

C/C++

开发专家

把

水

VC++

Visual
C++



白乔 左飞 编著
飞思科技产品研发中心 监制

2008十大IT图书作者最新力作!

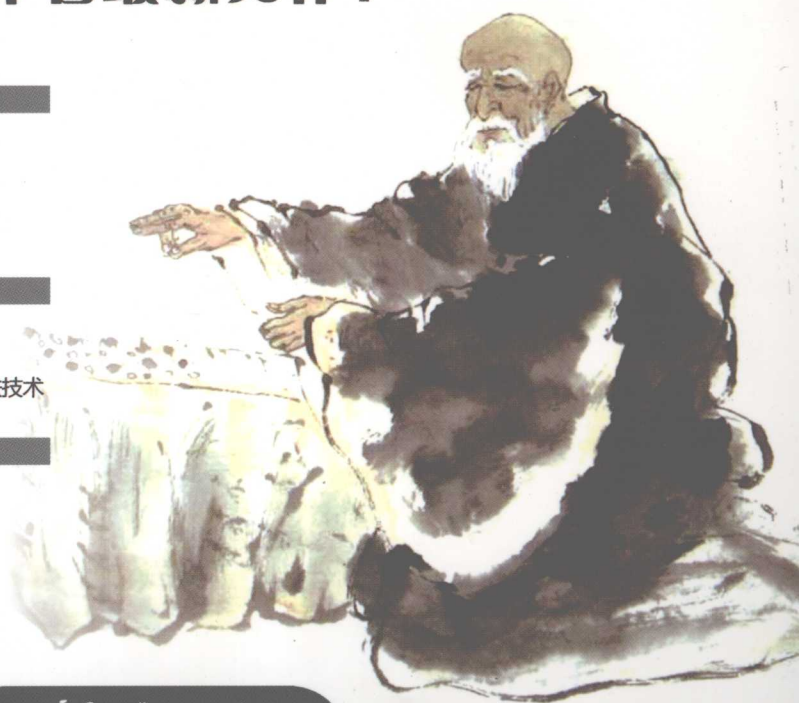
独辟蹊径，彰显 Visual C++ 编程脉络

对象的建模与表达技术
对象的持久化技术
对象的界面展现技术
对象的传输与调用技术

开放融合，以开放的思想阐述 Visual C++

标准 C++、STL、boost 标准库
OpenGL、DirectX、QT 等技术
Xtreme Toolkit、BCGControlBar、SkinMagic 换肤技术

代码剖析，揭示 MFC/ATL/COM 技术真谛



提供完整实例代码



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

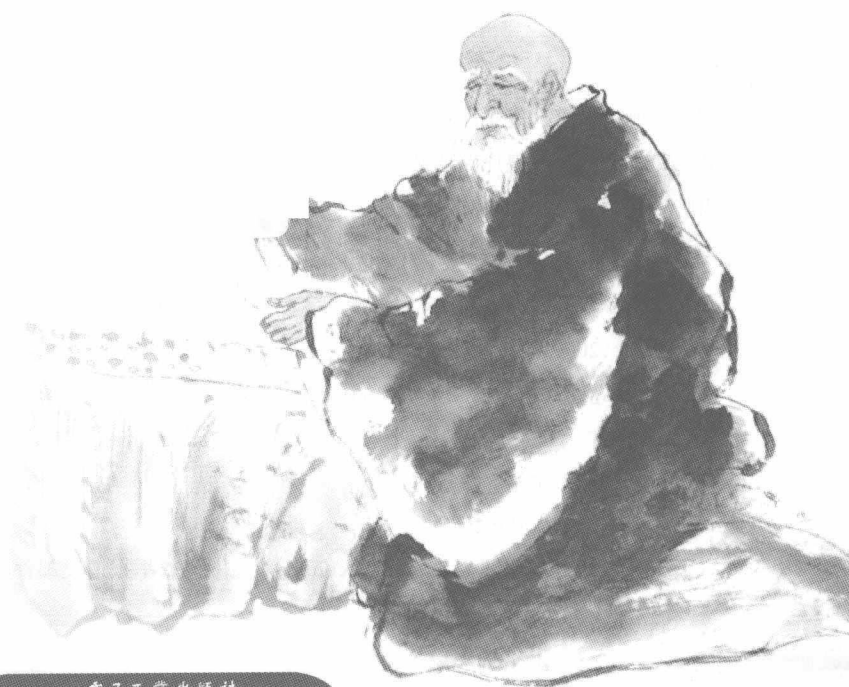
C/C++

开发专家

把VC++

Visual C++

白乔 左飞 编著
飞思科技产品研发中心 监制



电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内容简介

本书着重介绍 Visual C++ 的关键技术, 即 Visual C++ 程序员在实际工程开发中经常遇到的几大问题: 数据的内存表达、数据的转换与处理、数据的持久化、数据的界面展现、数据的传输与交换方面的技术。全书内容主要包括: 软件项目中的开发技术、关于 Visual C++ 的话题、常见对象的表达、对象的赋值与转换、对象容器、使用文件存取对象、使用注册表存取对象、使用数据库存取对象、界面的布局、界面的绘制与交互、界面的美化、进程间对象的传输与调用、线程间对象的传输与调用、模块间对象的传输与调用、网络间对象的传输与调用等。通过对本书的学习, 可以帮助读者快速掌握 Visual C++ 的编程技巧及编程思想。

随书所附光盘包含书中实例源文件。

本书面向初、中级读者, 适合于广大有志于学习 Visual C++ 的读者, 也适合作为高等院校相关专业师生的参考书, 还可作为相关培训机构的教材。

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

把脉 VC++ / 白乔, 左飞编著. —北京: 电子工业出版社, 2009.7

(C/C++ 开发专家)

ISBN 978-7-121-08617-5

I. 把… II. ①白…②左… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 049336 号

责任编辑: 杨 鹂

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

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: 787×1092 1/16 印张: 42.25 字数: 1081.6 千字

印 次: 2009 年 7 月第 1 次印刷

印 数: 5 000 册 定价: 69.50 元 (含光盘 1 张)

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 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。



又是一本 Visual C++ 图书?

