

当代计算机职业培训系列教程

C++ Builder

广正工作室 编著

实用教程

经验丰富的专家精心撰写，
使您的学习过程更有效

内容全面，
由浅及深，
带给您一份难以言传的惊奇

实例讲解，
边学边做，
让您事半功倍



机械工业出版社

CMP

当代计算机职业培训系列教程

C++ Builder 实用教程

广正工作室 编著

机械工业出版社

本书是介绍 C++ Builder 和 C++ 编程语言的基础读本。在简要介绍 C++ Builder 的环境的基础上,重点讨论了 C++ Builder 的 C++ 语言。通过实例介绍了面向对象编程的方法、创建表单的原则及如何创建多文档界面的应用程序,着重讨论了程序的调试和异常处理。

本书的宗旨是通过大量或深或浅的实例讲解 C++ Builder 的内容,每章后面均有练习,通过上机实践,必会让您事半功倍!

本书内容由浅入深,适合所有 C++ Builder 初学者和中级读者阅读。

本书由机械工业出版社出版,本书的任何部分不得以任何方式复制或抄袭。

版权所有,翻印必究。

图书在版编目(CIP)数据

C++ Builder 实用教程/广正工作室编. - 北京:机械工业出版社,1998.6

当代计算机职业培训系列教程

ISBN 7-111-06524-7

I. C… II. 广… III. C 语言-程序设计-技术培训-教材 IV. TP312

中国版本图书馆 CIP 数据核字(98)第 14191 号

出 版 人:马九荣(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:傅豫波 姜 南

昌平环球印刷厂印刷 · 新华书店北京发行所发行

1998 年 6 月第 1 版第 1 次印刷

787mm×1092mm 1/16·27.25 印张

印数:0 001-6 000 册

定价:44.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

序

C++ Builder 是一个全新的可视化的 C++ 编程环境，它为我们提供了一种方便、快捷的 Windows 应用程序开发工具。它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想，采用了弹性的、可重用的和完整的面向对象程序语言（Object-Oriented Programming Language）。对于广大的程序开发人员来讲，使用 C++ Builder 开发应用软件和数据库应用程序，无疑会大大地提高编程效率，而且随着应用的深入，我们将会发现编程不再是枯燥无味的工作——C++ Builder 的每一个设计细节，都将带给我们一份难以言表的惊奇。

关于 C++ Builder

笔者曾使用过 Pascal、C、Borland C++、Visual C++、Visual Basic 和 Delphi 写过程序，开发过应用系统，但从没有一个系统像 C++ Builder 这样给我们留下如此深刻的印象。虽然 C++ Builder 中的每个部件都是指针，会使有一定基础的读者感到畏惧（正如笔者最初也都曾对此表示怀疑一样），但这也正是让人倍感兴奋的地方（读者最终也会理解这种方式并对它情有独钟）。从逻辑的角度讲，对象的概念只有用指针才能最好的描述，正像种下一棵树苗，我们只是让它具有生长的可能，而不是让它生长。

C++ Builder 实际上是 C++ 语言的一种版本，但它与传统的 C++ 语言有着很大的差别。一个 C++ Builder 程序首先是应用程序框架，而这一框架正是应用程序的“骨架”。在骨架上即使没有附着任何东西，也仍然可以严格地按照设计运行。程序员的工作只是在“骨架”中加入自己的程序。缺省的应用程序是一个空白的窗体（Form），我们可以运行它，结果得到一个空白的窗口。这个窗口具有 Windows 窗口的全部性质：可以放大缩小、移动、最大化和最小化等，但我们却没有编写一程序。因此，可以说应用程序框架通过提供所有应用程序共有的东西，为用户应用程序的开发打下了良好的基础。C++ Builder 已经为读者做好了一切基础工作——程序框架就是一个已经完成的可运行应用程序，只是不处理任何事情。我们所需做的，只是在程序中加入完成所需功能的代码而已。

在空白窗口的背后，应用程序的框架正在等待用户的输入。由于我们并未告诉它接收到用户输入后作何反应，窗口除了响应 Windows 的基本操作（移动、缩放等）外，它只是接受用户的输入，然后再忽略。C++ Builder 把 Windows 编程的回调、句柄处理等繁复过程都放在一个不可见的覆盖物下面，这样读者可以不为它们所困扰，轻松从容地对可视化部件进行编程。

关于《C++ Builder 实用教程》

首先，本书中的内容没有一项是特别的技术。关于 C++ Builder 语言的精确定义可以从各种各样的资料上得到，其中最重要的一个就是 C++ Builder 的联机帮助系统。当然，只有理解了事情怎样做以及为什么这样做时，这些介绍才能有所帮助，否则语言描述本身就

太抽象了。

其次，本书是基于图形用户界面的，特别是基于 Windows 95/NT 4.0 界面，这样就很容易掌握一些最重要和最基本的概念。本书提出的概念不是按照传统的教学顺序组织的，而是按照对读者是否有用的顺序组织的。读者会发现，这样的组织方式将使学习过程更有效。

本书的宗旨是通过大量或深或浅的实例讲解 C++ Builder 的内容。贯穿整本书的是读者应该遵循什么样的步骤，应该采取什么样的思考方法，以及如何将自己的思考转化为正确的程序代码。因为只有在“干中学”才能有更大的收获，所以每一章中都包括了一些例子，这些例子有的难，有的易，而且在很多情况下还有一定的关联。我们希望读者在阅读本书的同时也上机实践。特别要指出的是，本书中所有的例子都是与版本无关的，它们既可以在 C++ Builder 1 中运行，也可以不加修改地在 C++ Builder 3 中运行。

