环境的工具栏处右击鼠标，这时就会弹出一个包含工具栏名称的快捷菜单，如图 1.13 所示，这就是“快捷菜单”方式。若要显示某工具栏，只需用单击该工具栏名称，使得前面的复选框带有选中标记即可。同样的操作再进行一次，工具栏名称前面的复选框的选中标记将消失，该工具栏就会从开发环境中消失。

2. 浮动与停泊

Visual C++ 6.0 的工具栏具有“浮动”与“停泊”功能。当 Visual C++ 6.0 启动后，系统默认将常用工具栏“停泊”在主窗口的顶部。若将鼠标指针移至工具栏的“把手”(☞)处或其它非按钮区域，然后按住鼠标左键，可以将工具栏拖动到主窗口的四周或中央。如果拖动