Visual C++ 领域从来都不乏好书,在图书市场极其丰富的今天,再来撰写一本关于 Visual C++ 的图书是需要足够的胆量和理由的。

相信很多读者都和我一样,书架上早就摆满了各种 Visual C++ 书籍,有关于 Visual C++ 5.0 版、6.0 版的,还有直到当今的 2005 版、2008 版的,Visual C++ 技术的千头万绪包含其中,如 GDI、OLE、ActiveX、COM、XML、ODBC、ADO 等。为了学好 Visual C++,有些读者可能还有一些关于 C++ 的书籍,ООP、POD、STL、泛型编程、智能指针等术语弥漫其中。此外,Visual C++ 程序员又常常谈起 boost、DirectX、ICE 等技术,它们和 Visual C++ 家族之间又是如何沾亲带故的呢?

除了术语灾难之外,我们再来看看来自不胜枚举的项目案例的风暴。除了一些大部头的语法书之外,Visual C++ 程序员还会发现越来越多的好书,这些书不无详尽地枚举出各种案例,适合应用于专业领域,如图像处理、视频处理、串口通信等。更有甚者,赫然一下子摆出百八十个生猛鲜活的项目实例(编程百例等),指导 Visual C++ 程序员这样做、那样做。这样的例子越多,刚刚入门的程序员就会越困惑,明白在做什么,但并不知道为什么要这么做。当遇到一个新的项目的时候,我们还是会晕头转向,不知从何下手。

越来越多的人在踏入 Visual C++ 大门之后,开始左右彷徨,谁能引我真正入门?谁能拨去我程序员大军眼前的层层迷雾? Visual C++,神乎?魔乎?

是该到了为 Visual C++ 开发技术做个盘点的时候了!尽管显得有点自不量力,但我还是想做个尝试,撰写此稿,和大家一起努力去理一理 Visual C++ 开发技术的细枝末节(套用范伟在《卖车》小品中的话就是“有点乱,有点乱,我们得捋一捋,捋一捋”),并将多年开发的感触和盘托出,以飨读者。



把住 Visual C++ 编程脉络!

这本书与中医没有任何关系,Visual C++ 尽管毛病不少,但这本书并不是来讨论如何治病救人的。书名中的“把脉”,表达了我们的一种美好愿望,希望通过本书的讨论,帮助读者在实战项目之前,不再犯晕,能够火眼金睛,一下子把住 Visual C++ 编程技术的奇经八脉。

任何一本大部头的书都会在其大纲中详细列出 Visual C++ 的开发技术,但是很少有书将这些开发技术做过分类,指出各种技术的渊源所在,行文上也是照本宣科,将诸多知识点散得七零八落。结果读书的人也只好跟着来个“难得糊涂”,好不容易将书“啃”完了,如释重负之余却发现仍然是寸步难行。Visual C++ 的技术盘根错节,每一门技术会有更多、更细的实现方式,那能否有一张详尽的脉络图来理顺 Visual C++ 的脉络呢?还有,这些主脉技术与当今流行的软件分层模式之间又是个什么关系?冲突乎?重合乎?我们该如何把握?

本书中我们妄以蚍蜉撼树,将这些技术问题一一给出答案,并给出了一张详细的技术脉络图(见彩页)。希望读者朋友们不嫌粗糙,仔细审阅。

本书内容系统、完整,共分为 5 篇 15 章。

第 1 篇: Visual C++ 速览。讨论软件项目的开发模式及 Visual C++ 的四大开发技术,结

合 Visual C++ 的历史讨论 Visual C++ 的真实面目，以及它与其他技术之间的关系。

第 2 篇：Visual C++ 中对象的建模与表达。讨论如何采用标准 C++、MFC、ATL 来表达对象，以及对象之间的赋值与转换问题，并结合 STL 和 MFC 集合讨论对象容器的使用。

第 3 篇：Visual C++ 中对象的持久化。讨论如何将对象存储到文件、注册表、关系型数据库中，并读取出来。

第 4 篇：Visual C++ 中对象的展现。讨论如何在 Visual C++ 中设计界面，如何使用 GDI 和 GDI+ 绘制图形，以及如何利用 Windows 强大的消息机制实现界面的交互。

第 5 篇：Visual C++ 中对象的传输与调用。讨论如何在各种模块中传输和调用对象，让对象跨越包括进程、线程、DLL、ActiveX、COM 及网络等在内的多种边界。



开放的 Visual C++

很多 Visual C++ 程序员一直认为 Visual C++ 是封闭的，与世隔绝的。在 8 年以前，这样的看法基本上还算客观。但是 2002 年以来，随着微软坚持不懈的努力，这些成见似乎需要做一些改变。