每一章后面都包含一些问题与练习，它们的答案通常可以在书中找到线索，但也有一些练习必须通过上机实践才能完成。笔者深信，只有实践才是出真知的最佳途径。当然我们也要指出，有一些练习是稍难的。笔者同样深信，只有独立完成稍难一点点的问题才是迅速提高的唯一途径，太易的练习没有任何意义，而太难的练习则失去了实践的价值。对于较难的练习我们将给出一定的提示，请读者务必完成它们。

本书是如何组织的

本书是这样组织的：

在本书的第一部分，我们将简要介绍 C++ Builder 的环境，由本书的前 4 章组成。第 1 章介绍事件、消息与面向对象编程的概念。第 2 章将介绍 C++ Builder 的可视化开发环境。第 3 章将介绍 C++ Builder 的基本部件，而第 4 章则讨论如何使用 C++ Builder 的工程管理和设计工具。

本书的第二部分着重讨论 C++ Builder 的 C++ 语言，由第 5 章到第 11 章组成。第 5 章的目的是让读者留下对 C++ 语言的一些印象。第 6 章将讨论 C++ 语言的字符串和数字处理。贯穿第 7 章的是分支与循环程序要注意的一些问题，这类代码在实际的编程中占有至关重要的地位。第 8 章主要讨论函数和编译指令，第 9 章则把注意力集中在 C++ 语言的一个重点——指针问题的处理上。第 10 章将讨论用户自定义的数据类型，它们是编写程序时必不可少的预备知识。面向对象是第 11 章的主题，它们构成了程序代码的基本骨架，正像窗体构成应用程序的基本骨架一样。将讨论对象编程的知识，我们通过一个有趣的快算 24 的实例来深入浅出的介绍对象的基本概念。

本书的第三部分是一些真正有用的应用程序实例，通过这些实例读者可以学习到很多编写实际应用程序时所需要掌握的问题。第 12 章则通过文本编辑器的实例介绍如何创建多文档界面的应用程序。图形图像的处理方法是第 13 章的内容，第 14 章将讨论文件输入输出的基本知识，毕竟没有输入输出的程序是没有任何意义的。最后的第 15 章到第 17 章是本书的重中之重，详细地讨论如何使用 C++ Builder 开发数据库应用程序，并给出了一个较复杂也非常实用的程序实例。本书的附录介绍了 C++ Builder 3 的环境和与 C++ Builder 的异同。

本书的内容由浅入深，没有阅读前面的内容就直接学习后面的章节可能会给读者带来一定的麻烦，当然对相应内容有相当程度了解的读者又另当别论了。

本书的前两章由林杜执笔，第3章和第4章由费广正执笔，第5章由王程铭执笔，第6章和第8章由王志海执笔，第7章和第9章由乔木执笔，第10章和第11章由蒋培执笔，第12章由蔡宏执笔，第13章和第14章由汪巨涛执笔，第15章和第16章由乔林执笔，第17章和附录由陈爱华执笔。书中如有错误当是作者水平有限，恳请读者谅解。

本书的读者对象

本书是介绍 C++ Builder 和 C++ 编程语言的基础读本。即使读者没有任何计算机语言的知识，也可以阅读本书。如果读者已经使用过诸如 Visual Basic 或 Delphi 等可视化产品，那么读者将会发现本书是一本可以快速掌握 C++ Builder 编程技术的书。

有一定的 C 语言编程知识是好的，但即使没有也没有太多的关系，读者完全可以通过本书的学习领会 C++ 语言的精髓。C 和 C++ 是不同的，我们不打算再介绍 C 语言以及 C 与 C++ 的不同之处，本书中所有的语言要点都是 C++ 的，当然在 C 语言中可能也有同样的内容。

为了理解本书的内容，读者应该熟悉 Windows 95/NT 4.0 操作系统的基本知识。但即使如此，它们也不是必要的。对读者最低限度的假设是能够知道 Windows 95/NT 4.0 的基本操作，能够知道如何使用鼠标和一些常用硬件设备就足够了。

使用过 Delphi 的读者将会发现向 C++ Builder 中移植 Delphi 的代码是最容易不过的事情，C++ Builder 本身就可以编译 Delphi V2.01 以上的源代码，我们甚至不需要任何改动！

我们再一次地重申，本书的对象是 C++ Builder 的初学者和中级读者。对于初学者而言，本书的第1部分和第2部分是极好的学习材料，而对于中级读者来说，在本书的第3部分肯定能发现很多感兴趣的内容。

“边做边学”的方法是最有效的。每学完一个例子，尝试着改变一点点，或者添加一点东西，并改变一些代码将帮助读者体验进步和成功的乐趣。

目 录

序

第一部分 环境篇

第 1 章 事件与面向对象	1
1.1 人与计算机	1
1.2 我要控制你,而不是你来控制我	2
1.3 对象与类的基本特征	3
1.3.1 何谓对象	3
1.3.2 Windows 95/NT 中的对象	4
1.3.3 对象的属性	4
1.3.4 对象的行为	6
1.4 小结	6
第 2 章 C++ Builder 可视化开发概念	7
2.1 安装 C++ Builder	7
2.2 C++ Builder 可视化编程环境	8
2.2.1 加速条	9
2.2.2 对象观察器	11
2.2.3 窗体设计器	12
2.2.4 代码编辑器	13
2.2.5 使用 C++ Builder 的代码编辑器	14
2.3 一个简单的用户界面	18
2.3.1 选取部件加入到窗体中	19
2.3.2 部件的调整与对齐	19
2.3.3 保存所做的工作	22
2.3.4 运行工程	22
2.4 改变对象的属性	22
2.4.1 用 Properties 页改变部件的属性值	22
2.4.2 设置窗体的缺省按钮	23
2.4.3 汉化界面及字体选取	23
2.4.4 使用图像编辑对话框	24
2.5 添加事件处理过程	25
2.5.1 为用户操作建立“事件”	25
2.5.2 使用颜色编辑对话框	25

