



独家引进 专有版权
新潮流电脑图书

轻松快乐 学 **VISUAL C++**

曾中浩 徐文斌 编著
陈劲州 徐华声



中山大学出版社

出版说明

中山大学出版社向台湾松岗电脑资料股份有限公司购买版权，出版了如下6本书：《轻松快乐计算机知识扫盲》、《轻松快乐学 PowerPoint 4.0 中文版》、《轻松快乐学 Delphi》、《轻松快乐学 Visual C++》、《轻松快乐学 Windows 95 中文版》和《轻松快乐学 BBS & Internet》。这些计算机图书有两个特点：一是资料比较新，能跟上计算机技术发展的潮流；二是写得生动活泼，适合大众阅读和使用，故用“轻松快乐”以冠之。

在编辑出版上述图书的过程中，遇到的一些技术性问题，有必要加以讨论和说明。

一、海峡两岸的文字差异问题

海峡两岸使用不同体例的汉字：简体字和繁体字。要把台湾版的计算机图书改为简体字出版，正文是不成问题的，只要对应地把繁体汉字改为简体汉字就行。问题在于如何处理图形界面中的汉字。

有人主张把图形界面中的繁体字“挖改”为简体字，以求得“字体一致”。从技术上看，这原则上是可以办得到的，但至今未见有哪一家出版社是这样做的。究其原因，乃是“内容的实质性一致”比“字体一致”更为重要的缘故。由于繁体字版的计算机图书是台湾作者写的，书中所解说的是繁体汉字版的计算机软件，与正文叙述的软件操作相对应的图形界面理应出现繁体汉字，这就像叙述英文版的计算机软件时，它的图形界面一定是显示英文词一样。原因在于图形界面总是软件的运行结果，软件是“源”，界面是“流”，两者应该一致。如果将图形界面上的繁体汉字硬是改为简体汉字，岂非“泉眼”流出的是“牛奶”，歪曲了软件的实际运行结果？可以肯定地说，谁也不会主张为了“字体一致”而把英文计算机软件的图形界面改成中文图形界面，对于繁体汉字版亦应如此。

二、海峡两岸的计算机术语差异问题

海峡两岸使用的计算机术语是有差异的，有的差异还比较大。例如“硬件”与“硬体”，“鼠标”与“滑鼠”，“文件”与“档案”等。不过，即使如此，从理解与使用图书的角度而言是不成问题的，因为香港和台湾地区的电脑图书、杂志，硬件说明书，软件尤其是游戏软件等在我国大陆地区早已相当流行，导致其计算机术语亦为大家所熟悉。问题是从出版的角度，该如何处理才好，至今是没有明令定规的。

有人主张，把所有台湾的计算机术语都改为我国大陆地区的计算机术语，以求“术

语一致”。但若真的这样做，当遇到图形界面时，必然是正文中是“文件”，而图形界面中则为“档案”，出现“文图不一致”，而且这种不一致将令读者十分费解。于是有人提出，可否在保持“文图一致”的原则下，保留正文中与图中对应的术语不要改动，而把其他计算机术语一律改为“标准术语”。但这样做又会带来新的问题。计算机术语之间是有关联的，例如，“档案”、“关闭档案”、“开启档案”、“存储档案”等是一致的，若按“文图一致”的原则保留其中一两个在图中出现的，如“档案”，而把另外几个只在正文中出现的，如“关闭档案”、“打开档案”、“存储档案”等，改为“关闭文件”、“打开文件”、“存储文件”等，则全书的术语又会混乱了。

三、图形界面和术语的处理方法

基于上述的认识，我们认为大体上可以按下列几条原则，对图形界面和计算机术语加以灵活处理：①保留原繁体字版的图形界面不变，以保持“源（软件）流（界面）一致”；②修改有显著差别的术语，但当这种修改可能导致“文图不一致”时，则不予修改；③在差别不大或容易理解时，可以保留繁体字版的术语，以便尽可能保持原书的风格。由于有着同种同文的根基，而且版权交易也是文化交流的一部分，在不妨碍理解和使用且差别不大的情况下，在技术性图书中保留一些原版的技术词汇将更为有利于促进我国海峡两岸的文化和技术交流。

四、计算机术语对照表

为了避免所做或未做的术语修改可能导致读者的费解或误解，在出版购买版权（繁体汉字版）的简体汉字版计算机图书时，附上一个如下所示的“繁体—简体—英文”计算机术语对照表是合理和有用的。