不要再将 Visual C++ 与标准 C++ 对立起来，Visual C++ 2003 已经完成对标准 C++ 高达 98% 以上的兼容性支持。我们很容易在 Visual C++ 中使用到 STL 容器及泛型算法，甚至可以导入类似于 boost 标准库、Qt 界面库这些 C++ 开发库。

不要再以为 Visual C++ 只有 MFC。MFC 因为其高超的封装性，让那些产生了 MFC 依赖症的程序员由爱生恨，百般非议。微软由此提出了 ATL，将那些小型的、有用的类扔到 ATL 中，而不必再依赖于 MFC 环境。当程序员再去编写一个使用 CString 的程序，就不再需要 MFC。如果想将一堆 CString 扔进一个 map 当中，同样也可以不再使用 MFC。

Visual C++ 是开放的，它除了拥有全球庞大的开发者队伍以外，还拥有贴心的 MSDN 网络与 MSDN 库，拥有诸如 Visual Assist X 等不断涌出的 IDE 增强工具，拥有更多的开发者网站，拥有更多的开源项目和第三方库，如 DirectX、OpenGL、Xtreme Toolkit、BCGControlBar 等。Visual C++ 程序员有了越来越多的伙伴，Visual C++ 程序员永远都不会孤独。

本书介绍了 STL 算法，boost 库，图形相关的 OpenGL、DirectX、QT 技术，以及界面相关的 Xtreme Toolkit、BCGControlBar、SkinMagic 换肤技术等。这么做绝不是哗众取宠，也绝不是盲目跟风，在 Visual C++ 实作项目中总会遇到这些技术的影子，我们努力以最短的篇幅来讲解这些技术的使用方法，希望读者朋友们能够笑纳。



本书适合哪些读者

本书适合那些 Visual C++ 入门级读者，以及对 Visual C++ 有着多年实践经验、但仍为 Visual C++ 一些奇怪的行为感到疑惑的程序员。

尽管本书提到过很多术语，但本书并不介绍如下内容：

- 关于标准 C++ 基础语法的介绍。
- 关于 C++/CLI 基础语法的介绍。

- 关于 Visual Studio 基本操作的介绍，如：如何设定编辑器的字体、如何设计对话框等。
- 关于 CLR 编程技术的介绍，虽然本书在介绍某些技术点时提到了 CLR 编程技术，但是本书还是尽量只介绍传统 Visual C++。
- 关于 Windows 操作系统、.NET 平台知识的介绍。

总之，本书适合作为 Visual C++ 初学者的第二本书。假设你已经熟悉了 C++ 的基础语法，并至少根据某一本书，做过一个 Windows 版的“Hello, World!”程序。

在阅读过程中，本书总是忍不住列举一些包含 Java、Visual Basic、JavaScript 等语法的例子，注意这仅仅是作为技术讨论的需要，读者大可不必因为不了解这些语言（或脚本语言）而担心理解上的困难。



图例与样式



开始动手

结合知识点的介绍开始一个具体的项目任务。



提示

对前文的一个备注，或者引发读者进一步思考。用语较口语化。



MSDN 参考

给出本节相关内容在 MSDN 文档库中的索引关键字。



光盘导读

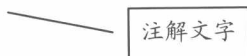
指明项目代码对应于光盘中的目录。



参数解释

指明函数参数的含义。

冒泡注解



添加在图片或者代码中的注解文字，帮助读者理解。

操作流程

细分操作流程，帮助读者尽快上手。

加粗文字

总结性的、较为重要的文字内容。



代码与光盘

光盘中收录了所有的实例代码。在每一节的实例中，我们都会通过“光盘导读”指出该实例对应于光盘的位置。

在笔者的机器上，安装有 Visual C++ 6.0、2005、2008 三个版本，它们一直相安无事很多年。撰写本书的示例项目时，笔者采用 Visual C++ 2005 而非 Visual C++ 2008 作为 IDE。其原因在于，Visual C++ 2008 的使用尚未普及，而采用 Visual C++ 2005 生成的程序代码也可以在 Visual C++ 2008 中自动升级转换（反过来则不行）。

本书所附光盘中的源代码皆以解决方案组织，解决方案名对应于每章，如 ch02（第 2 章）。在解决方案下会放置该章下对应的项目，如 stdcpp。我们在描述这些项目路径时，会表示成“\ch02\stdcpp”。



结语

本书由白乔、左飞编著，何洪波、沈良忠参与了部分章节的编写，水木清华“Visual C++”版块前任版主 hamxj，“嵌入式系统”版块前任版主 Emb，来自 vcer.net 的 jerry lucky 和郭克新先生，以及来自中国台北的 gerald，作为本书的第一批读者，就技术讨论和行文思路提出了非常中肯的意见，挚友杨汉玮对全书进行了文字润色，在此一并感谢！成书之际，特意感谢易飞思公司的策划张春雨先生，他支持着我对写作思路做的屡屡调整，并为此书逐步推敲出合适的名字。最后，感谢我的妻子，她让我明白在程序员的世界里，除了漫天飞舞的代码，还有登山观海、驱车听乐，以及初为人父的欣喜。在此，我也真心地祝福其他同行，除了以单调的 Visual C++ 为伴，也能够享受到更加多彩的人生。