2.6 使用联机帮助 Help	26
2.7 小结	26
第 3 章 C++ Builder 基本部件	27
3.1 使用 C++ Builder 常用部件	27
3.1.1 常用文本部件	27
3.1.2 按钮和检查框部件	28
3.1.3 滚动部件	29
3.1.4 分组和分界部件	29
3.1.5 网格和表格部件	30
3.1.6 图形和图像部件	30
3.1.7 关系图和文件列表部件	31
3.1.8 多媒体部件和 OLE 部件	32
3.2 使用 C++ Builder 不可见部件	32
3.2.1 菜单部件	32
3.2.2 计时器部件	34
3.2.3 对话框部件	35
3.3 小结	36
第 4 章 使用 C++ Builder 的工程管理和设计工具	37
4.1 创建多窗体工程	37
4.1.1 创建一个含有 About 框的例程	37
4.1.2 指定自动创建窗体	38
4.2 使用工程管理器	39
4.2.1 工程文件的组成	39
4.2.2 使用 Project Manager 进行工程管理	40
4.3 使用窗体模板和对话框向导	41
4.3.1 使用窗体模板创建窗体	41
4.3.2 存储自己的窗体模板	41
4.3.3 使用对话框向导 Dialog Wizard	42
4.4 使用工程模板和应用程序向导	42
4.4.1 使用工程模板	43
4.4.2 使用应用程序向导 Application Wizard	43
4.5 小结	44

第二部分 语言篇

第5章 使用 C++ Builder 语言	45	6.2.1 使用整数时会遇到的问题	89
5.1 C++ Builder 语句与语句块	45	6.2.2 类型转换	90
5.1.1 空白与语句	45	6.3 计算器——字符串与数的实例	92
5.1.2 表达式语句和空语句	46	6.3.1 我们要干什么	92
5.1.3 块与复合语句	47	6.3.2 设计思路	93
5.2 C++ Builder 的标识符与保留字	47	6.3.3 最终实现	94
5.2.1 标识符	47	6.4 小结	98
5.2.2 保留字	48	6.5 问题与练习	99
5.2.3 操作符与表达式	49	第7章 条件分支与循环	100
5.2.4 定义标识符的良好习惯	56	7.1 条件分支语句	100
5.3 常量与变量	57	7.1.1 布尔类型	100
5.3.1 对变量名的限制	57	7.1.2 if 语句	101
5.3.2 有关常量和变量的一些问题	58	7.1.3 switch 语句	103
5.3.3 变量的声明与作用域	59	7.1.4 嵌套的 if 语句与 switch 语句	104
5.3.4 使用变量	60	7.2 循环语句	105
5.4 C++ Builder 中的类型	61	7.2.1 跳转与无跳转编程	106
5.4.1 基本类型	61	7.2.2 for 循环语句	106
5.4.2 字符串类型	63	7.2.3 while 和 do while 循环 语句	107
5.4.3 数组类型	64	7.2.4 break 和 continue 语句	108
5.4.4 结构类型与联合类型	64	7.2.5 几种循环的比较	109
5.4.5 枚举类型	66	7.3 使用 TCanvas 类的实例	109
5.4.6 指针类型	66	7.3.1 TCanvas 类	109
5.5 类与对象	69	7.3.2 设计应用程序的界面	112
5.5.1 面向对象的程序设计	69	7.3.3 最终实现	114
5.5.2 C++ Builder 中的部件类与类 类型	70	7.4 数组	119
5.5.3 类与对象的属性与行为	73	7.4.1 一维数组的定义与引用	119
5.5.4 使用类	74	7.4.2 二维数组的定义与引用	121
5.5.5 创建非可视化类	76	7.4.3 TStringGrid 部件	122
5.6 注释你的代码	79	7.4.4 TStrings 类	124
5.7 小结	80	7.5 小结	127
5.8 问题与练习	80	7.6 问题与练习	127
第6章 字符串与数字	82	第8章 函数和编译指令	129
6.1 字符串类型与操作	82	8.1 函数和类方法的定义	129
6.1.1 字符串类型	82	8.1.1 函数和类方法的定义	129
6.1.2 字符串操作	83	8.1.2 内联函数	131
6.1.3 AnsiString 类	86	8.2 函数和类方法的参数	132
6.1.4 TLabel 部件与字换行	89	8.2.1 形式参数和实际参数	132
6.2 数字类型与操作	89	8.2.2 函数和类方法的返回值	133
		8.2.3 数组作为函数和类方法的参数	134
		8.3 函数和类方法的调用	135

