



C++ Builder

程序设计

张建平 主 编
丁志芳 副主编
毋茂盛

兵器工业出版社

责任编辑:宋丽华

封面设计:马 娜

C++Builder

内容简介

Borland C++Builder是Borland公司为C++语言编写的快速应用程序开发环境。本书以C++Builder6.0版本为基础,系统介绍了C++Builder的编程环境、VCL构件的属性、事件、方法和C++Builder提供的开发工具的用法。内容由浅入深,实例丰富精彩。

本书分三大部分。第一部分介绍C++Builder的安装和程序结构,说明C++Builder的主要功能、构件和操作,是学习C++Builder程序设计的基础部分。第二部分是C++语言部分,主要介绍C++语言的语法、概念,编写控制台程序方法。第三部分详细介绍C++Builder在各种情况下的编程技巧。全书都给出了完整的代码。



ISBN 7-80172-686-3
定价:39.80元

C++Builder 程序设计

张建平 主 编
丁志芳 毋茂盛 副主编

兵器工业出版社

内容简介

Borland C++Builder 是 Borland 公司为 C++ 语言编写的快速应用程序开发环境。本书以 C++Builder 6.0 版本为基础,系统介绍了 C++Builder 的编程环境、VCL 构件的属性、事件、方法和 C++Builder 提供的开发工具的用法。内容由浅入深,实例丰富精彩。

本书分三大部分。第一部分介绍 C++Builder 的安装和程序结构,说明 C++Builder 的主要功能、构件和操作,是学习 C++Builder 程序设计的基础部分。第二部分是 C++ 语言部分,主要介绍 C++ 语言的语法、概念,编写控制台程序方法。第三部分详细介绍 C++Builder 在各种情况下的编程技巧。全书都给出了完整的代码。

图书在版编目(CIP)数据

C++Builder 程序设计/张建平主编. —北京:兵器工业出版社,2006.7

ISBN 7-80172-686-3

I. C... II. 张... III. C 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 066915 号

出版发行:兵器工业出版社
发行电话:010-68962596, 68962591
邮 编:100089
社 址:北京市海淀区车道沟 10 号
经 销:各地新华书店
印 刷:北京市梨园彩印厂
版 次:2006 年 7 月第 1 版第 1 次印刷

封面设计:马 娜
责任编辑:宋丽华
责任印制:赵春云
开 本:880×1230 1/16
印 张:15.5
字 数:400 千字
定 价:39.80 元

(版权所有 翻印必究 印装有误 负责调换)

前言

C++语言是一门高效实用的程序设计语言。它不仅适合于系统程序设计,也适合于应用程序设计,既可以进行过程化程序设计,也可以进行面向对象的程序设计。C++实现了类的封装、数据隐藏和继承等设计思想,使得其代码容易维护及高度可重用。传统的计算机语言教学已经逐渐转向 C++语言。

2002 年上半年 Inprise 公司(原 Borland 公司)推出了 Borland C++ Builder 的新版本 C++ Builder 6.0。它继承了以前版本的优点,实现了许多新的功能,C++ Builder 系列产品是当今软件开发工具中不可多得的优秀产品,初学者学习使用 C++ Builder 很容易上手。程序开发人员使用 C++ Builder 能够高效率完成程序设计任务。尽管 C++ Builder 不是将 C++变得容易学习。但是,它确实将 C++语言变成一种有趣的语言,使得学习者在设计程序的时候就可以看到程序运行的效果(所见即所得)。作为使用 C++ Builder 开发工具的学习者或软件工程人员,使用 C++ Builder 能够极大地提高软件系统的设计、开发速度和开发软件的层次水平。

