



——一个Visual C++老鸟 10年学习与开发心得



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

九江学院图书馆



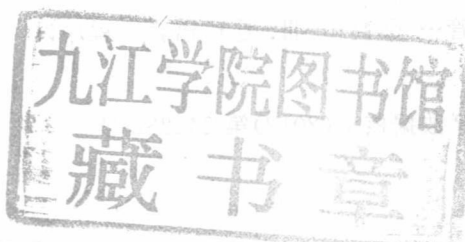
1563645

1623112

别样诠释

——一个 Visual C++ 老鸟 10 年学习与开发心得

管 皓 高永丽 编著



2012/23536



北京航空航天大学出版社

5118231

内 容 简 介

本书着眼于实际应用,循序渐进地介绍了 Visual C++ 的编程原理与技巧;着重于信息的整体处理流程,从信息获取、信息传输、信息存储、信息展现等方面构筑了 Visual C++ 的全新学习、认知架构。

本书与市面上绝大多数的 Visual C++ 不同,并没有进行技术知识点的简单罗列,而是从信息处理流程的角度出发,为读者构建一个完整而清晰的 Visual C++ 解决方案。书中包含了较为丰富的示例,讲解细致,有助于读者更快、更好、更深入地掌握 Visual C++ 编程技术。

本书适合于 Visual C++ 初学者及有一定基础的程序员,对于高校学生、研究生及科研、工程人员具有很好的学习参考价值。

图书在版编目(CIP)数据

别样诠释:一个 Visual C++ 老鸟 10 年学习与开发心得 / 管皓, 高永丽编著. -- 北京: 北京航空航天大学出版社, 2012. 11

ISBN 978-7-5124-0986-6

I. ①别… II. ①管… ②高… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 242236 号

版权所有,侵权必究。

别样诠释

——一个 Visual C++ 老鸟 10 年学习与开发心得

管 皓 高永丽 编著

责任编辑 刘亚军

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱: goodtextbook@126.com 邮购电话:(010)82316936

北京时代华都印刷有限公司印装 各地书店经销

*

开本:787×1 092 1/16 印张:16 字数:430 千字

2012 年 11 月第 1 版 2012 年 11 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-0986-6 定价:29.90 元

给你不一样的 Visual C++

首先感谢亲爱的读者朋友们选择这本书,我想本书不会辜负你的信任。大家手里拿的这本书,是我用心写就的。

自大学一年级上半个学期开始学习 C 语言到现在,近十个年头过去了。在这十年间自己在用软件、学软件再到编软件的过程中可谓经历颇丰、体味良多,既有失败的沮丧,也有成功的喜悦。在不断的积累中,我对于软件开发技术的认识在不断加深,同时学习技术本身也在影响着自己的性格。随着年龄的增长,我发现计算机软件技术越来越像一个始终陪伴自己的朋友,在一起时感到亲切温暖,而不是刚接触它时仅仅把它当成一种完成项目和毕业论文的工具。现在,我想把自己在接触 Visual C++ 过程中的所见、所学、所想、所感拿出来与大家分享,希望对于大家的日常学习、毕业设计、工作能起到良好的帮助作用。

我本科所学专业并非计算机,而是自动控制。在中国自动控制专业是“万金油”,学机械的、电子的、计算机的都可以从自己的技术角度来实现自动化。我们本科期间所学的东西真可谓是“大杂烩”——从硬件到软件啥都接触点儿,但是真学下来却啥也不精通。直到大学毕业,我也说不清自己的专业到底是干啥的,哈哈。幸好自己始终保留着对于软件的兴趣,这样到读研究生时就不再“一锅端”了,而是一门心思钻研自己喜欢的软件技术。其实对于技术的“专”与“博”的把握一直是个很有学问的事情。每个人的时间与精力有限,掌握一门技术首先要专注,也就是要做到“精通”,这是安身立命的基础。在精通了一门技术之后,也要放开视野,多去看看其他的技术门类(这里指软件技术范畴内),但不用太细,只求领会个大体的内容即可。这样日积月累,就可以不断提高自己对于计算机技术的认识。我自己在大学时学了很多计算机专业的同学不学的电气、电子、机械等硬件知识,尽管当时不太喜欢,但后来我发现这些东西对于自己认识软件起了很好的促进作用。

以上说了些貌似与本书关系不大的东西,但我想这些对于读者朋友是有用的。我不希望自己的这本书像市面上的一些教材一样板起面孔教导人,而是希望能通过拉家常的方式达到与读者分享技术经验的目的。因此,书中难免有不够严谨之处,大家尽管一笑了之吧。

好了,还是来说说本书的主角——Visual C++ 吧。

Visual C++ 简称 VC,对于这个简称真是让人喜爱万分。首先,维生素 C(Vitamin C)也是这个简称,看看,多有营养啊,如果你告诉别人你在学 VC,人家外行人有可能以为你钻研营养学呢;倘若你是个创业者,那么对 VC 这个词一定再熟悉向往不过了,因为它代表风险投资(Venture Capital),呵呵,多有含金量啊。有了这么促人奋进的名字(又有营养又有钱),相信我们学习 Visual C++ 一定是动力十足的吧。