8.3.1	调用 C++ Builder 方法和函数 的简单例程	135
8.3.2	调用 C++ Builder 的含参过程	137
8.3.3	函数和类方法的递归调用	139
8.4	编译指令	139
8.4.1	嵌入指令	140
8.4.2	宏定义	140
8.4.3	条件编译指令	142
8.5	小结	144
8.6	问题与练习	144
第 9 章	指针与引用	145
9.1	指针的概念	145
9.2	指针运算	146
9.2.1	指针的算术运算	146
9.2.2	指针的关系运算	146
9.2.3	指针的赋值运算	146
9.2.4	指针和数组的关系	147
9.3	多级指针	148
9.4	指针和数组	149
9.4.1	指针数组	149
9.4.2	指向数组的指针	151
9.5	指针和动态内存分配	152
9.6	引用	154
9.6.1	引用说明	154
9.6.2	引用参数	156
9.6.3	返回引用的函数	156
9.7	类型定义	157
9.8	指针和函数	158
9.8.1	指针作为函数的参数	158
9.8.2	返回指针的函数	160
9.8.3	指向函数的指针	161
9.9	特殊的指针与复杂说明	162
9.9.1	void 类型的指针	163
9.9.2	指针和 const 关键字	163
9.9.3	复杂说明	164
9.10	小结	165
9.11	问题与练习	166
第 10 章	结构	167
10.1	结构说明	167
10.2	结构和指针	168
10.3	结构与函数	169

10.4	复杂结构与链表	171
10.4.1	结构作为另一个结构的成员	171
10.4.2	链表	172
10.5	时间表应用程序	172
10.6	小结	180
10.7	问题与练习	181
第 11 章	面向对象的程序设计基础	182
11.1	构造函数和析构函数	182
11.2	数据成员的初始化	187
11.3	赋值	190
11.4	const 修饰的成员函数	191
11.5	重载	192
11.5.1	类中函数名的重载	192
11.5.2	运算符重载	194
11.6	继承	196
11.6.1	父类和子类	197
11.6.2	访问控制	200
11.6.3	支配规则	205
11.6.4	再谈类的构造函数和析构 函数	207
11.7	完整的快算 24 代码	208
11.8	小结	225
11.9	问题与练习	226

第三部分 应用篇

第 12 章	创建 MDI 应用程序——文本编辑器实例	227
12.1	多文档界面	227
12.1.1	创建父窗体	228
12.1.2	创建子窗体	229
12.1.3	窗体菜单的融合	232
12.2	文本编辑部件及应用	234
12.2.1	TEdit 部件	234
12.2.2	TMemo 部件	235
12.2.3	TRichEdit 部件	235
12.3	添加属性与方法	236
12.3.1	添加父窗体的事件处理代码	236
12.3.2	添加子窗体的事件处理代码	238
12.4	通用对话框的使用	244
12.4.1	文件打开对话框部件和文件保存对话框部件	244

12.4.2 查找对话框部件	247	14.4 非缓冲文件	302
12.4.3 替换对话框部件	249	14.5 C++ Builder 封装的 Windows 文件 管理标准函数	303
12.4.4 字体对话框部件	249	14.6 文件控件的应用	305
12.5 文件打印	251	14.7 小结	305
12.5.1 TPrinter 对象	251	14.8 问题与练习	305
12.5.2 打印对话框部件	253	第 15 章 使用 C++ Builder 开发	
12.5.3 打印机设置对话框部件	255	数据库应用程序	306
12.6 对文本编辑器的改进	255	15.1 数据库系统概述	306
12.6.1 添加工具栏	255	15.2 C++ Builder 的数据库特性及功能 简介	307
12.6.2 跟踪子窗体	257	15.2.1 C++ Builder 的数据库特性	307
12.6.3 使用 Tag 属性	259	15.2.2 C++ Builder 可以访问的数据 源 (DataSource)	309
12.6.4 再谈 GroupIndex 属性	261	15.3 C++ Builder 数据库的体系结构	309
12.6.5 菜单的动态控制	262	15.3.1 数据访问部件	310
12.7 小结	264	15.3.2 数据控制部件	312
12.8 问题与练习	264	15.3.3 数据库窗体向导	314
第 13 章 图形图像处理办法	266	15.4 C++ Builder 数据库应用程序的 开发方法和步骤	314
13.1 常用图形对象	266	15.5 简单的基于单表的数据库应用	316
13.1.1 TPen 类	266	15.5.1 选择相关部件	316
13.1.2 TBrush 类	268	15.5.2 设置部件的属性	316
13.1.3 TColor 类型	269	15.5.3 运行程序	317
13.2 画图程序	270	15.6 利用 TDBNavigator 部件创建存取 程序	318
13.2.1 响应鼠标事件	270	15.6.1 创建应用程序窗体	319
13.2.2 在窗体中加入工具栏	273	15.6.2 使用 TDBNavigator 部件移动 记录指针	319
13.2.3 绘图功能的实现	274	15.6.3 定制 TDBNavigator 部件	320
13.3 剪贴板	286	15.7 创建主要/明细型数据库应用	321
13.3.1 使用剪贴板传输文本	286	15.8 字段类的使用	323
13.3.2 剪贴板类	287	15.8.1 字段类的类型	323
13.3.3 利用剪贴板传输图像	289	15.8.2 创建永久性的字段类	323
13.4 常用图像类	290	15.8.3 设置字段类的属性	324
13.4.1 TGraphic 类	290	15.8.4 字段类的访问	327
13.4.2 TPicture 类	290	15.9 数据库的查询	329
13.4.3 TImage 部件	291	15.9.1 使用 GotoKey 方法查找数据 记录	329
13.4.4 TBitmap 类	291	15.9.2 使用 FindKey 方法查找数据库 中的记录	330
13.5 图像类的应用	291		
13.6 小结	297		
13.7 问题与练习	297		
第 14 章 文件	298		
14.1 标准文件类型和文件的打开与 关闭	298		
14.2 文件的读写	299		
14.3 文件的定位	301		

