白领就业指南：

C++ Builder 6.0

设计师之路

钱 栩 保春艳 康祥顺 编著

李正非 审校

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

白领就业指南：
C++ Builder 6.0 设计师之路

钱　栩　保春艳　康祥顺　编著

李正非　　审校

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是一本指导你如何最大程度地使用 C++ Builder 6.0 开发的就业指南。书中详细介绍了关键的编程概念以及利用 VCL 环境来开发应用程序的基础知识，还提供了众多在 VCL 环境下使用 C++ Builder 功能的技巧、具有实践性的建议以及可以立即运行的重要解决方案的详细代码，内容主要涉及标准 C++基础知识、VCL 库、Windows 窗体、图形编程、多线程技术、数据库应用、分布式应用、Web 开发等。通过本书的学习，你可以高效地利用 C++ Builder 开发应用程序，能够理解关键操作的内部实现机制，有助于迅速利用 C++ Builder 来实现大量的编程任务，成为一个聪明的设计师。

本书适用于大中专院校学生、程序设计人员和 C++ Builder 爱好者作为一本就业实践指南。

图书在版编目(CIP)数据

白领就业指南：C++ Builder 6.0 设计师之路/钱栩，保春艳，康祥顺编著.—北京：电子工业出版社，2005.11
ISBN 7-121-01787-3

Ⅰ.C… Ⅱ.①钱…②保…③康… Ⅲ.C 语言-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2005）第 111067 号

责任编辑：徐云鹏
特约编辑：卢国俊
印　　刷：北京天竺颖华印刷厂
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
　　北京市海淀区翠微东里甲 2 号　邮编：100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：18.75　字数：470 千字
印　　次：2005 年 11 月第 1 次印刷
定　　价：26.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。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。

前　言

C++ Builder 是 Borland 公司推出的优秀的 32 位 Windows 开发工具。它不仅具有 C++语言的所有优点，而且还继承了 Delphi 使用简便、功能强大、效率高等特点。C++ Builder 可以说是至今最好的 Windows 开发工具之一，其友好的集成开发界面、可视化的双向开发模式、良好的数据库应用支持以及高效的程序开发和程序运行，备受广大程序设计师的好评。

从最早的 Turbo C 到目前的 C++ Builder，Borland 公司的 C++编译工具一直享有盛誉，C++ Builder 除了延续高效的编译器的特点之外，更重要的是有很强的数据库和网络应用程序开发能力，而且能快速、简单地实现。此外 C++ Builder 通过引进结构化错误处理，支持多线程，快速建立和使用 Web 服务的功能，建立强大的数据库交互模式等，又扩大了原有的编程能力。如果要用 VC++完成这些功能是非常费时费力的。

正是由于 C++ Builder 的快速可视化开发环境，使得编写程序简单而又快捷。它能使你在最短的时间内迅速为公司创造利润，实现自身价值。

随着 C++ Builder 技术的进步和发展，它还会在今后很长时间内占据主流开发工具的地位。

本书全面深入地介绍了 Windows 平台上的编程技术、数据库应用开发技术以及基于 Internet 上的 WinSock、Web 开发，并提供了大量在 VCL 环境下使用 C++ Builder 功能和特性的技巧。每一种技巧都提供了现成的源代码，可以用来试验某个编程的概念，也可以把它们剪贴到自己的程序中。此外，在介绍每个技巧时，还提供了对代码执行过程中的每个步骤的解释。通过阅读本书，你将学习到以下几个方面的知识：

- 学会如何使用 C++进行面向对象的开发，如类、继承、多态等；
- 熟练掌握基于组件的开发，在程序中充分发挥 VCL 库的强大功能以完成特定的任务；
- 了解由 Windows、标准 C++和 VCL 提供的异常处理机制，这也是一般的程序员所忽略的；
- 掌握图形编程的基本技术；
- 学会用注册表保存应用程序的配置信息；
- 了解 C++ Builder 的支持多线程的功能；
- 使用 C++ Builder 开发高效的数据库应用程序；
- 使用 C++ Builder 开发分布式多层应用；
- 进行 Web 应用的开发，利用 WebBroker 快速生成 Web 页面。