我第一次集中研究 VC 是在大四的时候,因为毕业设计的缘故,我要做一个有关无人机上的材料耐压检测的软件。大体上就是由一个单片机读出某电压信号,然后将它通过串口通信的方式发送给 PC 上位机,由 PC 上位机进行处理与显示。功能其实不复杂,核心的东西其实就在于 VC 串口通信编程技术。如今看来小儿科的东西,当时可真是了命了。VC 本来就够不好学的

了,还牵扯到串口通信上的一些什么 VARIANT,ActiveX,COM,当时弄得我是迷迷糊糊。好在经过一阵努力,一知半解地将功能实现,顺利完成了任务。后来随着经验的日积月累,自己的技术实力逐渐提升,许多搞不明白的东西逐渐清晰起来。在这里要与读者分享的几条经验是:

① 务必先学扎实 C++。学 VC 主要是学习 MFC(Win32 API 开发不是一般开发人员能够驾驭的,当然它是基础)。MFC 给我们的是一个类库,其实就是一个一个现成的类,里面到处可见 C++ 语言的各种知识点。只要 C++ 扎实(有 C 语言的基础,认认真真地学上几个月 C++ 就应该算基本入门了),再熟悉 MFC 本身所具有的一些特色的东西,就能较快掌握 VC 开发。

② 软件开发背后涉及许多其他方面的知识,如操作系统、编译、网络、数据库、软件工程等。计算机科班的与其他专业的开发者往往在这方面差距明显。所以,如果你不是科班出身,多在这些方面下点功夫,对于软件开发功力的提升是很有帮助的。比如说,我学了面向对象、软件工程后,自然就接触了微软公司的 COM 技术,然后是 ActiveX,这样很自然就明白了许多知识。

VC 的学习曲线确实很陡,入门并不容易。许多入门级的开发者,往往被所谓的困难吓住了,转而投学其他更容易的开发工具了。VB 很简单易学,升级到 .NET 后功能更强了,但是我感觉它更像一种玩具语言,总让人感觉“隔了一层”,或许我本科有硬件的背景,所以喜欢底层一点的东西。C# 当然好,是 .NET 中的主角语言,但它更适合用在 Web 开发上,而且我不喜欢那份“臃肿感”。

VC 的好处就是能充分满足你的“控制欲”,因为它接近底层(别拿汇编和我抬杠啊,况且 VC 可以嵌入汇编嘛),速度快是非常明显的优势。看看现在异常火爆的网络游戏,其引擎有几个不是用 C/C++ 开发的(尽管可能是其他集成开发环境,但 VC 是 C/C++ 最主流的开发环境则毋庸置疑。对于那些常和硬件打交道的开发者,VC 是非常好的选择。我在研究所里做软件,感觉 VC 真的是简洁高效。

其实,VC 学起来也并非有多难,主要是许多人 C++ 学得夹生,再学 VC 时就拙襟见肘了。另外,市面上许多讲 VC 的书太千篇一律了,自己平时逛书店想淘本好书好好享受一下,可翻看各种 VC 书籍,不是什么《15 天学会 VC》就是什么《从入门到精通》,要不就是什么技巧大全。15 天学会绝对是噱头,150 天还差不多,学习不能急功近利;所谓“知识大全”类的只是技巧罗列,并没有一个好的体系将初学者或刚入门者的技术引领提升。我写这本书的目的,就是想针对市场上 VC 教材的缺陷,根据自己的体会,以全新的思路来教会迫切想学好 VC 的朋友们,之所以取名为《别样诠释》,就是想突出“别样”两字,通过比较新颖的方式,让读者朋友尽快学好 VC。倘能达到这个目的,我将感到非常满足与开心。

读者对象

如果你从未接触过编程,那本书会告诉你一个学习的脉络,照书中的指引去有意识地补习将会很快入门。希望你学过 C 语言,学过 C++ 语言,对于面向对象(不是面向你女/男朋友😊)编程有基本的认识。当然,如果你已经比较熟练地掌握了 C/C++ 语言(比如你累计学了一年),那当然更好。如果你也曾用过 VC“弹出”过几个对话框,但苦于不能更进一步提升 VC 技术,那么这本书将非常适合你。

本书特点

本书不同于市面上大多数的 VC 教材,采用全新的思路去引领读者学习好 Visual C++ 编程。主要体现在以下几方面:

① 不提倡什么“两周精通”之类的噱头词汇,因为那样往往是误导读者。技术学习的本质其实是一个不断积累的过程。在掌握了 VC 的基本编程结构与方式后,剩下的就是对于类库、API 函数的不断学习与积累。积累的时间越长,你的技术一般也越好。所以从这个意义上说,软件行业跟其他行业一样,越有经验越吃香。说什么软件是年轻人的专属,太可笑了,你调查调查去,一个有十年工作经验的架构师比一个只有一两年工作经验的“码工”身价高多少倍。所谓年轻之说,不过是告诉你,始终在一线开发会很累,但如果你真像微软公司里那些技术迷们爱着软件开发,就算编码到 50 岁退休也不嫌累;如果你不想一直编码,就向着设计师、架构师、CTO 发展啊,这些都需要经验啊。经验何来,积累!