本书的主要思想是将传统的计算机语言教学和学习程序设计的方法联系起来,强调在总体上把握程序设计的方法,从应用语言的角度系统地掌握计算机语言的语法和概念。而不拘泥于具体和繁杂的概念细节。作为一个完整的体系,在学习中使用 C++语言,在使用中提高 C++语言的编程技巧。在内容安排上,尽可能地培养读者从整体的角度理解程序设计的方法,而不是片面地强调具体的语法、语义规则。在内容安排上,首先介绍 C++ Builder 的安装和开发环境,对书中每一个程序都给出完整的代码,对具体语言中的细节不作太多的解释,这里不是说这些细节不重要,恰恰相反,在程序设计中,细微的差别可能给程序的执行时间和空间效率带来极大的影响,但是,作为一个学习程序设计人员或程序开发人员,更重要的是首先在总体上把握程序设计的方法和技巧,毕竟在学习程序设计中,对于一门计算机语言中的语法、语义、运算符、优先级、结合性等概念是相对容易掌握的,而从整体上把握程序设计的方法和技巧,是对程序设计人员的综合能力的考验。另外,从系统整体的角度学习程序设计,可以在更短的时间内做出一个比较像样的程序。对初学者来说,这种成就感和自豪感会成为学习的动力源泉。C++提供了强大的功能,这些功能是以大量的概念作为基础的,开始学习的时候就在一些繁杂的具体概念上纠缠,往往会使初学者失去信心。世界上每天都有大量的学习程序设计人员,很多人都是在学习由于信心、动力不足或者压力而放弃。作为一个长期从事计算机语言教学研究和程序开发者,每当看到这种情况,都会感到十分遗憾。因此,从一个新角度来考虑程序设计的学习方法,从初学者的成就感开始,培养读者的信心是本书的主要思想。

通过学习本书,可以得到如下的目标:

- (1)学会在 C++ Builder 下调试 C 程序、C++程序;

- (2)学会在 C++ Builder 的编程环境下编制简单的应用程序;
- (3)掌握面向对象的程序设计思想;
- (4)掌握当今流行的 C++ Builder 快速应用程序开发环境;
- (5)掌握包括帮助文件、DLL 和安装程序制作的完全程序开发过程;
- (6)掌握 C++ Builder 在多种应用需求中的编制程序的方法和技巧。

关于本书给出的程序代码,建议读者仔细领会编者的设计意图,逐个调试程序,必要的情况下可以在 C++ Builder 6.0 调试环境中,用单步执行的方法一步一步地执行程序,然后,按照自己的想法改变原代码,摹仿设计出自己的程序。

计算机语言是一门实践性很强的语言。建议读者多上机、多练习、多交流、多讨论、多思考,多参考例子程序,在实践中熟悉开发环境,掌握语法、语义等规则。Internet 上有大量的例子程序资源可以利用,这些例子程序很有参考价值。市面上有各种编程理论和资料也提供了大量的编程实例,研究实例程序能很快提高程序设计水平。编者认为:要学好计算机语言的最好的方法是在掌握一定语法、语义的基础上,从事一个项目的工作,在工作中使用语言,编程水平可以很快地提高;但是考虑到具体的学习环境,在学习从事项目工作有一定的条件限制,可以“无中生有”地创造一些项目。例如:结合身边的同学、同事、朋友等人员的需求编制一些简单适用的工具,如计算平均成绩、成绩排名次、平均收入、简单记事本、个人通信簿、选票统计等,赠送给他们,相信在实践中会有更大的收获。

编者相信,只要经过辛勤的劳动,加之良好的方法,一定能够得到一个更高的程序设计境界,极大地提高程序设计的能力和水平。无论从学习编程语言,还是进行应用程序的开发,选择 C++ Builder 都是正确的。

本书由张建平主编。其中第 3、4、5、6 章由丁志芳编写,第 10、11、12 章由毋茂盛编写,第 16 章由李学军、高艳平编写,第 17 章和第 18 章由聂世群、张兵利、裴亚辉编写,第 19 章和第 20 章由宋志飞、俞卫华、仝利红、石静编写。其余章节由张建平编写,全书由张建平统稿。