本书包含三个大的部分，共 15 章内容，通过这三个部分的学习，你将会对 C++ Builder 以及基于 VCL 的程序设计有深入的认识和掌握，能够进行一些应用系统的编写和处理，从而在就业中立于不败之地。

本书主要由保春艳、钱栩、康祥顺编写，另外参与本书编写的合作者有李欣、熊梅、康祥琴，本书拟订大纲和最后的统稿、协调工作由钱栩完成。最后还要感谢电子工业出版社和美迪亚电子信息有限公司的所有人员，感谢他们为本书的顺利出版所做的努力！

由于水平有限，书中难免仍有一些错误，敬请广大读者不吝赐教，编者在此表示感谢！

职业札记：做个聪明的程序设计师

前几天看了《中国软件业人才打造模式之辩》，又看了最近中国各大公司招聘的情况，真是大有感慨。中国的莘莘学子真的被抛弃了吗？希望有人能给他们一个信心，给他们一个信念，给他们一个帮助，给他们一个不被抛弃的理由——迅速开始学习现在的技术，做个聪明的程序设计师，做个能为公司赚取更多利润的白领一族！

一个公司，看名字看不出来有什么门道，也不知道规模如何，只在招聘中这样写到："精通 VB、Delphi、VC++、Java 语言编程，熟悉 Windows 和 UNIX/Linux 操作系统和 TCP/IP 协议，熟悉 Windows 和 Linux 系统编程和网络编程……"还有这样的："精通.NET 编程，5 年以上的.NET 下软件开发工作经验……"

不知道其他的软件人士或者程序开发人员是什么水平，我只说我认识的一些软件工程师和设计师们，谁会"精通"VB、Delphi、VC++、Java 这么多种语言啊？

我想，不仅仅是在中国，在世界上，也没有这样的几个人吧？以为是在招超人呢！！！

还有这个"精通.NET 编程，5 年以上的.NET 下软件开发工作经验"，.NET 才问世几年啊？怎么能有 5 年以上的开发经验啊？但是，别忘了，为了生存，我们必须适应社会，而不是让社会适应我们！

先说说自己的专业，计算机科学与技术，是个什么专业？既要学习计算机理论，又要去实践，既要学习软件，又有硬件和网络、通信类。要是外行看起来我学的专业，觉得以后工作选择余地一定很大，但是，实际上来说，根本不是这样的，我们虽然学了很多，但是基本上都是学习到了皮毛而已，由于涉及到的范围太多，都很难深入去研究。到了毕业，硬件方面比不过专门学电子的，软件方面比不过专门学软件的，网络通信方面也是这样。

庆幸的是，对于计算机这个行业，兴趣是最主要的，从初中开始就学习计算机，到现在，已经 10 年了！

从最早接触的 Basic 和 C 开始，也就注定了一直都是在 Basic 和 C 的基础上面来继续。然后一直到了面向对象语言，我就直接过渡到了 Visual Basic 和 C++ Builder 上面。

记得以前一些程序设计的书曾经讲过，C 语言是很基础，也是很有深度的一门语言，就算以后不用，也最好能掌握，这个语言的优势实在是很大。所以，我对于 C 语言也下过一番工夫。就因为这样，在有了 Java 语言后，我并没有下很大的工夫去深入学习，因为我觉得，语言到了最后都是相通的，掌握了软件设计的真谛，任何语言只是实现的一个途径，一个工具而已。

但是，现在的环境，让我感到非常困惑。现在招聘时动辄就是 Java 程序设计人员等。Java 软件开发工程师、Java 软件编程人员等。我不禁在问，难道中国的软件业发展就这么快？这么快就有这么多的程序员，这么多的软件设计师精通了 Java 语言吗？难道那些招聘 Java 人员的公司就有能力完成这样的东西吗？这个现象令我很不解。既然一个项目、一个工程可以用 Visual Basic 完成，也可以用其他语言来完成，为什么一定要追求 Java 呢？为什么不能使用更为简洁、快速、高效的工具呢？是程序设计师不聪明，还是人力资源部门的所谓专家"太专"了？

