

王海滨 薛建林 徐涛 编著

计算机最新技术培训教材

PEKING UNIVERSITY PRESS

C++ Builder 4

学习教程



- 优秀的C++语言编辑器
- 快速开发与高质量代码的统一

北京大学出版社
<http://cbs.pku.edu.cn>



计算机最新技术培训教材

C++ Builder 4 学习教程

王海滨 薛建林 徐 涛 编著

1501045
北 京 大 学 出 版 社

北 京

内 容 简 介

C++ Builder 4 是 Borland/Inprise 公司最新发布的用于快速开发 Windows 应用程序的一种编程工具。它具有比同类产品更加强大的功能、更加友好的界面、更易学易用的操作方法, 因此受到广大编程者的欢迎。

通过学习本书, 读者将掌握面向对象的编程方法, C++ Builder 4 中 VCL 的层次关系和各种各样的部件工具, 并通过大量的实例分析, 使读者能轻松自如地设计自己的应用程序界面, 定制自己的部件, 对不同的应用程序进行数据交换, 访问数据库并进行输出, 开发各种多媒体应用程序等。

本书内容丰富、条理清晰, 具有很强的使用性和可读性, 既可作为初学者从入门到提高的教材, 又可作为不同层次的程序设计人员的实用参考书。

图书在版编目 (CIP) 数据

C++ Builder 4 学习教程/王海滨, 薛建林, 徐涛编著. —北京: 北京大学出版社, 2000.3
(计算机最新技术培训教材)
ISBN 7-301-01831-2

I. C… II. ①王…②薛…③徐… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 07024 号

书 名: C++ Builder 4 学习教程

著作责任者: 王海滨 薛建林 徐涛

责任编辑: 杨锡林

标准书号: ISBN 7-301-01831-2/TP·137

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电子信箱: xxjs@pup.pku.edu.cn

排版者: 南方立德 (Leader) 信息技术中心

印刷者: 河北省滦县印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 22.25 印张 554 千字

2000 年 3 月第 1 版· 2000 年 3 月第 1 次印刷

定 价: 35.00 元

前 言

我们对 C++ Builder 可以说是一见钟情。作为程序员，我们一致地认为：C++ Builder 是非常棒的！它远远地超越了这个时代的其它竞争产品，以至于我们都有些怀疑 Inprise/Borland 的那帮设计人员会不会是外星球的天才怪物？当然这不可能。

让我们来客观地比较一下：

VB 可以用来快速地开发 Windows 应用程序，不过其解释执行的代码效率使其性能大打折扣；并且，作为对 Basic 的扩充，我们实在怀疑这种语言能力的上限。

VC++ 编译环境生成的代码执行效率确实很高，不过其开发软件的过程实在无法使用“轻而易举”来形容，尤其是对于那些新手，头几个星期犹如地狱。

Java 吗？它很小，学习起来也不难，（谁叫它是从 C++ 中衍生而来的呢？）其跨平台的特性使得它几乎可以在所有的操作系统上工作。比如，从有限电视的机顶盒到电话乃至烤面包机。不过，基于一些商业和技术的原因，尽管有 Sun 公司给它撑腰，Java 仍然还不够成熟，并且基于 Java OS 的网络计算机从来就没有流行过。

至于 Power Builder 之类的工具，它们适用于客户/服务器应用程序的开发，本身与数据库结合得相当紧密，不过其代码的灵活性和编译生成的代码效率实在不能让人满意，而且，它们本身就不能算作一个通用的开发工具。

与上述这些工具相比较，我们看到 C++ Builder 在各方面都是十分优秀的。比如高度易用的可视化开发环境、编译后高效的代码、对客户/服务器数据访问的完整解决方案、强有力的异常处理机制以及对 Microsoft 的 DDE、OLE 和 ActiveX 的完善支持，尤其对网络还做了进一步的地优化处理。对于这么好的产品，即使是最挑剔的顾客也难以说些什么了。所以，我们一直希望把我们的感受介绍给更多的朋友，好的东西应当大家一起分享。这本书成全了我们的想法。