繁体字术语	简体字术语	英文术语
资料	数据	Data
资料框	数据帧	Frame
资讯	信息	Information
磁碟	磁盘	Disk
档案	文件	File
韧体	固件	Firmware
硬体	硬件	Hardware
软体	软件	Software
萤幕	荧幕、屏幕	Shield
滑鼠	鼠标	Mouse
数据机	调制解调器	Modem
列印机	打印机	Printer

记忆体	存储器	Memory
I/O 埠	I/O 端口	I/O Port
周边设备	外部设备	Peripheral Devices
快速记忆体	高速缓冲存储器	Cache memory
体积电路	集成电路	Integrated circuit
人工智慧	人工智能	Artificial intelligence
类比	模拟	analog
运算码	操作码	Operation code
运算元	操作数	Operand
位址	地址	Address
资料汇流排	数据总线	Data bus
地址汇流排	地址总线	Address bus
冷开机	冷启动	Cold start
热开机	热启动	Warm start
字元	字符	Character
位元组	字节	Byte
磁轨	磁道	Track
磁区	扇区	Sector
规格化	格式化	format
同位元检查法	奇偶校检	Parity check
即时处理	实时处理	Real - Time processing
循序处理	顺序处理	Sequential access method
序列传输	串行传输	Serial transmission
网路	网络	Network
区域网路	局域网络	Area network
组合语言	汇编语言	Assembler language
行程	进程	Process
程式	程序	Program
子程式	子程序	Subprogram
阵列	数组	Array
批次档	批命令文件	Batch file
叙述	语句	Statement
巢状回路	嵌套循环	Nested loop

以上几点涉及实际出版工作中遇到的问题、认识和处理方法。问题客观存在，但认识是否正确，处理是否得当，则有待于广大读者和出版界同仁的认同或批评指正。我们

诚恳地希望得到各位有识之士的建设性意见，以利于我国海峡两岸的图书版权交易活动。

张亚拉
于中山大学
1996 年 11 月

自序

事情就是这样子开始的……

时间：1994 年 X 月（约倒退一年多的光阴）

地点：竹东少林寺（就是工业技术研究院）

场景：吃完晚饭后，一群人正在闲聊当中……

甲：“用 C++ 都两年多了，我们也该算是老贼的资历。”（注：从 MFC 1.0/MFC 7.0 来看，真是“大言不惭”）

乙：“当然是！都有勇气到外面顾问公司去上课、接受考验。”（注：绝没有影射任何事物，I Promise）

丙：“不过市面上倒没有几本介绍 Visual C++ 的中文书，有介绍的呢，好像也没谈到重点，MFC 都没提到几次。真是奇怪！”

众人：“对哦！”“真的很难看，只有原文的书还值得参考。”“可怜的 Programmer！”
“有同感。”“J. J. Hou 也还没有写吗？”

甲：“就趁着这机会，不如来规划出一本书，如何？”（注：带着开玩笑的语气）

众人：...XXX@@@...

经过漫长的煎熬与坚持，这本书真的终于出版了！

这本书的完成，可说是集合了众作者的智慧结晶；希望藉由多年的 Microsoft C++ Programming（合起来都超过“十”年了！）经验，能带给读者一些启示。

本书重点不在于如何操作 Visual C++ 的各种工具，而是在于如何掌握 MFC 的运用；因此从第 3 章后的各章节，每章都有其强调的重点，也包含此重点所需运用类别（Class）的介绍及范例程序的实际演练。也唯有了解各类别的特性及与其他类别的互动关系，才可以更迅速、确实地利用 MFC（或可叫 Application Framework）来产生自己的应用程序，提高自己的软件生产力。