15.9.3 使用 GotoNearest 和 FindNearest 执行不精确查找	331	16.3.3 数据集的导航	340
15.10 修改数据库中的记录	333	16.3.4 数据集集中的数据维护	341
15.10.1 Edit 方法和 Post 方法	333	16.3.5 数据集部件与数据浏览部件的 连接	343
15.10.2 插入和删除记录	334	16.3.6 数据集部件的相关事件	344
15.11 输入数据的有效性验证	336	16.4 TTable 部件	344
15.12 小结	336	16.4.1 TTable 部件的主要属性	345
15.13 问题与练习	336	16.4.2 TTable 部件的方法及应用	346
第 16 章 数据的访问	337	16.5 TDataSource 部件	348
16.1 C++ Builder 数据访问部件的层次 结构	337	16.5.1 TDataSource 部件的属性	348
16.2 TSession 部件	338	16.5.2 TDataSource 部件的事件	348
16.2.1 TSession 部件的重要属性	338	16.6 小结	350
16.2.2 TSession 部件的方法	338	16.7 问题与练习	350
16.3 数据集部件 TDataSet	339	第 17 章 数据库应用程序实例	351
16.3.1 数据集部件的几种状态	339		
16.3.2 数据集的打开与关闭	340		

第四部分 附录

A C++ Builder 3 环境	419
--------------------------	-----

第一部分 环境篇

第 1 章 事件与面向对象

与其他改变人类生活的重大技术相比，计算机问世的时间并不长，但它同时在很短的时间里却获得了巨大的进展。它们体积越来越小，速度越来越快，能力越来越强。然而机器毕竟是机器，没有人的参与和使用，再好的机器也是没有什么价值的。人的参与，也就是计算机操作员的参与，决定了计算机如何进行工作，以及工作的效率到底如何。不同的操作员执行计算机操作任务的效率是不同的，一个受过良好训练的操作员操作计算机的速度肯定会比没有受过训练的操作员要好得多，最简单的例子莫过于经过培训的打字员的速度是笔者在录入本书时的速度所不能比的。

这其实提出了一类重要的问题：随着计算机的日益普及和广泛使用，越来越多的人将接触到计算机或必须使用计算机进行工作。如果没有经过严格的培训，这些计算机操作和使用的门外汉如何才能较快地进行工作呢？计算机的应用程序正变得越来越复杂而庞大，这样的程序应该向用户或使用提示什么样的信息以方便用户的操作呢？在这样的大环境中，如何使用应用程序与用户的交互更简单明了就是用户界面领域研究的课题。

1.1 人与计算机

自计算机首次问世以来，它的功能就几经变迁，早已与原先的使用性质存在非常大的差异了。原先的计算机程序是一系列的记录在纸带设备上的孔洞，要输入的信息就存储在纸带上。鼠标是没有的，甚至显示器也是没有的（今天的读者是很难想象没有鼠标和显示器的计算机的）。如果程序员要使用计算机进行工作，就必须像操作万吨水压机一样装料、搬动电气开关、然后下料。

这类原始的操作方式延续了相当长的时间，即使当 DOS 操作系统和个人计算机广泛流行起来之后也顽固地生存了好几年。我们不能不承认 DOS 操作系统的意义是不可比拟的，尽管 DOS 操作系统的用户界面在很大程度上按照上述交互性能非常差的方式进行工作。

平心而论，改善用户界面的工作是非常困难的。即使是一点点的进步也充满了许多杰出的研究人员和计算机专家的不懈努力。想象一下，键盘通信刚刚被用户广泛接受，显示器的使用还刚刚起步（即使使用了显示器，它们也与我们今天使用的显示器差别很大），打印机的质量不高（也许会比机械式打字机稍好一点？），使用这样的硬件设备，用户界面能好到哪里去？最典型的例子莫过于：

计算机提示：

请输入姓名？

然后用户输入自己的姓名：

Qiao

之后计算机显示：

你好，Qiao！

（相当不错，计算机知道我的名字呢！）

下面有四种选择，请输入选择前面相应的数字：

- 1) 开始读入数据
- 2) 进行计算并将计算结果输出到文件中
- 3) 哈哈，老板不在，让我们先玩一会
- 4) 退出系统运行

你的选择是：

（居然是菜单界面！）

此时我们输入 5，系统将告诉我们输入错误请继续输入新信息。在我们输入了正确的数字之后，系统将进行下一步的工作，然后又是类似上面的界面。

上面的例子是我们假想的，请不要认为它过于幼稚，实际上有相当多的用户界面还不如它呢！像这样程序的编写总是大同小异的，好的程序会在某一点上终止运行，而坏的程序会怎么样，玩过拙劣的 DOS 游戏的读者肯定深有体会。

这样的用户界面被富有经验的程序员使用了很多年，直到计算机成为普通人的好帮手。没有多少计算机经验的用户肯定不喜欢这样难以驾驭的系统，用户总是希望能够完全控制程序的运行。