由于时间仓促,加之编者水平所限,书中难免有不足之处,欢迎广大读者批评指正,以便改进。联系邮件 E-mail 为:zhjianping@tom.com

编者

2006 年 6 月

目 录

第1章 C++Builder 入门.....	(001)	4.2.2 while 循环.....	(044)
1.1 C++Builder 简介.....	(001)	4.2.3 do while 循环.....	(045)
1.2 认识 C++Builder 6.0.....	(002)	4.2.4 for 循环.....	(045)
1.3 属性、事件和方法.....	(004)	4.3 break 语句和 continue 语句.....	(047)
1.4 C++Builder 的控制台程序.....	(009)	4.4 switch 语句.....	(049)
习题.....	(012)	4.5 变量的作用范围.....	(051)
第2章 C++Builder 的程序结构.....	(013)	4.5.1 局部变量.....	(051)
2.1 一个简单的 C++Builder 程序.....	(013)	4.5.2 全局变量.....	(051)
2.2 C++Builder6.0 的构件板.....	(016)	4.5.3 static 变量.....	(051)
2.3 简单计算器.....	(021)	4.5.4 auto 变量.....	(051)
2.4 AnsiString 方法.....	(022)	4.5.5 register 变量.....	(051)
2.5 彩票自动选号.....	(024)	4.5.6 extern 变量.....	(051)
2.6 字母的大小写转换.....	(025)	4.6 结构.....	(052)
习题.....	(025)	习题.....	(054)
第3章 C++基础知识.....	(027)	第5章 高级 C++.....	(056)
3.1 数据类型.....	(027)	5.1 指针.....	(056)
3.2 关键字和变量.....	(029)	5.2 指针变量作为函数参数.....	(064)
3.2.1 关键字.....	(029)	5.3 引用.....	(066)
3.2.2 变量的命名.....	(029)	习题.....	(068)
3.3 C++的操作符.....	(030)	第6章 C++类与面向对象编程.....	(070)
3.4 C++函数.....	(031)	6.1 类概念的建立.....	(070)
3.5 内联函数和递归函数.....	(033)	6.2 类的构成.....	(074)
3.6 函数的重载和默认参数的函数.....	(036)	6.3 类的继承(inheritance).....	(075)
3.7 数组和字符数组.....	(036)	习题.....	(079)
3.7.1 一维数组.....	(036)	第7章 活动字幕.....	(080)
3.7.2 二维数组.....	(038)	7.1 简单活动字幕.....	(080)
3.7.3 字符数组.....	(039)	7.2 带平板的活动字幕.....	(081)
习题.....	(040)	7.3 文本构件组应用.....	(082)
第4章 C++语句构成.....	(042)	7.4 实现窗体上文本或数据的复制.....	(084)
4.1 if 语句.....	(042)	7.5 三角函数发生器.....	(084)
4.2 循环语句.....	(044)	7.6 窗体上放置多个构件.....	(086)
4.2.1 goto 循环.....	(044)	7.7 一次动态生成多个构件.....	(088)

7.8 在画布上显示文字和数据.....	(089)	11.3.2 编写帮助文件(RTF 格式)···	(140)
7.9 模仿 LED 显示.....	(090)	11.3.3 使用帮助文件编译工具生成 HLP 文件和 CN 文件.....	(141)
7.10 查错器调试程序简单介绍.....	(092)	11.3.4 关于帮助文件制作的几点 说明.....	(142)
习题.....	(093)	习题.....	(143)
第 8 章 菜单设计	(094)	第 12 章 获取系统信息	(144)
8.1 手工设计菜单.....	(094)	12.1 获取 Windows 版本信息.....	(144)
8.2 利用菜单设计器设计菜单.....	(095)	12.2 获取 CPU 信息.....	(145)
8.3 工具条和状态条.....	(097)	12.3 获取内存状态.....	(146)
8.4 动态菜单技术.....	(099)	12.4 设置系统界面颜色.....	(147)
8.5 Sender 参数的用法.....	(101)	12.5 获取 Windows 路径及系统路径.....	(148)
8.6 实现菜单的功能.....	(102)	12.6 获取用户注册信息.....	(149)
习题.....	(104)	12.7 在窗体上关闭 Windows.....	(150)
第 9 章 MDI 程序设计	(105)	12.8 隐藏/显示桌面图标.....	(151)
9.1 MDI 程序.....	(105)	12.9 隐藏/显示任务栏.....	(151)
9.2 生成 MDI 程序.....	(105)	习题.....	(152)
9.3 利用向导建立 MDI 应用程序.....	(106)	第 13 章 制作安装程序	(153)
9.4 多页面应用程序.....	(107)	13.1 准备制作安装程序.....	(153)
9.5 对话框.....	(108)	13.2 制作软件封面.....	(154)
9.6 C++Builder 中的项目.....	(109)	13.3 制作安装程序.....	(157)
9.7 Project 菜单.....	(111)	13.3.1 建立制作安装程序项目.....	(157)
习题.....	(112)	13.3.2 制作安装程序.....	(158)
第 10 章 键盘和鼠标的响应	(114)	习题.....	(165)
10.1 键盘的响应.....	(114)	第 14 章 设计多线程程序	(166)
10.2 鼠标按键响应.....	(116)	14.1 线程的概念.....	(166)
10.3 限定鼠标的活动区域.....	(117)	14.2 编写一个新的线程.....	(167)
10.4 鼠标拖动事件.....	(118)	14.2.1 建立一个新线程.....	(167)
10.4.1 图像的拖动.....	(118)	14.2.2 编写新线程执行代码.....	(168)
10.4.2 构件属性的拖动.....	(119)	14.3 使用线程.....	(169)
10.5 打字练习程序.....	(119)	习题.....	(173)
10.6 获取窗体上控件及位置.....	(121)	第 15 章 设计自己的构件	(174)
10.7 在状态栏上显示鼠标的位置.....	(122)	15.1 在 C++Builder 中设计构件.....	(174)
10.8 功能键的检测和控制.....	(123)	15.1.1 确定构件的功能.....	(174)
习题.....	(125)	15.1.2 确定构件的祖先类.....	(174)
第 11 章 制作帮助文件	(127)	15.1.3 编写构件的原文件.....	(175)
1.1 C++Builder 的应用程序向导.....	(127)	15.2 编写构件的属性和方法.....	(176)
11.1.1 C++ Builder 的向导.....	(127)	15.2.1 属性数据成员.....	(176)
11.1.2 利用 Application Wizard 快速 生成应用程序.....	(127)	15.2.2 属性 write 方法和 read 方法···	(177)
11.2 完成我的写字板程序设计.....	(132)	15.2.3 属性直接访问.....	(177)
11.2.1 完善写字板程序的界面设计···	(132)	15.2.4 编写构件的方法.....	(178)
11.2.2 为写字板程序编写代码.....	(133)	15.3 编写构件的事件.....	(181)
11.3 编写帮助文件.....	(139)	15.3.1 理解事件.....	(181)
11.3.1 帮助文件的步骤.....	(139)		