② 学 VC 的本质是对 MFC 类库及 API 函数的掌握。用一个挺时髦的词儿来说,VC 是由不同的、十分有用的类组成的“解决方案”。这些类就像一个一个色彩斑斓的积木块,通过我们软件工程师的双手来搭建成不同的奇妙建筑。因此我们学 VC 的重中之重,就是了解清楚这些积木块。本书自始至终都秉持这样的教学理念,即“用类库的整体观念来串起整个 VC 的学习,让解决方案的思想引领 VC 的学习”,希望读者能从中领会。当你习惯于以这样的观念学习 VC 后,你会发现,.NET、Java 其实都是一回事。

③ 着眼于对信息整体流程的处理。一般的教材往往按照知识点,一个一个来安排章节教学,这样做是传统的套路,也无可厚非。但是我觉得,技术其实是一个整体,应该用一个内在的线连起来,这样会使人感到浑然天成。做过一些哪怕很小项目的人都应该有所体会吧,在项目中学到的知识比看书要深刻得多,也牢固得多。为什么?因为做项目时自己始终是围绕一个目标来进行,遇到不会的现学也针对性很强,这样自己所学的东西由一个项目整体串起来了,所以效果非常好。市面上也有许多诸如“VC 项目实战 XX 例”的书,但一般对于高手比较有用,初学者往往看不太明白,即使能看懂些,由于不是自己做的,所以跟自己亲自动手做的差远了。如何既能让初学者看得懂,又能形成一个整体感强的知识体系,是我写书时思考最为深入的问题。最后,我找到一条路——从 IT 这个概念本身入手。我们一般称呼搞软件的为做 IT 的,啥是 IT 呢?当然就是 Information Technology 喽,我们的对象是“信息”呀!所以针对“信息”处理的流程来介绍 VC 技术就成了本书的最大特色。

④ 尽量做到实用,使读者拿来就能用。本人学习技术以来,看的书不敢说有多么多,上百本肯定是有的。我十分理解真正热爱技术、想搞技术的人对于好的技术书籍的渴求:一要“有货”;二要通俗。通俗没问题,咱就是一个相声迷,写书追求的就是一个亲切自然外加小幽默。“有货”呢?我自己才疏学浅,技术有限,不敢和那些技术大牛相提并论,但有一点我可以保证,那就是我写书有个原则:尽量做到有用。何谓“有用”,那就是多给读者些实实在在的内容。曾看过一本书,大概是游戏程序开发之类的,当时我欣喜地捧起书(毕竟讲游戏开发的书还不算多嘛),结果 400 多页的书,将近 200 页在讲 C++ 语法,剩下的再讲讲游戏的历史,游戏的策划,最后也就没什么了。这样写出的书对真正想了解游戏开发的人有多少用处?另外,就是对于代码及一些细节要讲解详细一点。有时候技术高手往往以为某个技术点很简单不值一提,但其实未必,很多初

学者可能就会在细节上费很多工夫。我力争讲到哪个技术,大多都以自身的体会和理解来写,不人云亦云,哪怕有些不严谨之处,只要能能让读者理解就达到目的了。

⑤ 部分章节加入了扩展知识与实例,算是对主干内容的补充,以给读者提供更多的技术知识。

本书第 2、3 章由楚雄师范学院的高永丽编写,其余各章由管皓编写。书中所有实例的源代码请到北京航空航天大学出版社的“下载专区”中下载使用(<http://www.buaapress.com.cn/download.php?ptid=1&pmenuid=5>)。由于编者水平所限,书中的不妥或错误之处,敬请读者批评指正。

如读者有任何意见或问题,可以通过以下方式同作者联系:

Email: wudiguanhao001@163.com

作者的博客:<http://blog.sina.com.cn/guanhao001>(同步免费提供书中所有源代码供下载)

好了,现在就跟我开始全新的 VC 之旅吧!

目 录

第 1 章 IT、C++、Visual C++	1
1.1 混乱之治——计算机语言百家争鸣	1
1.2 永恒之塔——C/C++	2
1.3 C/S 与 B/S	3
1.4 我们的 IDE——Visual C++	4
1.5 多说几点	4
1.5.1 为什么选择 Visual C++	4
1.5.2 别忘了我们的平台——Windows	5
1.5.3 编程的背后	6
1.5.4 Visual C++ 学习路径	6
第 2 章 Visual C++ 基础	7
2.1 先利其器——安装必备工具	7
2.2 开发环境	8
2.3 Windows 编程概述	10
2.3.1 关于 Windows	10
2.3.2 Windows 下编程的学习顺序	10
2.4 Windows 编程模式	11
2.4.1 过程式编程模式	11
2.4.2 Windows 的全新编程模式——事件驱动式编程	11
2.4.3 Windows 系统的消息(Message)	12
2.5 Windows API 编程	12
2.5.1 C 程序与 Windows 程序的比较	13
2.5.2 真正的 Windows 程序	15
2.5.3 Windows 程序中几个函数的分析解读	21
2.6 Windows 下常用数据类型	26
2.7 Windows 编程由 C 模式(API)向面向对象模式(MFC)转换的必要准备知识	27
第 3 章 MFC 基础与 MFC 类库纵览	28
3.1 MFC 的前世今生	28
3.2 MFC 运行原理剖析	29
3.2.1 简易的 MFC 程序	29
3.2.2 MFC 程序运行机理分析	31
3.2.3 MFC 的消息映射机制	33
3.2.4 MFC 中具有消息映射机制的类	35
3.2.5 小改进——弹出单文档式窗体	35
3.3 MFC 类库纵览	36