下表中列出各章节的重点：

目 录

第一部分 Visual C++ 是什么

第 1 章 Visual C++ vs. MS Windows	(3)
1.1 MS Windows 的特色	(4)
1.1.1 图形的使用者界面	(4)
1.1.2 多用途的工作环境	(4)
1.1.3 多媒体的统一世界	(5)
1.1.4 共通性的操作界面	(5)
1.2 如何撰写 MS Windows 的应用程序	(5)
1.2.1 所谓的“资源”	(7)
1.2.2 Message 的接受	(7)
1.2.3 Graphics Device Interface (GDI)	(7)
1.2.4 动态连结程序库 (Dynamic Link Library, DLL)	(7)
1.3 什么是 Visual C++	(8)
1.4 Visual C++ 的组成	(10)
1.4.1 Visual Workbench——提高生产力的整合性开发环境	(11)
1.4.2 Project——Makefile 产生器	(11)
1.4.3 Source Browser——程序结构浏览器	(14)
1.4.4 Debugger——Bugs 终结者	(14)
1.4.5 AppStudio——产生及修改程序资源的工具	(15)
1.4.6 AppWizard——程序码产生器	(16)
1.4.7 ClassWizard——处理信息的好帮手	(18)
1.4.8 On—Line Help——不可或缺的工具	(18)
1.4.9 Books On line —— Information Wizard	(19)
1.4.10 MFC Class Library——程序设计师的宝典	(19)
1.5 欢迎来到 Visual C++ 的世界!	(20)
第 2 章 与 MFC 的第一次接触	(22)
2.1 MFC Class Library	(23)
2.1.1 MFC 简介	(23)
2.1.2 What, Why, Who, When, How	(23)
2.1.3 What——MFC 到底是什么东西?	(23)
2.1.4 Why——为什么要用 MFC?	(23)

2.1.5 Who——哪一类的程序开发者适于用 MFC?	(24)
2.1.6 When——什么时候是开始使用 MFC 的时机?	(24)
2.1.7 How——如何使用 MFC?	(24)
2.2 属于 MFC 2.x 的几点特色	(24)
2.3 一个重要但易被忽视的 Class: CObject	(31)
2.3.1 Serialization (Object Persistence)	(32)
2.3.2 Run-Time Class Information	(35)
2.4 继承或不继承 CObject	(36)
2.5 结语	(36)

第二部分 有用的 Class Library

第 3 章 写第一个 Visual C++ 的程序 (39)

3.1 第一个简单的 Windows 程序	(40)
3.1.1 范例程序介绍	(40)
3.1.2 范例程序的结构	(44)
3.2 CWinApp: 任何 Windows 程序的发动机	(45)
3.3 CWnd: 窗口环境中的“窗口”	(46)
3.3.1 什么是 Windows 对象	(47)
3.3.2 CWnd Class 结构	(47)
3.3.3 如何建立 Windows 对象	(49)
3.3.4 如何清除 Windows 对象	(51)
3.3.5 如何与 Windows 一同工作	(51)
3.4 范例程序剖析	(51)
3.5 结论	(53)

第 4 章 信息与使用者界面 (54)

4.1 Windows 中的“信息”(Message)	(55)
4.1.1 处理 Windows Message	(55)
4.1.2 Message 的种类	(59)
4.1.3 由 Message Maps 找寻对应的 Message Handler	(60)
4.2 User Interface Object	(61)
4.2.1 何谓“User-Interface Objects”	(63)
4.2.2 Update User-Interface Objects	(65)
4.2.3 Command Routing	(66)
4.3 菜单/工具栏/状态栏	(67)
4.3.1 菜单的制作	(67)
4.3.2 工具栏的制作	(67)
4.3.3 状态栏的制作	(68)
4.4 来一个可用的程序吧!	(69)
4.5 结论	(79)

第 5 章 Graphic Device Interface	(80)
5.1 Device Context Class	(81)
5.2 绘图对象及绘图模式	(82)
5.2.1 MFC 所包装的 GDI 对象	(82)
5.2.2 如何建立绘图的对象	(83)
5.2.3 如何使用这个对象	(83)
5.2.4 所谓绘图模式	(84)
5.3 对应模式	(85)
5.3.1 为什么会有对应模式	(86)
5.3.2 灵活运用各种对应模式	(88)
5.4 给程序来一点缤纷的色彩吧!	(90)
5.5 结论	(97)
第 6 章 对话框 (Dialog Box)	(98)
6.1 对话框——一个不可缺少的工具	(99)
6.1.1 MFC 中的对话框	(99)
6.1.2 Modal 和 Modeless Dialog Boxes	(99)
6.2 设计自己的 Dialog	(100)
6.2.1 AppStudio——设计 Dialog 的好帮手	(100)
6.2.1.1 启动 AppStudio	(100)
6.2.1.2 编辑新的或已存在的 Dialog	(102)
6.2.1.3 调整 Dialog 大小及内容	(104)
6.2.1.4 加入 Control	(106)
6.2.1.5 加入 User-Defined Controls	(107)
6.2.1.6 完成	(108)
6.2.2 ClassWizard——设计 Class 的好助手	(109)
6.2.2.1 启动 ClassWizard	(109)
6.2.2.2 加入 Data Member (DDX)	(111)
6.2.2.3 设定边界值 (DDV)	(113)
6.2.2.4 加入 Message Handler	(113)
6.3 Dialog Box 的生命周期	(114)
6.4 Dialog 数据的交换和验证	(116)
6.5 Common Dialog Classes	(117)
6.6 应用程序范例	(119)
6.7 结论	(132)