15.3 编写构件的事件.....	(181)	18.3 窗体上任意正三角形图形的绘制.....	(215)
15.3.1 理解事件.....	(181)	18.4 屏幕坐标表示画正三角形.....	(216)
15.3.2 编写事件代码.....	(181)	18.5 图形的动画技巧.....	(217)
15.4 完整的构件代码.....	(183)	18.6 图像的动画技巧.....	(218)
15.5 测试和安装构件.....	(186)	18.7 播放声音文件.....	(219)
15.5.1 构件的测试.....	(186)	习题.....	(220)
15.5.2 将新构件安装到构件板上.....	(187)	第 19 章 使用 C++Builder 工具.....	(221)
15.5.3 给构件加上的图像.....	(188)	19.1 C++Builder 查错器.....	(221)
习题.....	(188)	19.1.1 设置断点.....	(222)
第 16 章 设计自己的 DLL.....	(190)	19.1.2 变量观测器.....	(223)
16.1 理解 DLL.....	(190)	19.1.3 构件和类的查错.....	(224)
16.1.1 什么是 DLL?	(190)	19.1.4 高级查错工具.....	(225)
16.1.2 为什么要用 DLL.....	(190)	19.1.5 查错工具小结.....	(227)
16.1.3 DLL 中的函数的调用.....	(190)	19.2 其他 C++Builder 工具.....	(228)
16.1.4 内部函数、导入函数和导出函数.....	(191)	19.2.1 图形编辑器.....	(228)
16.2 设计自己的 DLL.....	(191)	19.2.2 ws32.exe 应用程序工具.....	(229)
16.3 动态装入 DLL.....	(195)	19.2.3 命令行工具.....	(229)
16.4 设计包含构件的 DLL.....	(197)	19.3 VC 转换工具.....	(230)
16.5 定制不同语言版本的应用程序.....	(199)	19.4 对象仓库配置工具.....	(231)
习题.....	(201)	19.5 选项工具.....	(231)
第 17 章 数据库程序设计.....	(202)	19.5.1 环境选项.....	(231)
17.1 数据库的基础知识.....	(202)	19.5.2 编辑选项.....	(232)
17.1.1 数据.....	(202)	19.5.3 转换工具选项.....	(232)
17.1.2 数据库.....	(202)	19.6 项目选项工具对话框.....	(233)
17.1.3 数据库管理系统(DBMS).....	(202)	19.6.1 Forms 页面.....	(233)
17.1.4 C++Builder 对数据库的支持.....	(203)	19.6.2 Application 页面.....	(233)
17.2 简单数据库程序设计.....	(203)	19.6.3 Compiler 页面.....	(233)
17.3 数据库表的设计.....	(204)	19.6.4 Advanced Compiler 页面.....	(234)
17.4 数据表结构对话框.....	(205)	19.6.5 Directories/Conditionals 页面.....	(234)
17.5 从数据库读出数据.....	(205)	19.6.6 其他页面选项设置.....	(234)
17.6 将数据写入数据库.....	(206)	习题.....	(234)
17.7 数据库报表.....	(207)	第 20 章 C++Builder6.0 与 office2000.....	(235)
17.8 多媒体数据库程序设计.....	(208)	20.1 最简单的方法.....	(235)
习题.....	(210)	20.2 组合应用 Office 功能.....	(236)
第 18 章 图形、图像与多媒体.....	(211)	20.3 Word 的更灵活方法.....	(237)
18.1 最简单的图形.....	(211)	20.4 Winexec 函数调用外部程序举例.....	(238)
18.2 绘图의初步体验.....	(212)	习题.....	(240)