3.3.1 学习类库的重要性	36
3.3.2 MFC 类库	37
3.4 一些重要且常用的类及范例	38
3.4.1 CWinApp 类及其范例	39
3.4.2 CFrameWnd 类及其范例	39
3.4.3 CDialog 类及其范例	40
3.4.4 CDC 类及其范例	44
3.4.5 CFile 类及其范例	46
3.4.6 CSocket 类及其范例	48
3.4.7 CRect 类及其范例	48
3.4.8 CString 类及其范例	49
3.5 扩展实例与技巧	50
第 4 章 组件技术——COM、ActiveX	52
4.1 组件技术概述	52
4.1.1 概述	52
4.1.2 组件的优点	53
4.1.3 COM 技术	53
4.2 ActiveX 基础知识	54
4.2.1 概述	54
4.2.2 ActiveX 控件	55
4.3 ActiveX 控件的使用	55
4.3.1 概述	55
4.3.2 ActiveX 控件应用举例	56
第 5 章 信息技术四部曲——信息采集	60
5.1 IT 技术带给我们的启示	60
5.2 一张正在编织的新的网——物联网	62
5.3 形形色色的信息、数据采集	63
5.3.1 概述——看《潜伏》有感	63
5.3.2 五彩斑斓的外部数据获取硬件	64
5.4 采集信息的重要阀门——计算机接口技术	67
5.4.1 概述	67
5.4.2 常用接口	67
5.5 Visual C++ 的串口通信编程技术	68
5.5.1 串口概述	68
5.5.2 VC 串口编程调试准备之一——虚拟串口软件	68
5.5.3 VC 串口编程调试准备之二——串口调试助手	70
5.5.4 VC 串口编程——MSComm 控件介绍	72
5.5.5 MSComm 控件编程示例	78
5.5.6 用户层协议示例	82
5.6 扩展实例与技巧	93

第 6 章 信息技术四部曲——信息传输	95
6.1 通信的力量——从草根明星到火暴的 iPhone	95
6.2 世界的神经——Internet	96
6.2.1 Internet 简史	97
6.2.2 Internet 的管理	97
6.2.3 Internet 的未来	98
6.3 TCP/IP	99
6.3.1 总体概述	99
6.3.2 网络层协议	100
6.3.3 传输层协议	101
6.3.4 应用层协议	102
6.4 Socket 概述	102
6.5 WinSock API 介绍及范例	104
6.5.1 基本函数介绍	104
6.5.2 WinSock API 编程示例 1——基于 TCP 套接字	110
6.5.3 WinSock API 编程示例 2——基于 UDP 套接字	119
6.6 WinSock MFC 及其范例	123
6.6.1 MFC Socket 类介绍	123
6.6.2 MFC Socket 类编程过程概述	124
6.6.3 CSocket 类编程范例	125
6.7 扩展实例与技巧	134
第 7 章 信息技术四部曲——信息存储	136
7.1 从 Oracle 说起	136
7.2 数据库及信息存储技术的发展简史	137
7.3 数据库诸侯混战	138
7.4 DBA, 信息仓库的守护人	140
7.5 Visual C++ 的信息存储——文件	144
7.5.1 利用 C 语言的标准库	145
7.5.2 利用 C++ 语言的标准库	148
7.5.3 利用 Windows API	150
7.5.4 利用 MFC 类库	154
7.6 Visual C++ 的信息存储——数据库	159
7.6.1 综 述	159
7.6.2 ADO 的对象模型介绍	160
7.6.3 ADO 编程模型	161
7.6.4 ADO 编程实例	167
7.7 扩展实例与技巧	182
7.7.1 创建文件夹	182
7.7.2 删除文件夹	183
7.7.3 删除文件	184

7.7.4 获取、设置当前进程目录	184
7.7.5 获取系统目录	185
7.7.6 INIT 文件操作	186
第 8 章 信息技术四部曲——信息展现	188
8.1 换肤技术	189
8.1.1 为程序披上 Windows XP 的外衣	189
8.1.2 用专业的界面库美化程序	193
8.1.3 用 AppFace 美化程序	197
8.1.4 手动贴图换肤	200
8.2 百花齐放——缤纷的控件	205
8.2.1 改变对话框控件的颜色	206
8.2.2 按钮控件	208
8.2.3 静态控件	218
8.2.4 文本框控件	220
8.2.5 列表控件	223
8.2.6 标签控件	226
8.2.7 菜单	229
8.2.8 工具栏	232
8.3 扩展实例与技巧	239
8.3.1 应用程序的桌面图标	239
8.3.2 系统托盘	239
8.3.3 曲线显示——应用 TeeChart 控件	241
参考文献	245