而且现在中国软件业也有一个非常奇怪的现象，大学学习的课程与社会需要的东西脱节。学习是要学习软件设计的真谛，是一个思想，而不是要学习那么多的语言。那些语言，都是要为完成的计划来服务的，只要能完成，用什么都可以的。

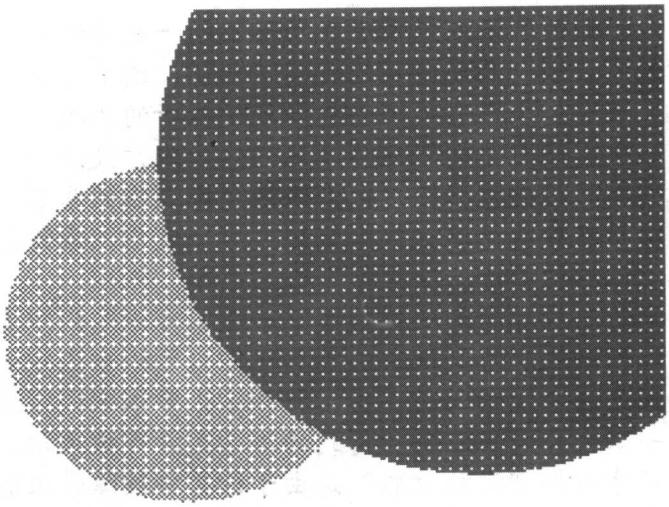
另外，现在学校教育非常严重地滞后于中国软件业的发展，大学里面都学到了一些什么东西呢？基本上没有什么在工作上特别有用的东西，有很多已经不再适合了，就连软件都有升级补丁、升级包，为什么大学的教科书就没有呢？这么多年了，虽然面向过程的语言是经典，但是看看现在，面向对象已经成为以后的大势所趋，如果再一味在旧的东西上花太多的时间去深究，我认为没有这样的必要。好像现在我们在大学所学的软件工程都是按照面向过程的方式来讲的，但是，在国外，软件工程和数据结构、算法一类的东西都是按照面向对象方式来讲授的。再来看看我们学的，差别是不是很大呢？也难怪大学生跟不上社会的步伐和节奏。

前几天公司项目培训，组长说："总公司项目会议，印度人占了七成，我想说，各位，如果我们还没有危机感，不去靠自己的努力工作阻挡印度人的话，我们的明天将寸步难行！"对于急功近利的、短视的中国软件企业，我们只能以其人之道还治其人之身，用快速的学习，聪明的学习方法来应对。毕竟我们需要生存！

记得我最开始学习 C++ Builder 的时候，原因很简单，就是看中了面板上的那一大堆控件，而且也习惯了使用 C++。VC 的 MFC 难以最做出美观的界面，VB 的语法太松散，都是我不喜欢的。随着学习的深入和实践经验的不断增加，我越来越感到 C++ Builder 的魅力，其博大精深是 VB 无法媲美的，和 VC 相比也毫不逊色。作为流行的 RAD 工具之一，C++ Builder（还有 Borland 公司的另一个开发工具 Delphi）真正体现了使用简便、开发快速的特点。而如果用户需要做非常底层的应用（如图形算法处理和直接对硬件端口的操作），那么 C++ Builder 也同样是一个好选择，它几乎对 Windows 所有的 API 都进行了封装，Opengl 单元中还提供了大量的涉及到 Opengl 编程的函数，用户也可以直接在程序中嵌入汇编代码段。这一方面可以使程序直接操作硬件，另一方面，即使不操作硬件，汇编的使用也使程序代码执行效率大为提高。如果既需要美观的界面，又需要底层的操作，那么 C++ Builder 将是你最佳的选择。C++ Builder 和 Delphi 的队伍在不断扩大，无论学术性强的学校还是在面向商业的公司（据我所知，北大方正、中软还有其他一些小软件公司的 Windows 主要编程工具就是 BCB 和 Delphi），C++ Builder 和 Delphi 都拥有众多的用户，而且必将越来越多。"真正的程序员用 C++，聪明的程序员用 Delphi"，C++ Builder 兼有 C++的深邃和 Delphi 的便捷，那么使用它的的程序员必定是聪明的职业人了。当然所有这些都归功于 Borland 公司对用户的真正体贴，让我们向他们表示由衷的感谢！