另外，我们不能不提到的是，Inprise/Borland 公司的另一大作 Delphi 5 为 C++ Builder 的姊妹（我们也搞不清楚，到底谁年长一些），两者都是非常优秀，它们的开发方式甚至界面都是如此的相似，以至于我们简直没有办法取舍。

一个比较好的建议是：

如果你熟悉 Pascal，并且偏爱这种具有严谨语言规范的风格，好了，去用 Delphi 吧，它不会让你失望的。

如果你更倾向于使用 C++ 的灵活控制，并且不是特别的粗心大意，不要犹豫了，C++ Builder 是你的首选。

关于这本书我们不想再说些什么，因为它已经在你的手里，读读前言，看看目录，然后随便翻上几页，相信你很快会得出你自己的结论。

本书包括哪些内容?

本书共分十三章，相关的内容如下：

第一章 概述——主要介绍了对象、事件、属性的概念，以及 C++ Builder 4 中应用程序的种类和文件的组成。

第二章 开发环境——在这一章里面，我们主要的精力放在对 C++ Builder 4 的工作界面的介绍上，你不妨把这看作是正式参加工作前的一个快速参观过程。毕竟，如果你熟悉了 C++ Builder 4 的工作环境，对整体的构架又能够有一个完整的理解，工作起来就会事半功倍。

第三章 开发工具——本章将介绍一些 C++ Builder 4 的工具软件，并且对你的开发环境做一番合理的设置。

第四章 C++ 的语法规则——正如本章标题所言，这一章的内容是为那些很长时间无暇复习 C++ 的人们准备的。C++ Builder 的语言基础仍然是 C++，你必须牢记这一点。尽管两者在外观包装、代码编写的方式上大相径庭，但是，它们底层的架构定义仍然是一致的，并且在最简单的语句上都遵循有关国际组织严格的定义。

第五章 高级开发概念——本章将学习一些高级开发的内容，比如，宏和指针等。在大多数情况下，这些概念和技巧让人感到枯燥无味。你必须有耐心，仔细地完成每一件事情。

第六章 部件——本章将简要地介绍 C++ Builder 4 提供给我们的各种部件。需要提醒的是，C++ Builder 4 之所以提供如此强大的部件，其目的就是为了鼓励重复使用代码，提高程序的工作效率。所以，熟悉 C++ Builder 4 中的各种部件是非常必要的。

第七章 表单——最实际的说法，表单就是一个窗口。它之所以被称为“表单”，很大程度上是出于商业因素的考虑（没有人希望自己的东西毫无特色，并且成为别人的附属品）。在这一章里，将介绍表单设计的规则、表单的属性、事件和常用的方法等令人激动的内容。

第八章 图形——任何一种程序都应当提供操作图形的能力。在本章里面将简单地介绍 C++ Builder 4 提供的图形部件及其使用方法。

第九章 MDI 应用程序——MDI 就是“多文档界面”（Multi Document Interface）的意思。你可以回溯到上一章的内容，实际上，表单就是 SDI（单文档界面）（Simple Document Interface）。这里将尝试做一个典型的 MDI 的应用程序，以从各个方面来理解 MDI 的妙用。

第十章 多线程设计——从某种角度来说，这是第一个真正的 Windows 应用程序。

第十一章 数据库开发——我们主要讨论访问和操作数据库的方法，以及报表输出的有关内容。

第十二章 多媒体——这是一个神奇的世界。我们将介绍 C++ Builder 4 中提供的多媒体部件，并且介绍一下多媒体的文件格式。

第十三章 制作安装程序——这里主要讨论有关应用程序的发布问题。

关于本书的一些说明

C/C++是一种高度结构化的语言，具有良好的语言风格。本书的示例将完全遵循其准则，并且在不影响排版的情况下，尽量保持良好的程序风格与可读性，在关键的程序代码段处加入详细的功能注释和说明。

本书全部示例已在简体中文版 Windows 98, C++ Builder 4 下编译调试通过。

我们希望读者认真读书中的示例，最好是边学习边实践，做到举一反三，这对熟练掌握和使用 C++ Builder 4 是大有好处的。

做好准备!