第 1 章

IT、C++、Visual C++

1.1 混乱之治——计算机语言百家争鸣

计算机软件技术发展至今,达到了一个繁荣时期,单单一个计算机编程语言就可以用百花齐放、百家争鸣来形容。不仅有 C/C++、Java 这样的通用编程语言,还有许许多多行业专用语言。这些语言各具特色,在各自所擅长的领域里发挥着光和热。图 1-1 是至本书截稿时止的一份编程语言排行榜。

Position Apr 2012	Position Apr 2011	Delta in Position	Programming Language	Ratings Apr 2012	Delta Apr 2011	Status
1	2	↑	C	17.555%	+1.39%	A
2	1	↓	Java	17.026%	-2.02%	A
3	3	=	C++	8.896%	-0.33%	A
4	8	↑↑↑↑	Objective-C	8.236%	+3.85%	A
5	4	↓	C#	7.348%	+0.16%	A
6	5	↓	PHP	5.288%	-1.30%	A
7	7	=	(Visual) Basic	4.962%	+0.28%	A
8	6	↓↓	Python	3.665%	-1.27%	A
9	10	↑	JavaScript	2.879%	+1.37%	A
10	9	↓	Perl	2.387%	+0.40%	A
11	11	=	Ruby	1.510%	+0.03%	A
12	24	↑↑↑↑↑↑↑↑	PL/SQL	1.373%	+0.92%	A
13	13	=	Delphi/Object Pascal	1.370%	+0.34%	A
14	35	↑↑↑↑↑↑↑↑	Visual Basic .NET	0.978%	+0.64%	A
15	15	=	Lisp	0.951%	+0.02%	A
16	17	↑	Pascal	0.812%	+0.10%	A
17	16	↓	Ada	0.783%	+0.01%	A-
18	18	=	Transact-SQL	0.760%	+0.18%	A
19	22	↑↑↑	Logo	0.652%	+0.12%	B
20	52	↑↑↑↑↑↑↑↑	NXT-G	0.578%	+0.35%	B

图 1-1 TIOBE 2011 年 4 月编程语言榜

这个图有什么作用呢?作用太大了!编程语言是程序员手中的武器啊,没有趁手的“兵器”如何练就好武艺呢?正所谓“工欲善其事,必先利其器”。《小李飞刀》里面不是有个兵器排行榜吗,上面的编程语言榜就是我们程序员的“兵器排行榜”。作为一名程序员,经常关注一下自己所使用语言的“行情”,也是十分必要的。对于想进入软件开发队伍的新手来说,选择语言应该算是头等大事了吧?经常看到的现象是一些编程新手们向老手、高手请教:我该学习哪种语言?哪种

语言更吃香、更时髦？许多编程老手、高手们则甚至把自己钟爱的语言上升到信仰的高度，为捍卫自己崇尚的语言而彼此攻击论战。从这之中，我们就可以看出计算机编程语言的重要性（也许有些编程高手已经到了手中有语言、心中无语言的境界，对语言之争嗤之以鼻，但大多数读者恐怕没那么厉害，选择一门适合自己的语言还是很有必要的）。

从图 1-1 所示的编程语言排行榜中，我们可以得到哪些信息呢？

如果你经常关注这个榜单，会知道其实在之前相当长的一段时间内，Java 语言一直占据着头名的位置。是的，Java 的大名估计在 IT 圈子里混得没有谁不知道吧？看看那些招聘会里的招工启事，你就会发现要 Java 的人真不少。鉴于 Java 的火，不少人入门时就选择了它。Java 为什么这么厉害呢？其实 Java 这个名字早已经不是一个简简单单的语言的概念了，而是一个技术平台与体系。这里面包含了非常多的技术。真正值钱的，倒不一定是你 Java 语言玩得多熟，而是你对于 Java 技术体系是否谙熟于心。那么 Java 到底是干啥用的呢？我不是 Java 专家，可说不全，更不敢在这贻笑大方，但有一点认识非常深刻，那就是大型企业信息化系统。大家知道一个非常火的 CIMS 这个词吧，就是“计算机现代集成制造系统”，清华大学就有这个专业，Java 几乎就是它的钦定语言。企业是经济的命脉所在，Java 又在企业信息化中发挥了巨大作用，当然要的人多，当然火了。

但 Java 现在遇到了强有力的挑战，那就是来自微软公司的 .NET 体系技术。如榜单中的 Visual Basic(简称 VB)、Visual C# 都是 .NET 中的主力语言。VB 号称是最容易的语言，是初学者学习编程的一种不错的选择。我曾经学过一段，确实简单易学，尤其加入 .NET 后，升级比较迅速，功能更强了。C# 现在也很火，可以说是与 Java 并驾齐驱。它是 .NET 技术体系的首选语言。我曾经专心学过一段 .NET 技术，学过后发现，C# 和 Java 简直太像了，有时候单看代码甚至都分辨不出。.NET 与 Java 两大技术体系，目前在外包行业中用得非常多，需求量也很大，可以说是一种“饭碗”技术，如果读者朋友对软件的学习只是出于谋生考虑，爱好无所谓的话，那随便报个学习班(比如北大青鸟)，估计也无外乎这两个。至于其他的语言，后面提到再说吧。我们这本书的主角语言可不是 Java 或 C#，而是 C++，下面的主角要登场了。