Visual C++ 的胸怀是宽广的，Windows 及 .NET 的世界是浩瀚的，撰稿之时，笔者深惜个人能力有限，在某些地方甚至无法深入，纰漏和欠缺之处在所难免，特别是关于 Visual C++ 庐山面目、COM 组件编程等开放性的话题，言语之中有失偏颇之处，还望读者不吝赐教和批评。联系信箱：bluejoe2008@gmail.com，详情请垂询 <http://books.vcer.net/vcmap>。我们相信，通过你我的共同努力，我们会成就一本真正有用的好书！

白乔

联系方式

咨询电话：(010) 68134545 88254160

电子邮件：support@fecit.com.cn

服务网址：<http://www.fecit.com.cn> <http://www.fecit.net>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

第 1 篇 Visual C++速览

第 1 章 软件项目中的开发技术	3
1.1 关于软件项目的话题	3
1.2 软件项目的开发过程	3
1.2.1 软件的开发过程	3
1.2.2 排错 (debug)	6
1.2.3 原型 (prototype)	6
1.2.4 重构 (refactor)	7
1.3 软件架构及软件分层	8
1.3.1 软件架构的原则	8
1.3.2 C/S 与 B/S 模式	9
1.3.3 MVC 模式	11
1.3.4 文档/视图结构	12
1.3.5 流行的三层结构	12
1.4 在 Visual C++项目中的 四大主脉技术	13
1.4.1 对象的建模与表达	14
1.4.2 对象的持久化	14
1.4.3 对象的界面展现	15
1.4.4 对象的传输与调用	16
1.4.5 从 QQ 和 MSN 说起	17
1.4.6 四大技术与三层结构	23
1.5 回到客栈管理系统	24
1.5.1 对象的建模与表达	24
1.5.2 对象的持久化	26
1.5.3 对象的界面展现	28
1.5.4 对象的传输与调用	29
1.6 本章小结	33
第 2 章 关于 Visual C++的话题	35
2.1 Visual C++与 C++	35
2.1.1 Visual C++不是唯一的 C++编译器	35
2.1.2 Visual C++不是唯一的 C++ IDE	37
2.1.3 Visual C++及 MFC/ATL 的版本	39

2.1.4 备受争议的 Visual C++	40
2.1.5 Visual C++与 boost、QT	42
2.2 熟悉 Visual C++ IDE	43
2.2.1 文件、项目与解决方案	44
2.2.2 Visual C++代码编辑器	46
2.2.3 Visual C++资源编辑器	47
2.2.4 Visual C++应用向导	49
2.2.5 Visual C++调试器	50
2.3 脱离 Visual C++编写程序	54
2.3.1 Dev-C++	54
2.3.2 控制台程序的编写	55
2.3.3 Windows 程序的编写	56
2.4 使用 Visual C++编写程序	61
2.4.1 Win32 控制台程序的编写	61
2.4.2 让控制台程序支持 MFC/ATL	62
2.4.3 Win32 窗口程序的编写	65
2.4.4 MFC/ATL 应用程序的编写	71
2.4.5 CLR 程序的编写	78
2.5 茴字的 N 种写法	79
2.5.1 使用 Windows API	80
2.5.2 使用 C++标准库 (stdc++	83
2.5.3 使用 CRT (C 运行时期库)	84
2.5.4 使用 CRT 库的宽字符版本	87
2.5.5 使用 CRT 库的安全版本	88
2.5.6 使用 MFC/ATL	89
2.5.7 使用 C++/CLI	91
2.5.8 该采用哪一种写法	92
2.6 Visual C++伴侣	93
2.6.1 源代码	93
2.6.2 MSDN 与 MSDN 库	95
2.6.3 Visual Studio Tools	97
2.6.4 Visual Assist X	98
2.6.5 Visual C++资源网站	99
2.7 本章小结	101

第 2 篇 Visual C++ 中对象的建模与表达

第 3 章 常见对象的表达	105
3.1 C++ 基本类型	105
3.1.1 原始类型	105
3.1.2 复合类型	106
3.1.3 自定义类型	108
3.1.4 __intN	110
3.1.5 <climits>	110
3.2 字符与字符串	111
3.2.1 <cctype>	112
3.2.2 C-style 字符串	114
3.2.3 Windows 字符类型	115
3.2.4 <cstring>	115
3.2.5 std::string	117
3.2.6 ATL::CString	120
3.3 日期与时间	122
3.3.1 <ctime>	123
3.3.2 ATL::CTime 与 ATL::	
CTimeSpan	124
3.3.3 ATL::COleDateTime 与	
ATL::COleDateTimeSpan	127
3.3.4 时间的格式化	127
3.4 位置与尺寸	129
3.4.1 POINT、SIZE、RECT	129
3.4.2 ATL::CPoint、ATL::CSize、	
ATL::CRect	130
3.5 智能指针 (smart pointer) ...	131
3.5.1 std::auto_ptr	131
3.5.2 ATL::CComPtr 与	
ATL::CComQIPtr	133
3.6 Windows 数据类型	134
3.6.1 基本数据类型	134
3.6.2 Windows 句柄	135
3.7 MFC 对象模型	138
3.7.1 MFC::CObject	139
3.7.2 CObject 与 RTTI	142
3.7.3 CRuntimeClass	145