本书为白领就业指南，专门为立志于成为一名聪明的软件设计师的读者编写的速成教材，书中共分三个部分：基础、实践和综合提高。每个技术要点都通过一个典型的应用案例来讲解，这些案例非常注重实际应用，所有的源代码可以直接应用到你的开发中去，也为学习这门技术的读者提供了最佳的程序思维方式和途径，使你能够在深入了解本书后，快速走上软件设计师的岗位，摇身一变，成为白领一族，做一个聪明的程序设计师！

目 录



第一部分 学习——跨越求职路上的鸿沟

引 言

艾青在其《礁石》一诗中写到:

“一个浪，一个浪，/ 无休止地打过来，/ 每一个浪都在它脚下，/ 被打成碎沫，散开……/ 它的脸上和身上，/ 像刀砍过一样，/ 但它依然站在那里，/ 含着微笑，看着海洋。”

读了这首诗，自然会令人想到浩翰的大海，它永远不停地翻着巨浪。海上的任何一只航船都会或多或少地经受风暴与巨浪的洗礼。这不正像一个人的一生，无论如何都不会一帆风顺，而总是伴随着大大小小的挫折吗？朋友，当挫折来临时，你会如何面对它呢?告诉你，面对挫折不要愁眉不展，而应勇敢无畏地含着笑容去面对!

应该认识到，古往今来，伟大成就的殿堂前无不以挫折为台阶。电灯的诞生，是爱迪生用几千次实验的失败换取的;《命运》交响曲，是贝多芬双耳失聪后的产物; 我们最熟悉的中国新民主主义革命，更是经历了无数风风雨雨、坎坎坷坷才获得最后的成功。可以说，没有挫折这块坚实的基石，就不会有今天人类的科技、文艺乃至社会历史进程的这一座座丰碑。我们要想树立起自己的成功之碑，也必须用挫折为它奠基。那么，怎样才能奠好这一块基石呢?这就需要我们正确地面对挫折。

挫折只不过是强者成功路上的一块垫脚石。这是因为，他们在面对挫折时，并不畏缩，而是微笑地迎接这一切。这微笑并不是漫不经心的，相反地，它恰恰反映了一个人有足够的勇气接受挫折的挑战。这样，他们才能清醒地审视挫折，从中发现自己的错误与不足，然后想方设法在今后去修正与弥补。同时，在努力战胜挫折的过程中，他们锻炼出更顽强的意志，铸就了更坚利的精神之剑，更有助于在今后的路上披荆斩棘，勇往直前，最终摘取成功的桂冠。这就是古人所说“艰难困苦，玉汝于成”之意。

而对于弱者，挫折成了一道不可逾越的鸿沟。他们在这条沟旁徘徊、唉声叹气。却没有想到这条沟正是他们自己给自己挖的。他们没有勇气面对挫折，因而也无法去继续闯荡，自己放弃了很多本来能得到的东西。曾经有一位日本青年到一家大公司去应聘，得到的消息是没有被录取。他在绝望中准备自杀，自杀未遂后才得知“没被录取”是由于计算机故障带来的误报。正当他接到聘书喜形于色之时，一纸解聘书又飞到他手中，说他不能很好地面对挫折，必不能胜任今后的工作。想想看，这位青年的成功机会就在自己手中，却因为承受不了挫折，而让这机会从他指缝间溜走了。没有勇气接受挫折的挑战，本已积累起的成功的筹码都会失去它的份量，而新的筹码你又不能拿到，那又怎么能达到成功的顶峰呢?

曾经的努力和希望在我们求职的路上遭到严重的打击，“唯有读书高”的思想已经被这个社会所抛弃，企业的短视、学校教育的滞后、中国特色的求职行情，使得用人单位需要立即能够为自己创造价值的人，这就是我们求职路上屡受挫折的根源所在。因此，学习企业所需要的技术、为社会创造价值就成为新的希望。万丈高楼平地起，你必须为跨越求职路上的鸿沟做好准备——学习，特别是基本功的修炼。本部分将会让你轻松跨越求职的鸿沟，引领你进入聪明程序设计师的殿堂，如果你已经有了C++基本功力，将会更加顺利和轻松!