1.2 永恒之塔——C/C++

图 1-1 中，C 与 C++ 是分开来列的。可谁都知道，它俩几乎就是一家的。如果把它们所占的比例加在一起，已经接近 25%，超过了 Java 许多。这样的比例足以说明 C/C++ 的王者地位。尽管一直有不少新型的时髦语言不断涌现，但这个比例却一直很稳定(即使是在 Java 最火的时候)。想想我们这代人上大学的时候(2000 年前后)，估计绝大多数的人入门语言是 C 语言，绝大多数用的是老谭(谭浩强)的书。因此对于 C 语言是有特殊感情的。

C/C++ 语言是这份榜单中资格最老的王牌语言。看看那些后起之秀们，不少是它们的后代，至少是借鉴了它们的。可以这样说，如果你精通 C/C++ 语言，那么再学榜单上绝大多数的其他语言几乎可以不费吹灰之力。学好学精 C/C++ 也同样需要付出巨大的努力，一般比学好其他语言要付出更多。但付出总有回报，一个好的 C/C++ 程序员往往会得到非常好的待遇(物以稀为贵嘛，C/C++ 高手还是相对少的)。

C 语言的魅力在于“简约而不简单”。其没有过于复杂的语法，但却拥有强大的底层控制能力，可以充分满足程序员的控制欲。C++ 是本书的主角，它是在 C 语言的基础上加入了面向对象方法后而成的语言，可以说是目前最为成功的面向对象语言，同时也几乎是最为复杂的面向对

象语言。没有几个人敢说自己精通 C++，一般敢说这话的恐怕是本科毕业生居多吧。但是你也别害怕，其实如果我们只作为一般的应用软件开发者，根本用不着 C++ 的许多复杂特性，掌握基本的、常用的即可。

C++ 曾经被认为是“万能语言”，从底层到应用层甚至 Internet 几乎是无所不在。但随着 Java、C# 等新兴语言的崛起以及诸如 PHP、Python、Ruby 等快速高效的脚本语言的火热，C++ 的领地在不断缩小。这其实是软件技术发展的必然结果。没有一种语言是万能的，最好是各司其职，在自己最擅长的领域发光发热。Java、C# 的优势是 C++ 略显不足的，而 C++ 的优势，同样是 Java、C# 等无法撼动的。那么 C++ 的优势是什么的？那就是速度和运行效率。由于设计初衷的差异，运行效率是 C++ 特别看重的。它本身就继承于偏向底层的 C 语言，所以和系统底层的贴近使它特别适合开发注重速度的应用。两个非常明显的例子，也是我非常熟悉的例子，一个就是当下非常火的网络游戏，还有就是工业控制。对于高校、科研院所搞工科的，比如搞机电控制、机器人的，玩单片机、嵌入式的，C++ 的身影到处可见。对于科研部门的软件工程师来说，C++ 的使用应该是非常广的（当然 C 也很广，但我不想总把它俩拆得特清）。

1.3 C/S 与 B/S

目前的软件开发有两大主流架构，即 C/S（客户端/服务器）和 B/S（浏览器/服务器）。我们使用的 QQ，就是典型的 C/S 模式的软件。它要求客户端（比如你用 QQ，那就在你的电脑上）下载安装客户端软件，而后通过与强大的服务器的通信来实现各种功能。B/S 架构的软件其实是 C/S 的一种特例，只不过是客户端不用安装特别定制的软件，只安装个浏览器就行了。传统的软件大都是 C/S 架构的，近些年来，Web 技术大行其道，B/S 架构日渐火热起来，所以你就看到了图 1-1 榜单中 PHP 的如日中天。现在很多人把 B/S 挂在嘴边，一些应用在逐渐由 C/S 向 B/S 迁移，看似好像 C/S 已经没落，而要被 B/S 取代一样。很多人跟风学 PHP、.NET，学 Web，弄 B/S 估计就是感觉 B/S 有前途，很火的缘故吧。

其实两种架构各有优势，并没有什么优劣之分。我们这本书的主角是 VC，它以开发 C/S 架构软件为主。因此，我们的着眼点就放在 C/S 架构上。C/S 好不好呢，火不火呢？初学者可能会这样问。我的回答是：当然好，当然火了。举两个例子，你就没话说了：一个是 QQ；另一个是网络游戏。这就够了吧！都是大大有“钱”途的呀。我们学技术当然是为了能有个好“饭碗”，要不学它干啥呢，对不对？我们学好了 VC，就为开发 C/S 架构的软件打下了良好的基础；网络游戏公司的程序开发人员待遇一般也是不错的。所以为学习 VC 所付出的努力是非常值得的。

图 1-2 所示为 C/S 与 B/S 架构的示意图。



图 1-2 C/S 与 B/S 架构示意图

1.4 我们的 IDE——Visual C++

IDE(Integrated Development Environment,集成开发环境)是专门为方便程序员开发程序用的应用软件。它把程序的编辑、编译、调试、运行等都统一集成在一起,大大加快了程序员开发软件的速度和效率。可以说,现在一般的软件开发都离不开 IDE 的使用(当然也有一些大牛愿意使用文本编辑器写程序,再编辑、编译,那就另当别论了)。