要完成本书的程序示例，至少应当具备以下的条件：

- ✧ 一台 Pentium100，至少 16MB 内存的多媒体计算机。
- ✧ 一个 4 倍速以上的光盘驱动器，以便于安装该软件，并且进行资料交流。
- ✧ Windows 95/98 中文或者英文操作系统。出于现实的原因，建议选用简体中文版的 Win 98，并且不要外挂第三方的中文平台，这样可以保证系统平台的稳定性。
- ✧ 各种驱动程序安装正确，没有不正常工作的设备。
- ✧ 最低 800×600 之分辨率。
- ✧ 至少 256 色显示能力。
- ✧ 最少剩余 200MB 的可用磁盘空间，用于安装 C++ Builder 4 开发环境和运行示例程序。

当然，如果有可能的话，建议配备不低于 PentiumMMX 200 的 CPU 和 32MB 以上的内存。根据我们的使用经验，这样的系统运行才比较流畅，不会让你在焦躁无聊的等待中放弃学习的努力。

本书由孙景利策划，王海滨、薛建林、徐涛主编。另外，林乐、高翔、刘子锐、黄少棠、黄瀚华、凌贤伍、瞿小玉、孙宝玉、黄昌明、王洪秀、田尊五、吴广志、刘世德、李小峰、汪春军、张志明、王国戟、朱黎、陈果、李剑、董团结、顾云飞、刘贤轶等参与了具体章节的编写。

当今世界科学技术发展很快，编著的时间非常短暂，加之编者水平有限，错误之处在所难免，敬请读者和同行批评指正。

编 者

2000 年 2 月

目 录

第一章 C++ Builder 4 概述	1
1.1 面向对象	2
1.2 对象	3
1.3 一切都是对象	4
1.4 Windows 下的程序开发	6
1.5 为什么选择 C++ Builder 4	7
1.6 对象的属性	11
1.7 事件和方法	16
1.8 消息处理机制	18
1.9 本章小结	21
第二章 开发环境	23
2.1 界面概述	24
2.2 主窗口	25
2.3 主菜单	26
2.4 快捷工具栏	36
2.5 部件板	38
2.6 对象检查器	40
2.7 表单设计器	41
2.8 代码编辑窗口	44
2.9 本章小结	46
第三章 开发工具	48
3.1 工程管理	48
3.2 集成调试器	54
3.3 设置好工作环境	61
3.4 对象存储器	65
3.5 辅助工具	66
3.6 本章小结	72
第四章 C++ 的语法规则	74
4.1 C++ 的由来	74
4.2 基本语法	75
4.3 语句	87

4.4 复杂数据结构	106
4.5 本章小结	114
第五章 高级开发概念	115
5.1 编译预处理	115
5.2 内置函数	119
5.3 字符串	120
5.4 指针	123
5.5 类	124
5.6 类的继承	136
5.7 多态性	144
5.8 C++ Builder 中类的层次	147
5.9 文件	149
5.10 变量的存储	153
5.11 参数的传递方式	158
5.12 函数重载和运算符重载	158
5.13 本章小结	162
第六章 熟悉工具——部件	163
6.1 Standard 选项卡	164
6.2 Additional 选项卡	168
6.3 Win32 选项卡	169
6.4 System 选项卡	172
6.5 Internet 选项卡	174
6.6 选项卡 Data Access	175
6.7 Data Controls 选项卡	176
6.8 QReport 选项卡	177
6.9 Dialogs 选项卡	178
6.10 Win 3.1 选项卡	180
6.11 Samples 选项卡	182
6.12 ActiveX 选项卡	183
6.13 公共的属性和事件	183
6.14 变量与对象的命名规则	184
6.15 命名的约定	185
6.16 本章小结	188
第七章 表单	189
7.1 设计优秀的表单	189
7.2 提高表单的可重用性	190
7.3 使得两个表单可以相互引用	191
7.4 表单的内部结构	195