第1章 C++Builder 入门

1.1 C++Builder 简介

C++Builder 是 Borland 公司推出的快速应用程序 (Rapid Application Development 简称为 RAD) 开发工具。它提供易学易用的图形化操作界面和功能强大的可视化构件库 (Visual Component Library 简称为 VCL), 可以使程序设计人员快速建立各种应用程序, 同 C++ 一样, 可以方便地编写系统软件和各种应用软件, 可以生成 Win32 控制台程序或 Win32 GUI (图形用户界面程序)、是当今市场上真正利用拖放技术, 快速开发应用软件的集成开发环境 (Integrated Development Environment 简称为 IDE) 之一。在学习 C++Builder 编程过程中, 能更快理解图形化应用程序设计的本质; 在应用程序设计中, 用 C++Builder 开发的程序具有 C++ 语言设计程序的所有优点, 即高效率 and 强功能。特别是对当今软件市场越来越大、应用环境越来越复杂的情况下, C++Builder 的优点更能够充分体现。

C++Builder 和 Delphi 是一对孪生兄弟, Borland 在 1985 年推出革命性的 Delphi1.0, 实现了可视化程序设计, 使得在程序的设计时就可以看到程序运行时的效果, 在软件世界引起轰动。但 Delphi1.0 仅是成功的开端, 接着 Borland 又推出 Delphi2.0, Delphi2.0 是一个真正的高效率程序开发环境, 它极大地提高了 Borland 公司的声誉和在软件业的地位。Delphi 的核心代码 (VCL 代码) 和程序代码都是用 Pascal 编写的, 具有编译速度快、程序开发速度快、运行效率高和可视化所见即所得等特点。在 Delphi2.0 取得成功后 Borland 公司 1987 年推出 C++Builder1.0, 尽管 C++Builder1.0 的 VCL 代码仍然是 Pascal 语言。但是, 对程序设计人员来说, 它使用的编程代码是 C++, 这正是它的优点所在。即既具有 Delphi 的快速开发特性, 又具有 C++ 一流语言的高效和强功能。Delphi 和 C++Builder 推出的时间顺序是这样的 Delphi 1.0、Delphi 2.0、C++Builder 1.0、C++Builder 2.0、Delphi 3.0、C++Builder3.0、Delphi 4.0、C++Builder 4.0、Delphi5.0、C++Builder5.0、Delphi6.0、C++Builder6.0、Delphi7.0…。可以预见这个过程还要继续, 而每一个新的版本出现, 都扩展了 Delphi 和 C++Builder 功能。

如果你是一个学习程序设计的新手, 编者要衷心地祝贺你, 可以使用 C++Builder 学习 C 语言、C++ 语言 (计算机教学语言几乎都是 C 语言、C++ 语言), 也可以直接学习 C++Builder 程序开发环境; 如果你是一个资深的 C++ 程序设计人员, 使用 C++Builder 将会真正体会到高效、强功能的集成开发工具; 如果你熟悉 Delphi 会发现它们的开发界面是如此相似 (有微小的区别), 而几乎不要重新学习新的开发环境, 而熟悉和掌握程序的开发环境是程序设计人员一项重要内容。并且, 由于 Delphi 和 C++Builder 共用相同的核心代码 (VCL), 原来程序的 Delphi 代码可以很方便地转换为 C++Builder 代码。