来吧!当你身处逆境之时，你依旧拥有希望，拥有力量!向那些前进道路上的障碍挥手告别吧!它们是生活对你的考验，同时又是生活给予你的馈赠。因为只有凭不屈不挠的精神跨越了所有这一切时，你体味到的快乐才是最真切、最刻骨铭心的!

第1章 夯实C++基本功

1.1 初步认识C++程序

下面是一段用于输出字符串的简单程序，从中可以了解到C++程序中需要用到的基本元素。

```
#include <iostream>
using namespace std;
void main ()
{
  char* s = "初步认识C++";
  cout << s << endl;
}
```

每一个C++程序，不论大小如何，都由函数和变量组成。函数中包含若干用于指定所要做的计算操作的语句，而变量则用于在计算过程中存储有关值。在本例中，函数的名字为main。一般而言，可以给函数任意命名，但main是一个特殊的函数名，每一个程序都从名为main的函数的起点开始执行。这意味着每一个程序都必须包含一个main函数。

main函数通常要调用其他函数来协助其完成某些工作，调用的函数有些是程序员自己编写的，有些则由系统函数库提供。上述程序的第一行语句：

```
#include <iostream>
```

用于告诉编译器去包含位于标准库里的流输入/输出功能的声明。

在函数之间进行数据交换的一种方法是调用函数向被调用函数提供一个值（参数）列表。函数名后面的一对圆括号将参数列表括起来。在本例中，所定义的main函数不要求任何参数，所以用空参数表（）表示。函数中的语句用一对花括号{}括起来。

在程序运行过程中其值可以改变的量称为变量。一个变量应该有一个名字，在内存中占据一定的存储单元。变量定义必须放在变量使用之前，一般放在函数体的开头部分。在程序运行过程中其值不发生改变的量称为常量。本例中的char* s = "初步认识C++"；是一个变量声明语句，它由一个类型名和变量表组成，这里将变量s声明为一个字符数组（字符串），并且将常量"初步认识C++"作为初值赋给s。

运算符<<将它的第二个参数写到第一个参数里。本例中，字符串变量s的值将被写进标准输出流cout。第二个运算符后面的endl用于输出换行。

1.2 类型和表达式

1.2.1 关键字

标准 C++中定义了 63 个关键字，如表 1-1 所示。

表 1-1 C++中定义的 63 个关键字

asm	Auto	bool	break
case	catch	char	class
const	const_cast	continue	default
delete	do	double	dynamic_cast
else	enum	explicit	export
extern	false	float	for
friend	goto	if	inline
int	long	mutable	namespace
new	operator	private	protected
public	register	reinterpret_cast	return
short	signed	sizeof	static
static_cast	struct	switch	template
this	throw	true	try
typedef	typeid	typename	union
unsigned	using	virtual	void
volatile	wchar_t	while	

这些关键字不能作为其他对象的名称使用，例如，不能作为变量名或函数名。关键字都由小写字母组成，C++是区分字母大小写的，因此 int 是关键字，INT 则不是。但是最好不要使用这些关键字改变大小写后产生的名字，因为它们可能在一个庞大的程序中产生混淆，例如，有可能在某个程序库中已经用 typedef 将 INT 定义为 int 的别名。

1.2.2 标识符

标识符是用来指代变量、函数或其他用户定义对象的名字。标识符必须是由字母、数字或下划线组成的字符序列，而且第一个字符必须是字母或下划线。例如，以下的标识符是合法的：

```
abc123,_xyz,xy_z
```

而这些则不是合法的标识符：

```
123abc,x?y,a..b
```

在 C++中，字母是要区分大小写的，因此，name、Name 和 NAME 是三个不同的标识符。标识符不能和关键字重名，也不能和函数名重名。

变量与常量是程序中所要处理的两种基本数据对象。声明语句说明变量的名字及类型，也可以指定变量的初值。运算符指定将要对变量与常量进行的操作。表达式则用于把变量与常量