7.5 表单属性	196
7.6 表单事件	211
7.7 表单方法	214
7.8 本章小结	215
第八章 图形	216
8.1 画布部件	216
8.2 画笔部件	229
8.3 画刷部件	232
8.4 位图部件	233
8.5 形状部件	234
8.6 图像部件	236
8.7 本章小结	238
第九章 MDI 应用程序	239
9.1 建立一个 MDI 应用程序	239
9.2 MDI 的属性、事件和方法	242
9.3 利用模板创建 MDI	243
9.4 本章小结	247
第十章 多线程设计	248
10.1 理解线程	248
10.2 新建一个线程	252
10.3 线程的同步和加锁	254
10.4 利用多线程排序	254
10.5 本章小结	261
第十一章 数据库的开发和报表	262
11.1 关于数据库	262
11.2 数据库部件	266
11.3 SQL 查询语言	285
11.4 几个常用的工具	289
11.5 报表的设计	299
11.6 本章小结	308
第十二章 多媒体	309
12.1 多媒体概述	310
12.2 开发一个多媒体应用程序	310
12.3 常见的多媒体文件格式	312
12.4 TMediaPlayer 部件	313
12.5 简易的声音播放程序	317

12.6 视频播放器	319
12.7 D3D 绘画	321
12.9 本章小结	334
第十三章 制作安装程序	335
13.1 制作安装程序	335
13.2 本章小结	346

第一章 C++ Builder 4 概述

为什么要用 C++ Builder 来开发 Windows 应用程序呢？这里有许多富有挑战性和无可辩驳的理由会使你选用 C++ Builder 作为开发平台：

如果你是初级程序员，你会发现 C++ Builder 易于学习，使用起来魅力无穷，并且它的功能极其强大。（难道有人敢怀疑 C++ 的功能和 Inprise/Borland 的技术？）总而言之，让人激动不已。

如果你有过使用其它开发工具（比如 Visual Basic 或者 Visual C++）开发 Windows 应用程序的经历，那么，你可能遇到过对于一些你要实现的东西难于跨越的障碍，比如感觉到陷入了消息句柄以及控制逻辑的汪洋大海而不知所措。对于这些问题，我们敢保证，绝对不会在 C++ Builder 中发生。它不会受限于任何的东西。它的为数众多的性能和优势可以由以下几个方面体现出来：

- ◇ 能够创建真正独立的可执行文件及动态链接库（DLL）。
- ◇ 风格优美，易于使用的可视化开发环境。
- ◇ 一种十分强大而且灵活的底层编程语言。
- ◇ 一流的代码编辑和工程管理工具。
- ◇ 通过使用模板及部件来支持你重用自己的代码。软件的重用性是程序设计的一个重要的方面。
- ◇ 可以通过自身创建的新部件来扩充 C++ Builder 的能力。
- ◇ 对所有 Windows 消息的响应能力，具有无可怀疑的兼容性（虽然它不是 Microsoft 公司的产品）。
- ◇ 完完全全的面向对象编程方式。

并且，C++ Builder 4 还提供了一些开发的工具，使你可以轻而易举地把 VC++ 的工程文件转换成为 C++ Builder 4 的工程文件，这样就保护了投资者的利益。毕竟，没有人愿意白白地在编程工具上丢掉一大笔投资！

其它的可视化开发环境：

除了 C++ Builder 以外，还有其它相当数量的可视化开发环境。在它们当中，最流行的有 PowerBuilder（Sybase 公司出品），Visual Basic 和 Visual C++（它们都是 Microsoft 公司出品）。就其开发风格而言，Visual Basic 和 C++ Builder 最为接近。而就底层语言而论，Visual C++ 与 C++ Builder 都采用 C++。

除了一些难度特别大的开发环境以外，一般的可视化开发环境进行可视化开发的方式几乎完全相同。当然，在如何创建可视化对象以及如何实现这些对象的细节方面，它们存在着一定的差别。但是，在一般的情况下，当你创建一个简单的表单的时候是体会不出这些差别的。