3.7.4 百般非议 MFC	148
3.8 MFC 与 ATL	148
3.9 命名规则与名字危机	149
3.10 本章小结	151
第 4 章 对象的赋值与转换	153
4.1 对象的初始化与赋值	153
4.1.1 struct 与 class	153
4.1.2 对象的初始化	154
4.1.3 聚合 (aggregate)	154
4.1.4 对象的赋值	155
4.1.5 sizeof 及字节对齐	155
4.1.6 内存块操作函数 memxxx	157
4.1.7 POD 对象	158
4.2 对象的类型转换 (cast)	160
4.2.1 隐式转换	160
4.2.2 显式转换	161
4.2.3 向上转换与向下转换	163
4.2.4 变体 VARIANT	163
4.2.5 _variant_t、CComVariant 与	
ColeVariant、CDBVariant	166
4.3 对象的数值转换	
(conversion)	168
4.3.1 xtoy	168
4.3.2 sprintf、sscanf	168
4.3.3 std::stringstream	170
4.4 数学运算	171
4.4.1 <cmath>	171
4.4.2 随机数	173
4.4.3 复数 std::complex	173
4.5 字符与编码	174
4.5.1 字节与字符	175
4.5.2 ANSI、MBCS 与	
UNICODE	175
4.5.3 TCHAR、_T 与 _TEXT	177
4.5.4 strcpy、wcscpy 与 _tscpy	178
4.5.5 MessageBoxA、MessageBoxW	
与 MessageBox	179
4.5.6 ATL::CStringT 与 std::basic	
_string	179

4.5.7	BSTR、_bstr_t 与 ATL::CComBSTR	180
4.5.8	字符串的长度	181
4.5.9	字符的编码转换	183
4.6	使用第三方库	187
4.6.1	万能类型 boost::any	187
4.6.2	万能转换器 boost::lexical_cast	189
4.6.3	有理数 boost::rational	191
4.7	本章小结	192
第 5 章	对象容器	193
5.1	常见容器	193
5.1.1	向量 (vector)	193
5.1.2	列表 (list)	194
5.1.3	映射 (map)	195
5.1.4	集合 (set)	195
5.1.5	队列 (queue)	196
5.1.6	栈 (stack)	196
5.1.7	选择合适的集合	197
5.2	C++标准容器	198
5.2.1	std::vector	199
5.2.2	std::list	200
5.2.3	std::deque	200
5.2.4	std::map、std::multimap	201
5.2.5	std::set、std::multiset	202
5.2.6	std::bitset	203
5.2.7	std::queue、std::priority _queue	204
5.2.8	std::stack	205
5.3	泛型算法	206
5.3.1	指示器	206
5.3.2	函数对象	207
5.3.3	泛型算法介绍	210
5.4	MFC/ATL 容器	215
5.4.1	MFC::CArray	217
5.4.2	MFC::CList	218
5.4.3	MFC::CMap	218
5.4.4	集合元素的遍历	219
5.5	本章小结	221

第 3 篇	Visual C++中对象的持久化
第 6 章	使用文件存取对象
6.1	文件操作的几种方式
6.1.1	使用 CRT 函数 std::fxxx()
6.1.2	使用标准 C++库 std::fstream
6.1.3	使用 Windows API
6.1.4	使用 MFC::CFile
6.1.5	文本文件与二进制文件
6.2	使用文本文件存取对象
6.2.1	文本内容的存取
6.2.2	properties 文件的读取
6.2.3	INI 文件的读取
6.2.4	CSV 文件的读取
6.3	使用二进制文件存取对象
6.3.1	文件指针的定位
6.3.2	定长数据块的读写
6.3.3	变长数据块的读写
6.3.4	使用 MFC::CArchive
6.3.5	Windows 资源文件的读取
6.4	使用 XML 存取对象
6.4.1	XML 及其语法
6.4.2	XML 的解析方式
6.4.3	使用 MsXML 存取 XML
6.4.4	使用 ADO.NET 存取 XML
6.5	本章小结
第 7 章	使用注册表存取对象
7.1	注册表及其结构
7.1.1	注册表的结构
7.1.2	常见的注册表根键
7.2	使用注册表 API
7.2.1	关于注册表操作的 API
7.2.2	使用 Windows API 写注册表
7.2.3	使用 Windows API 读注册表
7.3	使用 CRegKey 存取注册表
7.3.1	ATL::CRegKey