第三部分 Document vs. View

第 7 章 Document 与 View (I)	(135)
7.1 设计概念的大变动!	(136)
7.2 什么是 Document/View 结构	(136)

7.2.1	预铸式的程序	(136)
7.2.2	没有 Document/View 的日子	(137)
7.2.3	设计概念的新宠儿——Document/View	(139)
7.3	Document/View 结构中的基本成员	(140)
7.3.1	Document	(140)
7.3.2	View	(140)
7.3.3	Frame Window	(141)
7.3.4	Document Template	(141)
7.3.5	Application Object	(141)
7.4	Team Work! 各对象间的互动	(141)
7.4.1	鸟瞰——各 Object 之间相互沟通的关系	(142)
7.4.2	Application Object 与 Document Template Object	(142)
7.4.3	Document Template 与 Document, Frame Window, View	(146)
7.4.4	Document 与 View	(147)
7.4.5	View 与 Frame Window	(151)
7.5	水到渠成——命令信息传递	(152)
7.6	应用程序范例	(155)
7.7	结论	(163)
第 8 章	Document 与 View (II)	(164)
8.1	壁垒分明的 SDI 和 MDI	(165)
8.1.1	什么是 SDI, MDI	(165)
8.1.2	程序设计上的差异	(165)
8.1.3	AppWizard 产生程序主体	(171)
8.2	应用程序范例	(173)
8.3	再看看 View 的绘图方式	(189)
8.3.1	C/SDK 的绘图方式	(189)
8.3.2	Document/View 的绘图方式	(190)
8.4	结论	(195)
第 9 章	文件 I/O 与 Serialization	(197)
9.1	MFC 与文件处理	(198)
9.1.1	文件处理	(198)
9.1.2	MFC 中的 CFile	(198)
9.1.3	CFile 及其衍生的 CStdioFile 和 CMemFile	(200)
9.2	Document 与 Serialization	(201)
9.2.1	Serialization	(201)
9.2.2	Framework 与 Serialization	(201)
9.2.3	Document 与 Serialization	(202)
9.2.4	使用 Serialization, 一切 OK?	(202)
9.3	细部分解——近观 Serialization	(203)

9.3.1 CArchive——仓库管理员	(203)
9.3.2 CArchive——如何管理 Object	(204)
9.3.3 如何建立和结束 CArchive 对象	(205)
9.4 存取对象数据	(206)
9.4.1 通过 Archive Object 载入/储存数据	(206)
9.4.2 Serialize Embedded Object 及 Pointers	(208)
9.4.3 Serialize Collections	(210)
9.5 做一些不同的 Serialization	(211)
9.5.1 如何动手改变	(211)
9.5.2 保密防谍自己来——实例演练	(214)
9.6 轻轻松松学应用	(215)
9.7 结论	(225)
第 10 章 打印及模拟显示	(226)
10.1 Windows 的打印功能	(227)
10.2 模拟显示的功能	(228)
10.3 打印的程序设计	(228)
10.4 AFX 中关于打印的运作方式	(229)
10.5 模拟显示的达成	(231)
10.6 程序范例	(232)

第四部分 一个完整的应用程序

第 11 章 Context-Sensitive Help	(243)
11.1 软件中的 Help 功能	(244)
11.1.1 “Help”系统	(244)
11.1.2 Windows 的 Help 系统	(244)
11.2 制作一个简单的 Help 文件	(245)
11.3 由程序中调用 Help	(247)
11.4 利用 AFX 在程序中加入 Help 功能	(248)
11.5 Context-Sensitive Help 的功能	(250)
11.6 AFX 对于 Help 命令的处理	(251)
11.7 程序范例	(251)
11.8 结语	(252)
第 12 章 轻轻松松来应用	(258)
12.1 设计的理念	(259)
12.1.1 C++ 应用时的基本概念	(259)
12.1.2 应用程序的框架是怎么建立的	(259)
12.1.3 MFC 和 Windows API 可以混用吗	(260)
12.1.4 MFC 的源头——CObject	(260)
12.1.5 WinMain() 怎么不见了	(261)