毫无疑问，所有这些可视化开发环境——尽管它们彼此之间存在着这样或者那样的

差别和不同，但它们都具有极大的相似性。如果你具有操作其它可视化开发环境的经验，那么，我们相信，你一定能够迅速地适应 C++ Builder 4 开发环境的风格和开发方式。

C++ Builder 采用的底层语言是 Borland 公司的 C++ 版本。实际上，在面向对象的开发方式兴起之前，Borland 公司的 C++ 是最为广泛使用的开发环境（在以后的章节里面，我们将把这种风格的 C++ 简称为 C++，而不考虑不同公司和版本之间的差别）。

C++ 是一种相当灵活、高效、简洁的语言，这些优点无需我们多费口舌。如果你是刚刚从 Basic/Visual Basic 转过来的话，可能会感到有些不适应，尤其是在变量的数据类型的声明和不同类型变量之间的转换上面。不过，C++ 这种良好的风格可能更加适合程序员的口味，你必须适应它，从而养成良好的工作习惯，这样可以大大节省你的时间和精力，并且减少那些不必要的错误。

有关 C++ 的语法基础，我们将在第四章再做详细的讨论。

✓ 本章包含如下的内容：

- ✧ 面向对象的含义。
- ✧ 对象。
- ✧ 一切都是对象。
- ✧ Windows 下的程序开发。
- ✧ 为什么选择 C++ Builder？
- ✧ 对象的属性。
- ✧ 事件和方法。
- ✧ 消息处理机制。
- ✧ 本章小结。

1.1 面向对象

在讨论对象的时候，首先让我们来回顾一下计算机的基本原理。

当你用计算机来解决现实世界中的问题的时候，你需要使用计算机程序设计语言来对现实世界中需要求解的问题加以描述，然后由你的程序编译器把这些描述翻译成为 CPU 能够识别的二进制代码并使之运行求解。

现在，人们主要使用的描述方法分为两种：过程式描述和对象式描述。

采用过程式方法描述现实世界中问题的时候，程序员把该问题分解成为一系列的步骤并按顺序执行。与之相对应的是，在计算机中程序员使用一系列按顺序执行的指令来描述该问题的解决途径。

比如，在屏幕上画一个圆，过程式的描述方法如下：

- (1) 存当前的屏幕状态。
- (2) 把屏幕设置为图形状态。
- (3) 按照程序的规定，以某一点为圆心，按照给定的半径来画一个圆形。
- (4) 给该圆填充上指定的颜色。
- (5) 清除屏幕。

(6) 回到原始的屏幕状态。

需要说明的是,当处理简单问题的时候,使用过程式的描述方法是非常有效的。但是,当我们面对的是复杂问题的时候,就会显得力不从心。比如,你可以考虑实现一个与微软的 Word 97 类似的文字处理软件,这时你会发现很难使用一连串连续的过程来描述该过程。即使你一定要使用过程式的描述方法,编写出来的代码也非常庞大,并且难以读懂和维护。

在这种情况下,人们提出了面向对象(OOP)的描述方法。

简单地说,面向对象(OOP)设计是一种以数据(对象)及其接口(Interface)为重点的程序设计技术。我们以工程师建造大厦来打一个比喻:一位“面向对象”的工程师最重视的是他正在建造的大厦,其次才是他用来建造大厦的材料和其它的工具;相反地,一位“非面向对象”的工程师则是把主要的心思放在使用的建筑材料和辅助工具上面。

面向对象(OOP)的程序设计,即现在流行的可视化开发(Visual Development),主要的任务就是定义程序模块如何实现即插即用(PNP)的机制(这类似于 Windows 95/98 提供的硬件检测和安装功能,它能够大大提高硬件的兼容性,减少你在工作中的麻烦),你的应用程序的界面如何实现所见即所得(What You See is What You Get, 简称为 WYSWYG)的方法(这类似于 Word 的功能,不过的确是非常有用的)。