7.3.2	使用 CRegKey 写注册表	287	9.1.5	位图与图标 (Bitmap、Icon)	361
7.3.3	使用 CRegKey 读注册表	289	9.2	界面的静态布局	363
7.4	本章小结	292	9.2.1	使用对话框	364
第 8 章	使用数据库存取对象	293	9.2.2	使用控件	368
8.1	数据库概述	293	9.2.3	动态创建对话框和控件	369
8.1.1	关系型数据库	293	9.2.4	使用代码控制控件	371
8.1.2	SQL 语句	295	9.2.5	映射控件变量	373
8.1.3	常见数据库访问技术	296	9.2.6	使用容器放置控件	379
8.2	使用 ODBC 操纵数据库	297	9.2.7	通用对话框	383
8.2.1	ODBC	297	9.2.8	使用 Windows Forms	386
8.2.2	使用 ODBC API	298	9.3	界面的动态布局	390
8.2.3	使用 MFC::CDatabase	302	9.3.1	框架窗口	390
8.2.4	封装自己的 Recordset	304	9.3.2	文档/视图结构	394
8.3	使用 DAO 操纵数据库	309	9.3.3	文档与文档模板	396
8.4	使用 OLE DB 操纵数据库	313	9.3.4	视图与 CView	398
8.4.1	OLE DB 的结构	313	9.3.5	框架窗口都忙些什么	399
8.4.2	OLE DB 使用者模板	314	9.3.6	拆分窗口	401
8.4.3	使用 OLE DB 访问数据库	315	9.4	系统托盘	405
8.4.4	使用 ATL 向导创建 OLE DB 使用者	318	9.5	本章小结	408
8.5	使用 ADO 操纵数据库	321	第 10 章	界面的绘制与交互	409
8.5.1	ADO 是对象封装接口	322	10.1	窗口与绘图	409
8.5.2	常见数据源的连接串	322	10.1.1	在 DOS 下画一个圈	409
8.5.3	使用 ADODB 组件	323	10.1.2	在 Windows 下画一个圈	411
8.5.4	使用 ADODC 控件	326	10.1.3	使用 GDI 绘制的假按钮	414
8.6	使用 ADO.NET 操纵数据库	330	10.1.4	创建真正的按钮	415
8.6.1	ADO.NET 与 ADO	330	10.1.5	使用 GDI+绘图	417
8.6.2	ADO.NET 的联机模式	330	10.2	让界面交互起来	420
8.6.3	ADO.NET 的脱机模式	332	10.2.1	一个假的控制台 消息系统	420
8.7	本章小结	337	10.2.2	真正的 Windows 消息系统	425
第 4 篇	Visual C++ 中对象的 界面展现		10.2.3	鼠标的 3 次不同点击	428
第 9 章	界面的布局	341	10.3	MFC 对消息机 制做了什么?	430
9.1	常见界面元素	341	10.3.1	消息映射 (Message Mapping)	431
9.1.1	窗口 (Window)	341	10.3.2	消息映射宏	435
9.1.2	菜单 (Menu)	346			
9.1.3	控制栏 (Control Bar)	349			
9.1.4	客户区与非客户区	358			

10.3.3 消息路由 (Message Route)	436
10.3.4 消息反射 (Message Reflection)	440
10.3.5 消息泵 (Message Pump)	445
10.4 使用第三方库	447
10.4.1 DirectX	447
10.4.2 OpenGL	450
10.4.3 Qt	452
10.5 本章小结	454
第 11 章 界面的美化	457
11.1 界面美化的基本途径	457
11.1.1 设定窗口样式 (Window Style)	458
11.1.2 调用函数设定控件的 扩展样式	462
11.1.3 调用函数设定颜色	464
11.1.4 调用函数设定背景图	465
11.2 重载消息进行界面自绘	466
11.2.1 WM_PAINT	466
11.2.2 OnDraw()与 OnPaint()	468
11.2.3 WM_ERASEBKGDND	468
11.2.4 =WM_CTLCOLOR	470
11.2.5 =WM_DRAWITEM	472
11.2.6 =WM_MEASUREITEM	476
11.2.7 =NM_CUSTOMDRAW	479
11.3 使用 ActiveX 控件 增强界面	482
11.3.1 ActiveX 控件的注册	483
11.3.2 在对话框中引入控件	484
11.3.3 将控件映射成变量	486
11.3.4 常见 ActiveX 控件推荐	488
11.4 使用第三方库	491
11.4.1 Xtreme Toolkit 与 BCGControlBar	492
11.4.2 SkinMagic 与换肤技术	494
11.5 本章小结	497

第 5 篇 Visual C++ 中对象的 传输与调用

第 12 章 进程间对象的传输与调用	501
12.1 进程的操作	501
12.1.1 进程及其常见操作	501
12.1.2 关于进程操作的 CRT 函数	502
12.1.3 关于进程操作的 Windows API	503
12.1.4 Windows 进程的创建	504
12.1.5 Windows 进程的终止	508
12.2 进程间的通信	510
12.2.1 使用 Windows 消息 传输对象	510
12.2.2 使用文件映射共享数据	512
12.2.3 使用剪贴板传输对象	518
12.2.4 使用匿名管道传输对象	521
12.2.5 使用命名管道传输对象	526
12.2.6 使用邮槽传输对象	530
12.3 本章小结	533
第 13 章 线程间对象的传输与调用	535
13.1 线程的操作	535
13.1.1 关于线程操作的 CRT 函数	535
13.1.2 关于线程操作的 Windows API	536
13.2 MFC 对线程操作的封装	536
13.2.1 工作者线程	537
13.2.2 带界面的工作者线程	539
13.2.3 界面线程	541
13.3 线程通信	543
13.3.1 利用全局对象进行通信	543
13.3.2 利用入口参数进行通信	545
13.3.3 利用 Windows 消息 进行通信	547
13.4 线程同步	549
13.4.1 编写安全的售票程序	550
13.4.2 同步对象 CSyncObject	553