C++Builder 的新版本是 2002 年推出的 C++Builder6.0, 在 Internet 网上许多程序员将它简称为 BCB6。C++Builder6.0 针对不同的用户开发不同的系列版本, 分为 Enterprise (企业版)、Professional (专业版) 和 Personal (个人版) 三个版本, 本书以介绍企业版为主, 三个版本的操作界面基本相同, 主要差别如下:

企业版: 适用于软件开发商、企业开发人员、系统集成公司等大型应用程序开发和数据库应用程序开发。

专业版: 适用于大多数商业软件开发和数据库应用程序的开发。

个人版: 适用于为初学者提供可视化的开发工具, 完全的图形界面, 轻松学习程序设计, 快速掌握 C++Builder 编程技术。

对于学习程序设计人员来说, C++Builder 提供的是完全可视化编程环境; 对于程序员来说, C++Builder 提供的是一个完整的系统解决方案。可以利用 C++Builder 生成新构件 (VCL)、生成动态链接库 (DLL)、生成库文件 (LIB)、生成资源文件 (RES)、生成任何形式的可执行文件 (EXE、面板程序、屏幕保护程序等)、支持团

队开发、支持多种本地数据库、远程多层数据库的开发、分布式数据库程序等各种大型数据库应用开发、方便的网络编程环境能快速开发 WEB 应用程序、完善的系统支持环境、性能良好的向导功能支持快速开发应用程序、智能化的在线系统帮助功能,能够利用 C++Builder 自带的安装程序制作工具,设计制作具有专业水准的安装程序。另外,大量的网络资源和庞大的 C++Builder 学习和使用人员,提供了良好的学习、使用和交流环境。在软件业有这样的说法:聪明的程序员用 Delphi,真正的程序员用 C++,而真正聪明的程序员用 C++ Builder。

1.2 认识 C++Builder 6.0

C++Builder 6.0 有两张安装光盘,将安装盘 DISK1 放入光驱中,安装程序会自动引导用户安装到硬盘中。安装过程中需要用户输入一系列内容,可以按照提示要求输入,当完成安装后,在开始菜单中将出现如图 1.1 所示的内容。

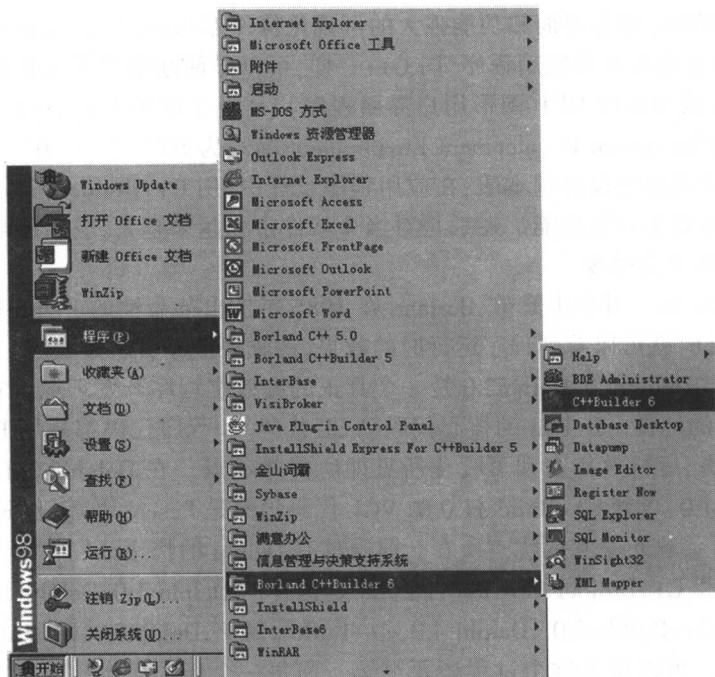


图 1.1 C++Builder 6.0 的[开始]菜单

在开始菜单中将鼠标移动到 C++Builder 6.0 上,就可以打开 C++Builder 6.0 应用程序开发环境,其界面如图 1.2 所示为打开 C++Builder 6.0 后的程序界面,这个界面就是将要工作的主体。