我们可以比较一下,相对于传统的过程式(或者称之为函数型)开发方式,可视化开发更像是在搭积木:一些基本的材料(比如基本的表单、菜单、按钮和对话框等)都已准备就绪,它们类似于现实生活中的木材、石头、钢筋等,你所要做的就是发挥你超群的想象力,运用你天才的头脑,进行严谨的设计,从而构造出一座宏伟的宫殿。

实际上,面向对象重视的是最终的结果——相应对象的行动并作出相应的处理;而过程式描述则更重视实现功能的过程以及是采用何种方法来实现的。

下面,我们来探讨一下对象的概念。

1.2 对 象

我们可以回忆一下,在现实的世界中,我们每时每刻看到的一切事物,都可以认为是对象。比如,一本书、一支钢笔、一条狗、任何一个人,等等。我们完全有理由认为,整个现实的世界就是由各种各样的对象构成的。在这个客观的世界里面,各种不同的对象之间的关系就是你要解决的问题。

让我们回到计算机中,只要你是在操作 Windows 系统(哪怕是最早的 Windows 3.1),那么你每时每刻都在接触一些鲜活的计算机对象。比如,在 Windows 95/98 中的窗体、按钮、滚动条和对话框等。这方面最让人着迷的恐怕就是电脑游戏中创造的角色了。比如,“帝国时代”(Microsoft 公司 1998 出品)中的农民、“红色警戒”(Westwood 公司 1996 年出品)中的各种坦克等,它们都是对象。

那么,它们是如何描述的呢?

对象本身可以描述为“一组逻辑上相关联的、可以从外部进行有限访问的数据”。它是面对计算机的,而后者就是一个抽象的处理二进制数据流的装置,对象内部的这些关

联的数据从计算机的角度来看，就变成了一些具体的具有外观的有生命有行为的对象了。我们以“红色警戒”中一辆坦克为例，来说明这个抽象的概念：

对象—>坦克

内部数据—>（如下）

Height=100

//坦克的高度为 100 英寸

Width=80

//坦克的宽度为 80 英寸

Color=Green

//坦克的外观为绿色

MaxSpeed=80

//坦克的最大时速为 80 公里/小时

Attack=100

//相对于一个普通的机枪手击中一次的攻击能力为 1，这辆坦克的攻击能力为 100

Code=1013

//坦克的编号是 1013

Money=100

//制造该坦克的开销是 100 货币单位

BuildTime=5

//制造该坦克所花费的时间是 5 时间单位

MaxNumber=200

//坦克的最大上限为 200 辆

.....

//省去其它的数据

当你把这些信息传递给电脑的相关程序以后，一辆十分真实的坦克便出现了——你的第一个对象诞生了。

通过上面的操作，你建立起了一些基本的概念了吗？如果你还有一些不明白的地方，建议你好好阅读本节，这些概念对于一个新手，尤其是刚刚进入可视化开发的人来说，是有一定的理解难度的。

接下来，我们将展示在 C++ Builder 中的一个最基本的对象——表单。

1.3 一切都是对象

C++ Builder 4 就是按照面向对象（OOP）的方式进行操作。在 C++ Builder 4 中，一切都是对象，如果你没有充分利用这一点，那么你会失去一半的乐趣，更不用说它的强大威力了。

编写面向对象的应用程序，意味着你能够重复使用你的工作成果，并且可以使得你的程序维护起来更为容易，程序扩充起来也更为方便，并且不会导致大量的错误。这是因为

它们都是十分容易分离开来的对象。

需要说明的是：对象的实现是基于类（Class）的。每当你新建一个对象的时候，它实际上已经继承了原始的类生成了一个新的类，并且定义了该类的一个实例——它就是你现在使用的对象。这意味着你可以在现有的对象类型（也就是类）的基础上创建新的对象。这个新的对象将继承其先辈们的所有的事件、属性和方法。而且你会发现，在对衍生的对象添加新的函数方面，你有着十足的灵活性。此外，该对象不必以同样的方式像其它的对象那样对函数的调用做出反应，你可以用普通的方式来设计你的执行特定功能的应用程序。