13.4.3	利用临界区 (Critical Section) 实现同步	554	14.3.6	使用 COM API 调用 COM 组件	616
13.4.4	利用互斥体 (Mutex) 实现同步	556	14.3.7	使用 #import 方式调用 COM 组件	619
13.4.5	利用信号量 (Semaphore) 实现同步	558	14.4	本章小结	621
13.4.6	利用事件 (Event) 实现同步	559	第 15 章 网络间对象的传输与调用	623	
13.4.7	CMultiLock 和 CSingleLock	563	15.1	使用 Socket 完成网络通信	623
13.5	本章小结	566	15.1.1	如何向网络上发送 一个比特?	623
第 14 章 模块间对象的传输与调用	569		15.1.2	TCP/IP 协议与 TCP、UDP	624
14.1	静态库与动态链接库	569	15.1.3	Winsock 与 Winsock API	625
14.1.1	将 mcd 包装成 mcd.lib	569	15.1.4	用 Socket 实现 C/S 模式	626
14.1.2	将 mcd 包装成 mcd.dll	571	15.1.5	使用 Winsock 打造 mcd 的 网络版	627
14.1.3	mcd.dll 的调用	574	15.2	使用 MFC::Csocket 完成网络通信	632
14.1.4	显式调用与隐式调用	576	15.2.1	Winsock 与 CAsyncSocket、 CSocket	632
14.1.5	将类封装成 DLL	578	15.2.2	使用 CSocket 打造 mcd 的 网络版	634
14.2	ActiveX 控件	581	15.2.3	使用 CSocket 打造 mcd 的 UDP 版	638
14.2.1	将 mcd 包装成 mcd.ocx	581	15.2.4	使用 CSocket 事件	640
14.2.2	为 mcd.ocx 添加 SetNumbers 方法	586	15.2.5	使用 CSocket 连接 Web 站点	648
14.2.3	为 mcd.ocx 添加 Direction 属性	589	15.3	使用 WinInet API 完成 网络通信	653
14.2.4	在 HTML 中测试 McdOcx 控件	593	15.3.1	WinInet API 与 WinInet 类	653
14.3	COM	596	15.3.2	使用 WinInet API 连接 mcd 的 web 版	654
14.3.1	一个不是 COM 的 C++ 接口	596	15.3.3	使用 MFC::WinInet 类连接客栈的 Web 版	658
14.3.2	一个貌似 COM 的 DLL	601	15.4	本章小结	663
14.3.3	真假 COM	609			
14.3.4	使用 ATL 设计 COM 组件	611			
14.3.5	在 JavaScript 中测试 COM 组件	615			

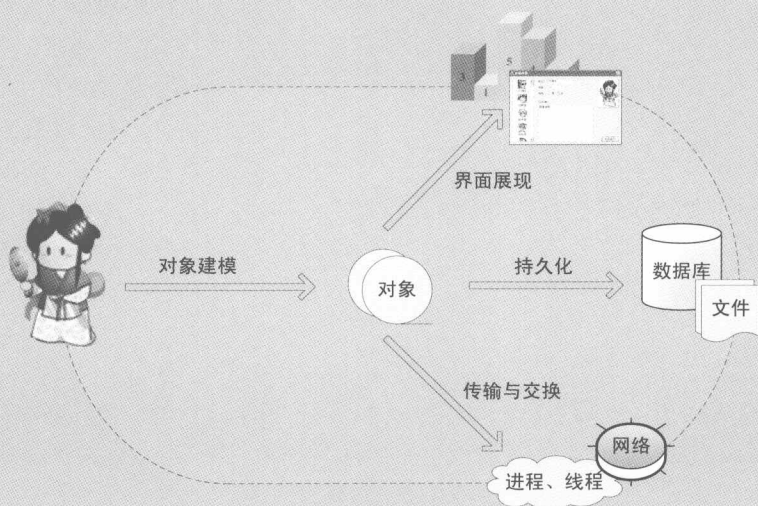


Visual C++速览

相对于 Visual Basic、Visual J++ 这些语言来说, Visual C++ 绝对占有一席之地, 它是 Visual Studio 中最为浓墨重彩的一笔 (至少在 C# 出现之前是这样的)。那么, 既然有了 C++, 为什么我们还需要 Visual C++? Visual C++ 的真实身份是什么? 编程语言乎? 集成开发环境乎? 应用程序框架乎?

我们常常自诩为 Visual C++ 程序员, 并宣称我们通过使用 Visual C++ 开发了这些、那些的工程项目 (project)。那么, 软件项目的开发通常都包含哪些内容呢? 它的开发过程又是怎样的呢? 再者, 就像 20 世纪六七十年代“模块化设计”的概念风靡软件界一样, 近年来由于 Web 应用的日益普及, “软件分层”的概念又一次被提出并迅速成为一种时尚, 这对我们开发 Visual C++ 项目又意味着什么? 鸡肋乎? 融合乎?

本篇我们来讨论软件项目的开发过程, 以及流行的分层架构模式, 并最终结合软件项目讨论 Visual C++ 中四大堪称主脉的关键技术, 以及与分层架构的呼应。在第二章, 我们再花费一章的篇幅八卦一下与 Visual C++ 相关的话题。通过这些话题的讨论, 读者朋友们应该很清楚 Visual C++ 是什么, 不是什么? 有了 Visual C++ 该怎么编程, 没有 Visual C++ 又该怎么编程? 如此等等。



第 1 章 软件项目中的开发技术

在接触到本书之前，相信有很多读者了解到 Visual C++ 中的各种编程技巧，比如：窗口的显示、菜单的设计、鼠标事件的捕捉、文件的读写等。那么如此之多的编程技术，在实际的项目开发中是什么样的组合形式呢？它们之间又是什么关系呢？本章我们讨论一下软件的开发过程及流行的软件架构技术，并结合 Visual C++ 的技术点谈及本书的主题，即 Visual C++ 的四大主脉技术。