12.1.6 继承否? 延伸否?	(261)
12.1.7 Message Map——信息的分派者	(262)
12.2 各种具有特色的 Classes	(265)
12.2.1 与使用者界面有关的“视觉”Classes	(265)
12.2.2 Document 与 View——一体之两面	(267)
12.2.3 Serialization——存取文件的好帮手	(269)
12.2.4 荧幕与打印——毕其功于一役	(271)
12.3 快快乐乐来写程序	(271)
12.4 结论	(289)
附录 A Keyword 名词对照	(290)
附录 B MFC 的 Message—Map	(294)
B.1 WM_COMMAND Message Handler	(295)
B.2 Child Window Notification Message Handlers	(295)
B.2.1 Generic Control Handler	(295)
B.2.2 User Button Handlers	(295)
B.2.3 Combo Box Handlers	(295)
B.2.4 Edit Control Handlers	(296)
B.2.5 List Box Handlers	(296)
B.3 WM_Message Handlers	(297)
B.4 User-Defined Message Handlers	(299)
附录 C 参考数据	(300)
附录 D 错误诊断与异常处理	(302)
D.1 前言	(303)
D.2 除错版与完成版	(303)
D.3 转储对象内容	(304)
D.4 TRACE 宏	(306)
D.5 ASSERT 宏	(307)
D.6 ASSERT_VALID 宏	(308)
D.7 内存使用的检查	(308)
D.8 异常处理机制	(310)
D.9 异常处理方式	(312)
D.10 自行丢出异常对象	(314)
D.11 定义异常类别	(314)
D.12 结语	(315)

第一部分

Visual C++ 是什么

第 1 章 Visual C++ vs. MS Windows

第 2 章 与 MFC 的第一次接触

章别	章名	重点
一	Visual C++ vs. MS Windows	Visual C++ 的全貌
二	与 MFC 的第一次接触	MFC 的基本认识
三	写第一个 Visual C++ 的程序	如何构建一个最基本的应用程序
四	信息与使用者界面	所谓的“信息”
五	Graphic Device Interface	画图的必备工具
六	对话框	对话框的操作与运用
七	Document 与 View (I)	Document/View 的结构
八	Document 与 View (II)	Document/View 的实作
九	文件 I/O 与 Serialization	I/O 的处理
十	打印与模拟显示	MFC 的打印与模拟显示
十一	Context-Sensitive Help	如何建立自己的 Help
十二	轻轻松松来应用	回顾
附录 A	名词对照	本书中中英名词的对照
附录 B	Message-Map	所有的 MFC Message-Map
附录 C	参考数据	作者们使用的参考数据
附录 D	错误诊断与异常处理	Debug 的必需用品

要感谢的很多，在此也不一一列出。特别谢谢我们的朋友茆世民先生，提供了附录 D：“错误诊断与异常处理”的文章，相信这一篇文章的介绍对程序的写作者有极大的帮助。

写这一本书真的是很辛苦，几次都差点中断掉，幸赖大家的通力合作才得以完成。辛苦了！各位伙伴。如果这本书能带给读者任何的帮助、启发，那真是我们最大的荣幸。

愿各位 Programmer 在软件的世界努力再加油！

曾中浩

1995 年 8 月 8 日于台湾新竹

第 1 章

Visual C++ vs. MS Windows

听说过“Visual C++”吧！想了解“Visual C++”的全貌吗？想知道这套工具的功能吗？且听我们慢慢道来！

这是本书的第1章，我们大家怀着谨慎、兴奋的心情，希望能把我们努力的过程忠实地记录下来。这本书中主要介绍的系以 Visual C++ 为基础的 MS Windows Programming；希望能从实用的角度出发，介绍 Visual C++ 的点点滴滴。

1.1 MS Windows 的特色

“Microsoft Windows”这个名词应该是个人电脑从业人员所熟悉的名字吧！从1990年5月 MS Windows 3.0 的问世开始，这套操作环境（实在无法称之为操作系统，但谁叫它那么聪明地以 MS-DOS 操作系统作为垫脚石）立即吸引了成千上万的 MS-DOS 使用者的注目及欢迎，等到1992年4月 MS Windows 3.1 的成功演出，至此“Windows”这个名词成为了某种图腾：成千万（真的超过千万套哦！）的使用者视它为新一代的电脑环境，使人更易于亲近；显示卡、硬盘、鼠标及存储器靠着它的基本需求而需求量大增；Microsoft 公司也由这套系统获取大量利润；OS 竞争厂商也争先恐后地号称和“MS Windows 的应用程序”相容；至于我们“劳苦功高”的 Programmer 也只好被时势所趋地踏上 MS Windows Programming 的这一条路。