比如，我们假定“Animal”（动物）是一个祖先类，而“狼”、“羊”、“鱼”以及“人”都有一个“eat”（进食）的过程，但是，显而易见这些对象的行为的表现形式都是不同的。对于“eat”命令，狼会根据自己的情况来捕杀弱小的动物，而羊只会寻找鲜嫩的绿草，其它两个对象的情况也是如此。我们举出这个例子只是让你对对象的概念有所思考。

接下来，我们以一个标准表单为例，来说明什么是对象，在以后的章节里，你会看到更多的利用对象编程的内容。

示例：一个表单（Form）。

C++ Builder 已经为我们提供了许多可视化元素的对象，其中“表单”就是一个最常用的例子。所谓“表单”，就是“窗口”的另一种叫法。

需要说明的是：为了避免引起混淆，在以后的所有的示例和描述中，我们将全部使用“表单”这个术语称谓，请记住，它就是一个标准的 Windows 95/98 的“窗口”，也是你的应用程序的界面的重要组成部分。

启动 C++ Builder，你会发现 C++ Builder 已经创建了一个新的项目，名称为“Project1”，这个项目只包含一个缺省的单一的表单，它的名称为“Form1”。

现在，你可以看到，不费吹灰之力，我们已经拥有了一个具有 4 个边界的表单。它有一种背景颜色、一个标题栏、一个最小和最大化按钮以及一个标题“Form1”。

要想看一看 Form1 作为实际的应用程序究竟是一个什么样子，你可以通过 3 种方式来从 C++ Builder 4 的开发环境中运行“Project1”这个工程：

- ◇ 从“Run”菜单中选择“Run”命令。
- ◇ 使用快捷键【F9】。
- ◇ 从 C++ Builder 4 的加速按钮条上按下绿色的箭头按钮。

该应用程序运行后的界面如图 1-1 所示。

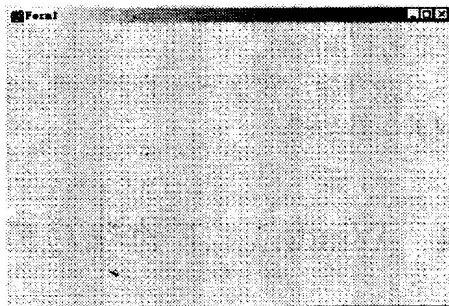


图 1-1 一个表单

观察一下，你会发现这个最简单的程序已经具有许多特性：

(1) 它是可以确定大小的。你可以通过使用鼠标拖动表单的边框来改变它的大小和位置；你可以通过鼠标单击标题栏上的按钮来最小化、最大化或者恢复表单的尺寸。

(2) 它在标题栏上有一个缺省的图标。你可以通过单击它来打开系统控制菜单，选择相应的菜单命令，来对该表单进行类似的改变大小和关闭的操作。

(3) 它有一个名字，显示在标题栏上。这说明该表单（窗口）已经在你的 Windows 系统中注册了，就像这样——“嘿！Windows 老弟，我是一个表单（窗口），我的名字叫做 Form1”。

此外，在 Win95 /98 系统中，表单的标题栏继承了系统默认的特性（除非你另外指定）。比如，标题栏的颜色、使用的字体样式和大小、最大化与最小化图标的大小，等等。

并且，你还可以尝试一下，给这个表单加上一些东西，比如一个 Edit（编辑框），如图 1-2 所示。

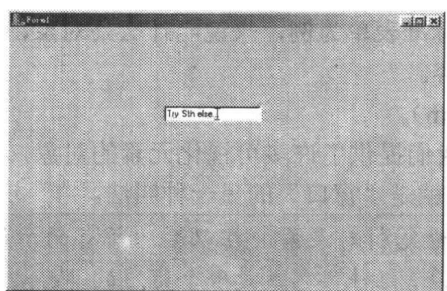


图 1-2 添加一个编辑框表单

尝试运行这个程序，你会发现除了上面提到的一些特性以外，当你把鼠标指针移动到编辑框“Edit1”中的时候，鼠标形状会发生相应的改变，这时你就可以在该编辑框中输入任何你所喜欢的东西了。当输入完毕以后，你的鼠标一旦离开（术语称之为“Exit”更为合适一些），鼠标指针的形状马上就会恢复初始的形状。