1.2 我要控制你，而不是你来控制我

用户控制程序的概念尽管很显而易见，但却要求大量的硬件和软件的技术支持。事实上程序员往往会发现编写一个控制性差一些的程序要比编写一个控制性好的程序还要简单一些。DOS 虽然与用户交互，但它的界面往往不太友好。一般来说，计算机系统的资源总是应该由操作系统来控制的，但在 DOS 环境下，一旦运行某个应用程序，该应用程序将接管大多数系统资源。对用户来说，操作系统此时在干什么就不知道了。一旦应用程序出现致命的错误，结果往往只有一种——死机。

Windows 操作系统的出现彻底改变了人们对个人计算机的印象。在 Windows 3.x、Windows 95/NT 操作系统的环境下，用户几乎总是可以脱离某个应用程序，将程序隐藏在后台执行，然后在未来的某个时候返回该应用程序。从这样的变革中可以看出，操作系统和应用程序一起进化了。那么我不禁要问，是什么使这一变革成为可能的呢？

导致这种变革的原因是事件（event）。在计算机领域，我们称实际发生的某件事称为一个事件。按一次键盘是一个事件；移动鼠标是一个事件；计算机内部时钟的每次变化是一个事件。

事件对程序员意味着什么呢？假设存在这样的程序：

- 1) 程序开始运行；
- 2) 做某件事；
- 3) 从用户处得到输入；
- 4) 根据用户的输入做相应的事；
- 5) 结束运行。

整个程序的运行路径是不可改变的。计算机在某一时刻等待用户的输入，用户没有输入就不能进行其他的操作。使用事件的方法重新思考上面的程序，我们有：

- 1) 程序开始运行；
- 2) 如果某个事件发生，则处理之；
- 3) 否则继续当前的活动；
- 4) 如果用户没有命令程序结束，继续监视下一个事件；
- 5) 否则结束程序的运行。

在这样的程序里，程序不关心的事件即使发生也将被忽略。

1.3 对象与类的基本特征

事件是用户界面变革的基础，而对象是事件的最基本原动力。从概念上说，对象是开发系统中任何被观察到的实体。在构造应用系统的过程中，程序员分析问题领域，并对解决问题所需要的组件形成一个观察（即产生一个抽象的对象表示）。我们经常说对象是真实世界的模型就是这个道理。

相比较而言，面向对象的程序设计方法的最关键之处是匹配问题求解领域，对象的抽象概念支持应用程序开发者所需要的功能；结构化的程序设计方法提供了与基础硬件结构非常密切的一组抽象，因此我们需要做更多的工作以生成应用程序。

类是面向对象技术的进一步深化。支持类的目的是提供一种基本的分类形式。类通过允许创建对象的组支持分类来支持这些对象共享完全相同的行为，它是创建对象的模板，包含了创建对象的属性描述和行为的定义。

总之，没有面向对象的基本特征就没有可视化的编程方法。

1.3.1 何谓对象

面向对象的程序设计（Object - Oriented Programming，简记为 OOP）是 C++ Builder 诞生的基础。OOP 立意于创建软件重用代码，具备更好地模拟现实世界环境的能力，这使它被公认为是自上而下编程的杰出代表。它通过给 C 语言中加入扩展语句，把函数“封装”进 Windows 编程所必需的“对象”中。面向对象的编程语言使得复杂的工作条理清晰、编写容易。说它是一场革命，不是对对象本身而言，而是对它们处理工作的能力而言的。对象并不与传统程序设计和编程方法相兼容，部分面向对象的方式往往会使情况更糟。除非整个开发环境都是面向对象的，否则对象产生的好处还没有带来的麻烦多。而 C++ Builder 是完全面向对象的，这就使得 C++ Builder 成为一种触手可及的促进软件重用的开发工具，从而具有强大的吸引力。

一些早期的具有 OOP 性能的程序语言如 Smalltalk，虽然具有面向对象的特征，但不能轻松地刻画可视化对象，与用户交互能力也较差，程序员仍然要编写大量的代码。C++ Builder 的推出大大改变了这种现状。我们不必自己建立对象，只要在提供的程序框架中加入完成功能的代码，其余的都交给 C++ Builder 去做。欲生成漂亮的界面和结构良好的程序丝毫不必绞尽脑汁，C++ Builder 将帮助我们轻松地完成。它允许在一个具有真正 OOP 扩展的可视化编程环境中，使用它的面向对象语言。这种革命性的组合，使得可视化编程与面向对象的开发框架紧密地结合在一起。

1.3.2 Windows 95/NT 中的对象

上面的介绍也许会令部分读者感到晦涩难懂，不过在使用 Windows 95/NT 的过程中，读者可一直在使用对象。在桌面上出现的任何一个图标都是对象，如图 1-1。请注意当用户双击图标时，窗口是如何打开的，这个过程意味着这些对象是相互关联的。