本书的主角——Visual C++,就是我们开发 Windows 下应用程序的 IDE。请注意这句话,Visual C++是 IDE,许多初学者往往会问一个非常外行的问题——学 C++好还是学 Visual C++好?其实,这两个是不能相提并论的。Visual C++是我们利用 C++语言开发 Windows 程序的工具,而不是一种语言。C++ Builder 也是开发 Windows 程序的一种 IDE,所以 IDE 并非只有 VC 一种,但 VC 是最主流的。

VC 的版本已经更新到了 Visual C++ 2010 了,Visual C++ 6.0(简称 VC 6)以后的版本都是基于 .NET 平台的。但有一个有意思的现象,就是 VC 6 仍然是使用者最多的。为什么呢?我也是使用 VC 6 的,我自己给出的原因是:

① VC 6 足够高效、轻巧; .NET 太臃肿,我不喜欢。

② .NET 技术体系中 C# 才是主角,VS 98 中 VC 6 是主角,VS 98 作为经典的开发环境积累了大量的程序,因此有很多学习的例程,这些都是很宝贵的财富。

VC6 的经典界面如图 1-3 所示。

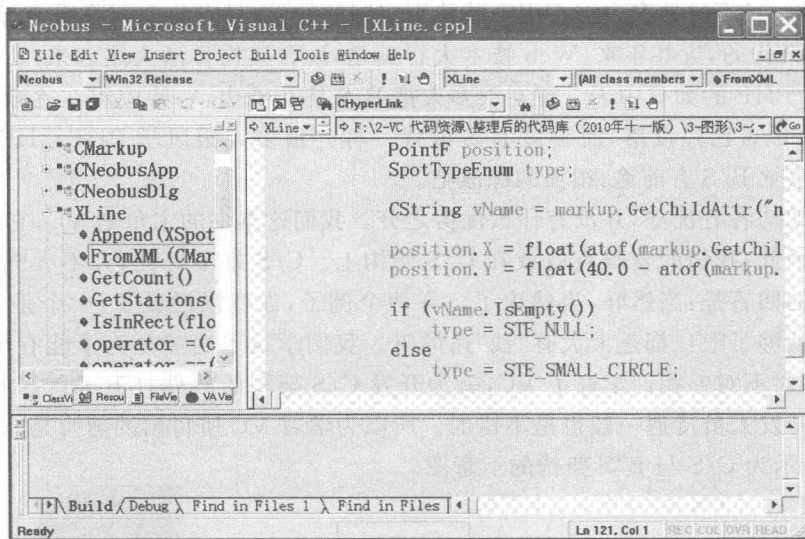


图 1-3 C++经典 IDE——Visual C++ 6.0

1.5 多说几点

1.5.1 为什么选择 Visual C++

C/C++语言作为最为经典的程序设计语言发展到今天,已经非常成熟与稳定,而且在无数

的聪慧的软件大师、程序员的努力下,积累了大量优秀、经典的代码。这些代码资源是后来软件开发者及相关 IT 从业人员非常重要的宝库。通过学习前人高手的代码,可以让我们更快登堂入室、少走弯路,更快地向开发高手迈进。可以说,C/C++ 语言是计算机软件技术发展以来影响最大的语言,没有哪种语言的影响力可以与之比肩。而同时,C/C++ 的代码资源也是最为丰富的,所以即便是从学习资料方面来考虑,选择 C/C++ 也是非常明智的。

C/C++ 的语言特性非常丰富。C++ 融入了面向对象思想非常多的特性,以致使它变得有些复杂,学通学精它不容易。后来者如 Java、C# 都针对它其中非常复杂易出错的部分进行了剔除或改进,最为典型的就是指针与多继承。其实我们可以反过来想,正因为它特性丰富,我们才可以学到更多的东西;否则,你学 Java,说它去掉了指针更安全,你连一点感觉都没有,为什么更安全,指针到底有什么不安全的?恐怕只有谙熟 C++ 才会对这些东西理解体会得更深刻。况且,如果你连最不好学的 C++ 都学会了,其他的任何语言还不都是小菜儿吗?选择学习 C++,值得。作为 Windows 平台上利用 C++ 开发软件的 IDE,VC 绝对是首选。因为它的开发者是微软公司。这层关系摆在这儿,不选它选什么呢?一般学技术,选个腿粗的来抱不会错。微软公司的综合软件技术实力是所有软件公司里最强的,所以学习它的技术你不会吃亏。当然,凡事都不能绝对化,也要看看其他开发者在用什么。从这点看,Windows 平台 C++ 开发用得最多的也就是 VC。所以综合考虑,选择 VC 没有问题。

IT 世界里的竞争从来都是残酷的。就连软件霸主微软也是时刻感受到来自其他公司的威胁。近几十年的软件发展史可谓是跌宕起伏、趣味横生的一段历史。建议大家有空可以了解了解,比如 Borland 与微软的 IDE 之争,Sun 与微软的恩怨导致的 Java 与 .NET 对立,等等。从中读者可能品味出一些东西,对于技术的判断也能增加一些认识。

1.5.2 别忘了我们的平台——Windows

我们编写软件不能在光溜溜的硬件板上,而是要依托一个平台——操作系统(当然,如果你能绕过操作系统写个商用 BIOS,那你就可以笑傲天下了,直接去微软公司做个技术总监吧)。