稍微对事件处理有一些印象的人都应该明白这个过程。我们指的是进入编辑器“Edit1”和退出该编辑器，鼠标指针的形状发生改变，实际上已经完成了一个事件的处理过程，只不过 C++ Builder 4 替你完成了这一切。

好了，你可以看到，以上所有的这一切都是 C++ Builder 免费提供的，我们不必花费任何的精力就可以获得这些基本的可视化元素，并且控制该表单和其它部件对象的操作行为。

这就是面向对象（OOP）的初步威力。

不过，请记住，这仅仅只是开始，在接下来的章节里，我们会展示更多的有关 C++ Builder 的神奇的、不可思议的奥妙功能。

1.4 Windows 下的程序开发

接下来，我们讨论一下 Windows 下的程序开发模式。这主要是针对那些长期在 DOS

操作系统下面开发软件，习惯于过程式开发风格的程序员而言的。

你可以看到，在 Windows 下面利用各种可视化工具来开发应用程序，你的绝大部分时间就是和各种各样的对象（这在 C++ Builder 中被称之为“部件”）的属性、事件、方法以及 Windows 的消息打交道。你需要利用这些东西来最终设置出你的应用程序的外观，并且编写处理事物的代码来完成该应用程序相应部分的功能。

所有这一切，对于那些习惯于 DOS 下编程的人们来说会有些无所适从，他们是完全的过程式思维方式，其一般的工作模式如下：

- ◇ 分析需求。
- ◇ 划分步骤。
- ◇ 编写每一个步骤（实际上也是一个模块）的代码。
- ◇ 测试模块代码功能。
- ◇ 完成该软件。

他们手工编写自己的每一行代码，调用每一个中断来实现屏幕上的每一个对象的外观和操作，把一切问题都划分成为许多细小的过程和步骤。所以，他们一旦进入了 C++ Builder 的面向对象（OOP）的世界，会有一种无所适从的感觉——似乎一切都由 C++ Builder 做了，他们还能做什么？

真的是这样吗？不，C++ Builder 替我们做的只不过是免除那些琐碎的与底层（比如中断）打交道的工作，使得程序员不再为细微的战术问题而重复劳动。我们这么说，是认为把大量的精力投入到诸如编制每一个相似的窗口和菜单上是毫无意义的，也是一种时间和精力上的重复和浪费行为。而且其代码是极其脆弱和不能够高度共享的，并不符合软件工程发展的趋势。

C++ Builder 4 给我们提供了良好的开发环境，省去了你大量的麻烦，能够轻而易举地实现代码的高度共享，使得你能够在软件的战略层面上，花更多的精力来重视软件的定义、代码的逻辑和划分、以及软件的其它相关方面，以求更好地完成该程序。

而且，在具体处理事务的阶段，你仍然可以借鉴过去的编程经验和风格，在函数和过程级别，DOS 和 Windows 的差别不是那么明显。唯一可能令人遗憾的是，Windows 出于安全的考虑，你的代码不能够直接操作一些硬件了，比如，直接写屏技术，而这样做的效率往往会高一些。

可以毫不夸张地说，在 C++ Builder 4 中，你只有想不到的，而没有做不到的。

接下来，我们将列举 C++ Builder 4 的一些优点。

1.5 为什么选择 C++ Builder 4

一口气介绍了这么多的基本概念，现在又回过头来讨论为什么要选择 C++ Builder 4 来作为我们的开发工具？C++ Builder 4 之所以获得我们的青睐，当然源于它自身的特性。它实在是非常棒！

（1）一种风格灵活的语言

C++ Builder 并不是一种新的语言，它的前身 C++，可以说是非常的普及。