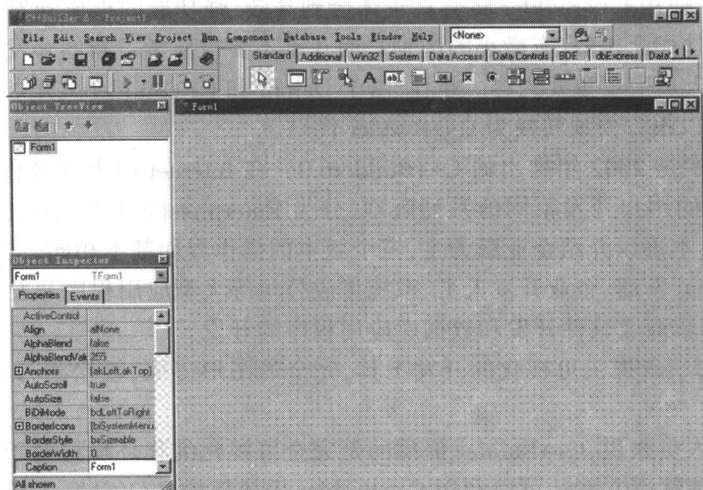


图 1.2 C++Builder 6.0 应用程序开发界面

C++Builder 6.0 开发界面中,最上边的是主菜单,主菜单列出了系统的全部功能,学习和使用本工具的一个重要任务就是尽快地熟悉系统的菜单和开发界面;在其左下方的是部分主菜单的快捷按钮,这些按钮是常用菜单的快捷操作,可以按照用户的习惯自己配置(定制)和随意排列,是十分有用的快捷工具栏,用户可以将常用的菜单命令放到快捷按钮中,这样在操作 C++Builder 时不必在菜单上查找;工具栏的右方的是 VCL 构件板,构件板提供了在应用系统编程中使用的各种构件,是 C++Builder 系统真正快速开发工具的灵魂所在,构件板也是可以由用户配置(定制),用户可以随意排列构件在构件板上的顺序,甚至可以将自己设计或第三方设计的构件(在 Inetnet 上有大量的第三方构件资源)加入到的构件板中。一般情况下,用户可以将常用的构件放置在开发界面的显著位置。将命令按钮加入到工具栏的方法是:在工具栏上右单击鼠标,出现工具栏弹出菜单,如图 1.3 所示。

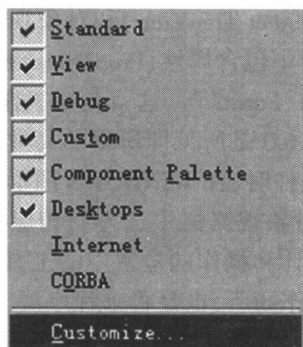


图 1.3 工具栏弹出式菜单

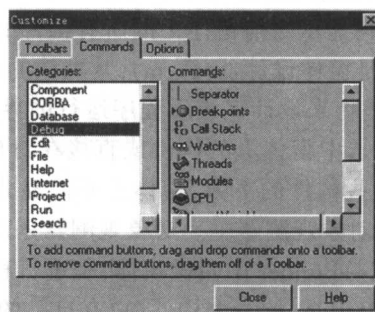


图 1.4 工具栏配置对话框

选择 Customize... 菜单项目,出现如图 1.4 所示的工具栏配置对话框,用鼠标选中某个 Commands 中的命令,直接拖到工具栏中就可以增加一个按钮。删除按钮的方法是:首先打开图 1.3 工具栏弹出菜单,用鼠标左键选中要删除的按钮,拖出工具栏即可。

因为系统提供了大量的构件,为了便于管理这些构件,构件分类地放置在构件板的不同页面上,如 Standard、Additional... 配置构件板上的构件的方法是:在主菜单【Component】->【Configure Palette...】出现如图 1.5 所示的构件板配置对话框,这个对话框分两栏,左栏可以选定构件板的页面,右栏是选定构件板页面上的构件,下边有两排按钮,上排五个按钮为“Add”、“Delete”、“Rename”、“Move Up”和“Move down”,这些按钮的作用像它们的名字一样,分别为“增加”、“删除”、“重命名”、“上移”和“下移”,对应的构件板页面或页面上的构件就会在相应的位置增加、删除、和移动一个位置,和配置快捷工具按钮一样,这里对构件的删除,并不是真正意义上的删除,只是在构件板上显示处删除,需要的时候还可以随时增减按钮,或将需要的构件重新加入构件板中。最后单击下边的“OK”按钮确定操作,也可以选择“Cancel”取消本次操作。构件板是应用最多的系统工具,在开发程序时,要将常用的构件放在显著的位置以方便操作。在主菜单【Component】中,还有几个分项菜单,它们的功能都是对构件板或构件进行操作。必要的时候可以查看帮助文件,只要熟悉一个功能以后,就能够触类旁通。另外,也可以在构件板上右单击,出现一个弹出式菜单,包含五个菜单项(Tab、Show Hints、Hide、Help、Properties),选择 Properties,也会出现如图 1.5 所示的构件板配置对话框。