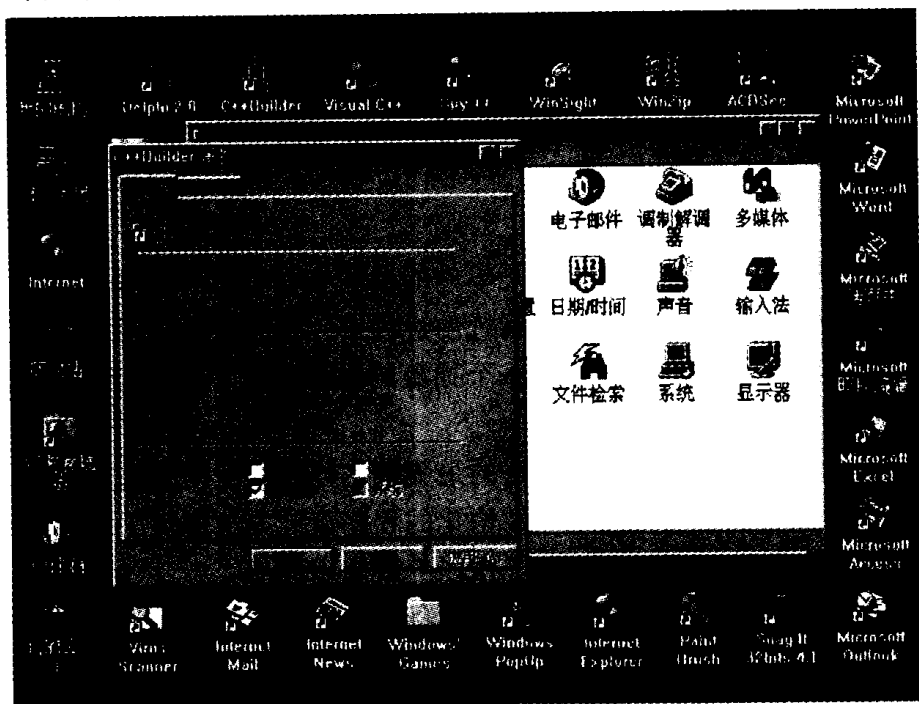


图 1-1 Windows 95 桌面系统包含的对象以及与其他对象的关联

Windows 95/NT 中的大多数窗口都可以改变大小，放大缩小，或者在屏幕上移动。这些对象也是关联的，尽管在其内部有许多不同。

那么窗口内部有什么呢？系统菜单图标、最大化按钮和最小化按钮等。它们的操作方式是相同的，不管是运行 PBrush 还是运行 C++ Builder。所有这些内容都可以看作对象，同时这些对象属于同一类，例如窗口类或菜单类等。前面已经说过，这些对象按类分组，并且可以通过属性进行区分。

1.3.3 对象的属性

现实世界中没有两个对象是相同的，Windows 95 中的对象也如此，如图 1-2。请注意任务栏上打开的多个应用程序。在最大化的情况下，我们不可能看到两个以上的应用程序，因为其中的一个应用程序覆盖在另一个之上。

Windows 95/NT 使用桌面对象管理屏幕的外观，虽然从一定角度上说，Windows 95/NT 的桌面对象的缺点不少，但无论如何，它为我们创建了一个屏幕的视觉印象。用户可以重新安置这些对象，甚至可以让对象一个挨一个排列起来，而且任何两个对象都不会显示在同一个地方。为什么呢？因为位置是对象的属性，所以不同的对象是不可能出现在桌面上的同一个地方的。

现在我们重新排列这些应用程序，使它们都出现在屏幕上，如图 1-3 所示。请注意这些窗口中只有一个激活的窗口，我们称当前激活的窗口为焦点窗口。可以使用 Windows 95/