1.1 关于软件项目的话题

本书的构思之际，正逢尚敬执导的电视剧《武林外传》热播之时，和很多在生活中需要笑声的程序员读者一样，笔者一直是《武林外传》的忠实观众，因此本书关于软件项目的话题，我总忍不住要从如下假设的情境开始。

设想一下，某个阳光灿烂的星期一上午，你的老板（Boss、导师，或者公司的领导）找到睡眼惺忪的你，让你准备开发一套系统：

“小六，我们需要为娄知县开发一套“七侠镇客栈管理系统”……”

没错，假设你就是那个备受老板青睐的燕小六，而且你是七侠镇远近闻名的一名 Visual C++ 程序员，据称你熟悉 Visual C++ 6.0、Visual C++ 2003，甚至 2005 和 2008。以下可能是老板（有可能就是邢捕头）继续的功能描述：

“就是提供一套窗口界面，每个客栈的掌柜都可以方便地进行员工信息的编辑和浏览……可以插入照片和一些文档附件，最好能够直接将文件拖曳进去，必要的时候需要生成 Word 报表并进行打印，还有一点，稍微复杂一些，每个掌柜使用系统之前都需要进行认证，这个认证模块已经由另外一个公司做了，是六扇门软件公司的指纹认证 LsmAuthentication 系统，对了，还需要及时向衙门上报每天的住宿情况，衙门有一套基于 J2EE 环境的 QiXiaOnLine 系统，提供有信息接收的 Web Service 接口……”

实际上，有很多的软件项目就是从这样的星期一开始的，下一步该怎么走？我们该如何继续？接下来我们就来讨论软件项目所涉及到的开发过程及开发技术。

1.2 软件项目的开发过程

软件项目的开发是有章可循的，本节我们讨论一下软件在实际开发过程中的几大步骤，以及程序员在此间可能会遇到的技术问题。

1.2.1 软件的开发过程

软件的开发向来都不仅仅只是编写代码。实际上，软件开发是一件很严谨的事情，据说需要完整的、系统的软件工程理论知识；另一方面，对于软件开发，每个人也会有自己钟爱的开发方法和习惯（正如有的人偏爱用左手写字一样）。在此并不想花多大的篇幅讲述

令人头疼的软件项目、项目管理的概念，我们尽可能以简单朴实的阐述方式与读者们探讨一种实际可行的流程。

按照传统的观点，完整的软件开发至少包括以下几个阶段。

1. 项目启动

启动一个软件项目，包括问题定义、可行性分析和制订计划。当然这件事一般是你的老板去做，他准备启动，然后作为软件设计师的你，必须配合其做一些包括进度计划的方案策划。做任何事情之前必须有个完整的计划（不要老提“计划赶不上变化”），不要将你的项目置于无序、失去控制的状态中。

当然，没有一成不变的计划，在软件的后续开发过程中，以上制定的计划内容和开发进度会根据实际情况进行调整和细化。

在项目启动阶段，最常用的工具就是 Word（它用来编写项目合同甚至实施方案书），或者 PPT（Microsoft Power Point，PPT 常被 IT 界戏称为“骗骗他”）。在制定计划和资源分配的时候，我们也有可能用到其他的工具，如：Microsoft Project。

2. 需求分析

在需求分析阶段，我们需要确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、清晰、具体的要求。

对软件项目来说，需求分析至关重要。如图 1-1 所示的是在互联网上曾经流行的一个富有讽刺意义的软件开发过程：一个软件项目通过客户对需求的描述，到项目经理的理解、分析师的设计、程序员的编写、客户的合力建造、糟糕的技术支持，最终造成了一个客户完全不需要的东西。在这整个过程中，需求分析即是第一步！由此可见项目人员与客户之间的沟通、对需求的理解与整理非常重要，它是决定软件项目是否成功的关键一步。

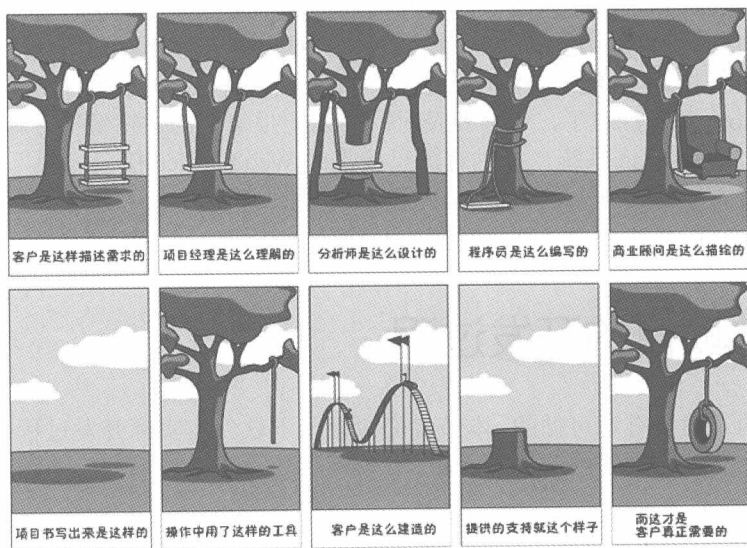


图 1-1 客户的需求与最后的项目产品

在需求阶段中，我们需要通过各种途径与客户进行沟通（尽管“沟通”这个词语现在已基本上用滥了，但它确实是不可缺少的一项工作），完成一份双方确认的（有可能是多级