到底“MS Windows”为什么会创造出这样的风潮呢？相信在将来的电脑历史书中自有定论；对使用者而言，MS Windows 的主要特色（或者可以说是吸引点）有哪些呢？

1.1.1 图形的使用者界面

对一个现今使用电脑的人而言，似乎彩色的荧幕、图形化的界面是一件再正常不过的事情；但却不知仅仅在三四年前，大家的个人电脑却都还是黑白（嗯！该称为“单色”才对）的世界，与电脑的沟通都还是用敲键盘的！

等到 MS Windows 的流行，电脑一下子全都换了新装：色彩缤纷的世界！尤其在画图、影象处理及排版等方面更是如鱼得水，所谓的“What You See Is What You Get”（WYSIWYG）正是其写照；像现在我正在写的这本书，想要知道印出来的情形如何，不再需要真的印在纸上，只要在电脑荧幕上就可以看到其结果。

1.1.2 多用途的工作环境

想象一下如果你有下列事情想要做：写一份技术文件给老板、上 BBS 读点新鲜有趣的小道消息和把自己的程序文件备份起来，你会如何呢？如果在 MS-DOS 环境下，只好先将老板要的文件准备好，休息一下上 BBS，关机前再赶快将程序备份起来。在 MS Windows 中就不同了，先上 BBS 看看，至于给老板的文件呢？别急，启动一个文书处理器来编辑，也许一边看着 BBS 一边会有更多的灵感来写。写累了，休息一下，顺便开个

DOS 窗口将程序先备份好。等文件完成后，嗯！夜已深了，该离开 BBS 准备关机。不亦快哉！

当然，有读者会说：我还可以一面打印报表、编译程序再加上 Debug 呢。这也是可以办到的，但是执行太多的应用程序实在是 MS Windows 的一大“信任”（尤其是中文版的）！适可而止吧，总不能一心太多用吧！

1.1.3 多媒体的统一世界

“多媒体”！是个多么吸引人的口号，事实上，人类就生活在多媒体的世界中。个人电脑又能给使用者些什么东西呢？在 MS Windows 中，图形、影像已是其最基本的要求；声音方面，不管是 CD-DA（就是一般的 CD 唱片）、声音特效或是 MIDI（Musical Instrument Digital Interface）的支持都具备，使用者不管使用哪一家的硬件，只要是支持 Windows 的，任何 Windows 的应用程序都可以发挥其功用；视讯方面，提供一套软件解压缩的工具：Video for Windows，对目前风行的 MPEG 压缩则订出一套标准的 MCI（Media Control Interface）。

在这个统一的世界中，各软件厂商可以不必担心要支持各种的界面或系统，只要提供或使用 MS Windows 所提供的界面就可以了；每个人可以尽情地在自己专业的领域中发展。

1.1.4 共通性的操作界面

对一个初接触电脑的人而言，电脑是一个可怕的怪兽，每次不知道或不小心中按到什么键就发生搞不清楚的状况，能学会一套工具的操作技巧就是了不起的事了！

在 MS Windows 下就不用再担心了。真的，你只要熟悉一套工具的操作，其他的应用程序都依此要领办理，加上鼠标（Mouse）的应用，按一下左键就是选择，连按两下就是去执行，更是简化学习的时间。

1.2 如何撰写 MS Windows 的应用程序

产生 Windows 应用程序的工具，我们大致可以将其划分见图 1-1 所示。

明显的，本书就是介绍程序语言的方面，它最具威力，自然也要下最多的功夫。谈到使用 C 语言来撰写 MS Windows 的应用程序，真是 Programmer 的一篇血泪史。怎么说呢？且先撇开 ANSI C 的众多函数，光 Windows 本身就增加了数百个新的函数。再要发展 OLE，很好！加上百个函数，再要发展 Video for Windows，很好！加上百个函数……真是“多多益善”！

要想知道如何利用 C/C++ 来撰写 MS Windows 的应用程序，当然首先要先了解建