现在流行一句话:伟大的梦想需要实现的舞台。再好的创意没有实现的平台也是白扯。我们编写程序需要的最重要的平台就是操作系统。不同的操作系统有不同的特性、不同的编程接口与编程模式。因此,要想开发出优秀的软件,需要对于开发软件所依托的操作系统非常熟悉。现在有两大主流操作系统:Windows 和 Linux/UNIX。Windows 是大众最为熟悉的,因为它是应用最为广泛的客户端操作系统;而 Linux/UNIX 则是服务器端操作系统。近些时期,随着 Ubuntu 等桌面型 Linux 版本的逐渐发展,Linux 的桌面份额有所上升,但和 Windows 仍然无法相比。Windows 的用户最多,我们选择它没有问题。我们选择 VC 编程,因为它就是用来开发 Windows 操作系统环境下的应用程序的。利用它,我们既可以直接利用 Windows 提供给我们的 API 进行编程,也可以利用经典的类库 MFC 进行开发。前者主要是 C 模式,后者则是 C++ 模式。在本书中,主要针对 C++ 模式的 Windows 程序开发,也就是 MFC 编程。

既然是在 Windows 上开发,我们就要知道 Windows 程序是什么样的,有什么特性?这对于用过无数 Windows 软件的你也许不是大问题,毕竟 Windows 下的软件大都一个样嘛。但这只是对于大众用户的要求而已,而作为软件工程师的你当然不能就此止步。你要更深入一步,了解 Windows 注册表是怎么回事,怎么用;了解 Windows 的硬件驱动是怎么回事,等等。再往下,你就得知道 Windows 操作系统的基本内核是怎么组成与运行的,Kernel.DLL、USER.DLL、GDI.DLL 是干什么用的,等等。这些你可以不深究,但知道个皮毛对于以后的开发都是大有益处的。

虽然 Windows 不开源,不过还是有许多介绍其原理的书籍,你有必要看一看。最后,最重要的一点,就是 Windows 给程序员的编程接口,也就是常说的 Windows API,通过它们我们基本可以了解一个大略的 Windows。请有空看一看介绍 Windows API 的书籍,最好日积月累地记住一些重要的有代表性的 API 函数,这一点很重要,是学好 VC 的根本保障。

1.5.3 编程的背后

编程是什么?很多人以为:不就是在键盘上敲敲代码么。当然,编程的外在表现就是在键盘上敲代码,可是在这背后却蕴含了太多的东西。许多人学了点语言,编了点例程就觉得是在编程了,其实这还差得很远;许多人利用 VC 能够“弹出”个对话框实现一些能交互的小功能后就觉得会编程了,实际远非如此。要想成为一个好的软件开发工程师,需要学习非常多的东西。建议读者朋友,如果想在软件方面有所发展,多去学习一些与软件开发密切相关的知识。这其中最为重要的是数据结构与算法、操作系统、数据库、多线程、网络 TCP/IP、软件工程、设计模式。如果你不是计算机科班出身,那么建议你去著名高校的计算机系下载一下人家的课程表,看看计算机技术的基本知识体系是什么样子的,这样对于自己知识结构的整体把握是很有好处的。

同高手的交流也是非常重要的。你可以直接向他们请教,也可以通过读他们写的代码间接地向他们学习。这其中,一方面是向他们学习技术,另外他们有很多是业内人士,通过与他们交流你可以了解到现在 IT 的行情,不同工种的工作内容等。总之,闭门造车是不利于自己的长远发展的。入门易,学精难。计算机技术的学习与掌握需要不断学习、实践与积累。多看书、多动手写代码是成为高手的不二法门。

1.5.4 Visual C++ 学习路径

学好 VC 并不容易,如果没有一个好的方法,东一榔头西一棒子地去学,效率低下不说,还会影响自己的学习积极性。如果遵从一些过来人的经验就会少走很多弯路,更加快速和顺利地踏上利用 VC 开发 Windows 软件的大路。

首先,你要学好 C/C++,这是语言基础,否则先去补一补,不然根本无法学 VC。我们主要提倡用 MFC 编程,所以 C++ 成为必要(用 Windows API 编程模式只需要 C 语言基础,但难度过大)。当然,也不用特别害怕,作为初学者对于 C++ 的掌握点到为止即可,因为 C++ 有许多复杂特性,初学者没必要深究,掌握最基础的东西够用就行了(封装、继承、多态、基本库就是最基础也是必须掌握的,多继承、模板、STL 等可以暂时不掌握)。

尽管我们用 MFC 编程,但首先要学习一下 Windows 程序的结构,了解 Windows 程序的运行原理。MFC 只是对于 API 模式的 C++ 封装,所以理解 Windows API 模式原理对于学习 MFC 是非常有帮助的。熟悉、掌握 MFC 类库是学 VC 的主要学习内容,对类库的掌握程度决定着 VC 水平的高低。

最后就是本书特色了,那就是着眼于信息整体处理流程的处理过程,形成自己的“技术解决方案”。通过这样的方式,读者会更加深入地理解 VC 各个技术点的用途,以避免由于琐碎知识点的拼凑而造成的“只见树木不见森林”的状况。相信本书会使你对 VC 的学习更有成就感。