NT 提供的切换方法在不同的窗口间切换并分别激活它们。

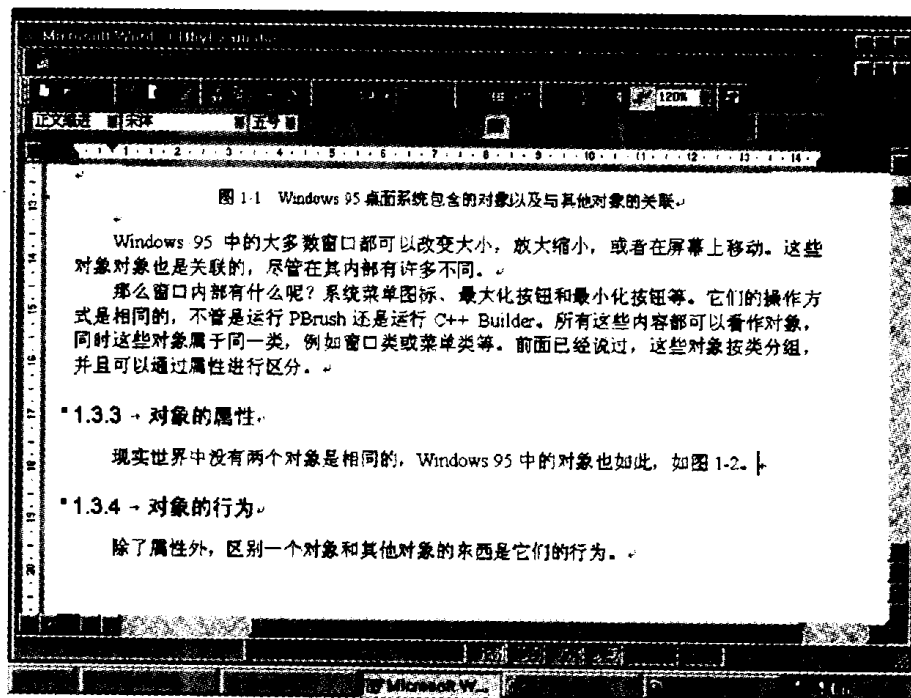


图 1-2 在 Windows 95 中同时打开多个应用程序

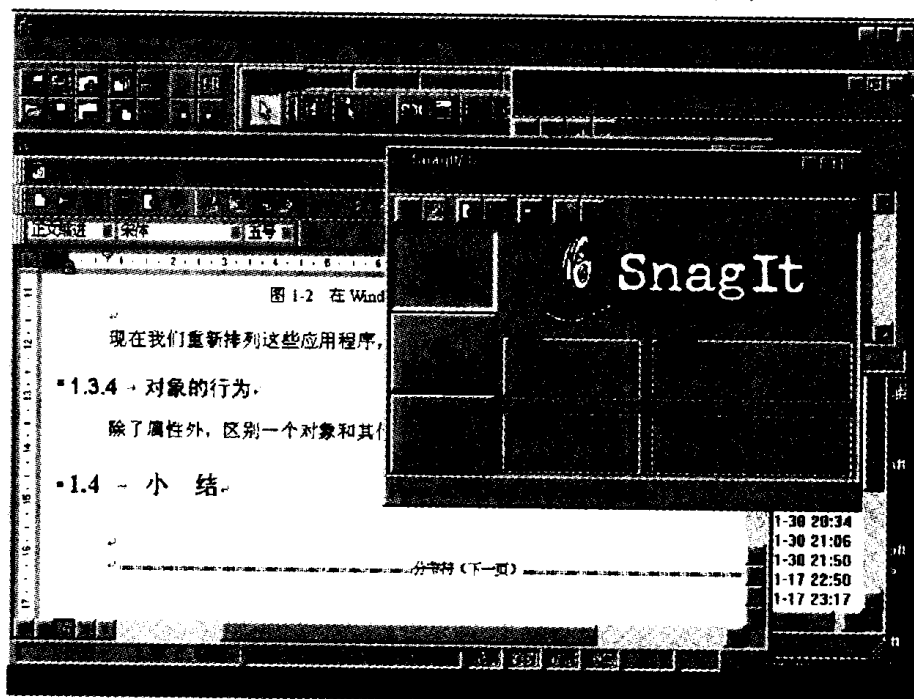


图 1-3 在桌面上叠放多个打开的应用程序

既然窗口对象可以聚焦，那么窗口中的对象能不能聚焦呢？答案是肯定的。我们单击桌面上的“我的电脑”图标，显示“我的电脑”窗口，单击它的驱动器 C 图标，该对象就变成了焦点对象，如图 1-4 所示。

其实 Windows 95/NT 和 C++ Builder 将处理所有这类工作，我们提及它的目的是让读者了解 Windows 95/NT 的用户界面设计达到了什么样的程度。在程序设计的过程中，程序员可以使用 Windows 95/NT 的标准对象，然后改变它们的属性以形成新的程序。

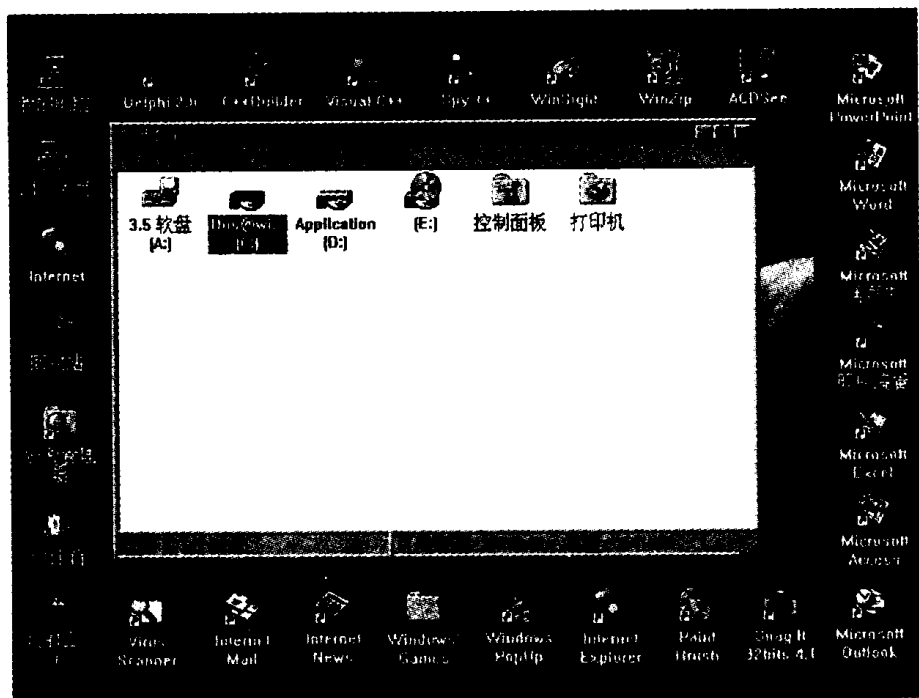


图 1-4 “我的电脑”窗口中的焦点图标对象

1.3.4 对象的行为

除了属性外，区别一个对象和其他对象的东西是它们的行为。对象的行为有时也称对象的方法，是指该对象可以执行的动作或操作。Windows 95/NT 和 C++ Builder 为对象提供了许多缺省行为，在一般情况下我们只需要添加自己希望该对象要做的工作即可而不必改动我们不关心的行为。

1.4 小 结

对计算机而言，程序是一个指令序列，这些指令由人使用计算机编程语言编写而由计算机执行。编译程序的意义是将这些由人编写的程序翻译成机器可以识别的机器码。

现在的应用程序已经从不考虑用户对程序的控制进化到允许用户对程序进行完全的控制。程序通过对事件的识别让用户控制程序。所谓事件是指用户或程序应该响应的计算机做的任何事情。这里要特别说明一点，不做响应并不意味着真的不做响应，而是做“不做响应”的响应（这有些像诡辩，它的意思是忽视该事件而不是无视该事件）。

对象是真实世界的模拟，它们具有特定的属性与行为。