在构件板配置对话框选择相应的 Pages 栏或 Components 栏,可以移动构件页面或构件在界面上的显示顺序、增加或删除构件。对于初学者来说,系统的默认配置就可以满足要求,经过一段时间熟悉

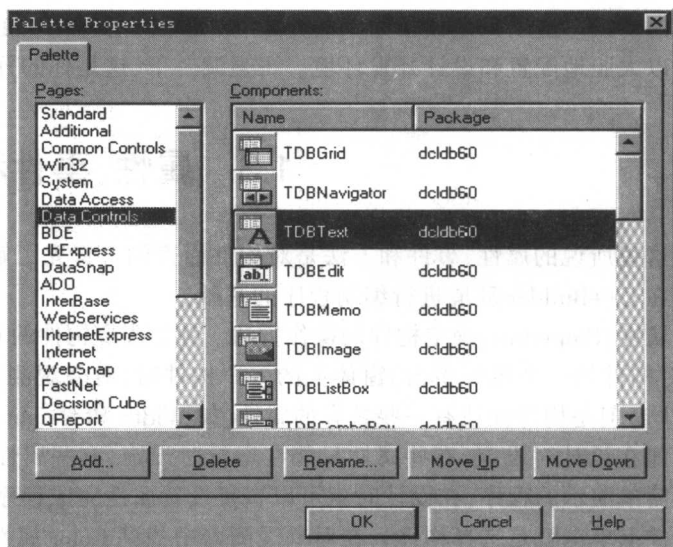


图 1.5 构件板配置对话框

以后,可以将常用的命令按钮加入到工具栏中,将常用的构件和构件页面放置到显著位置。C++Builder 6.0 提供的配置选项还有很多,例如,代码配置、TOOLS 配置、编译环境配置、查错环境配置等。在学习一段时间以后,就可以将系统设置为适合你自己的开发环境。和前面介绍的一样,配置开发环境中删除按钮和删除构件不是真正的删除,只是在显示界面上的删除,当需要的时候还可以将它们放置到界面上,所以要大胆地配置开发环境,因为它是你的程序开发工具,只有驾驭开发工具,才能方便地使用工具。

构件板中的构件分两种:一种是可视化构件,即设计时和将来程序运行时是完全相同的,这就是所见即所得;另一种是非可视化构件,即在程序设计时可以看见,将来程序运行时是不可见的。每个构件都有一系列属性、方法和事件,对这些构件的操作就是进行程序开发。C++Builder 之所以能够快速开发程序,原因之一就是利用系统为构件提供的属性、方法和事件,在后台为程序员完成大部分工作的细节,程序员只要有效利用构件就可以完成既定的大部分任务。再往下是对象观测树(Object TreeView)和对象观测器(Object Inspector),对象观测树显示当前窗体上的所有构件的名称。对象观测器中包含属性(Properties)和事件(Events)两个页面。其右边是窗体设计器(Form),其默认的名字分别是 Form 1、Form2 等,这就是要工作的窗体。对可视化构件来说,这是将来设计完成后程序运行的界面,即 C++Builder 6.0 是所见即所得的高效率的图形应用程序设计工具。当选中窗体上的构件或直接在对象观测树中选择构件时(在 WINDOWS 图形编程中常将一些构件放置在另一些构件的下方,窗体上不能显式看到,这时可在对象观测树中选择),相应构件的属性和事件就自动出现在对象观测器中。对象观测器是对构件进行程序设计和操作的最常用的工具之一。

Form 在 C++Builder 中叫做窗体,是进行程序设计的主要工作场所,也是将来程序展现给用户的界面,它本身也是 C++Builder 的一个构件,也有自己的属性、方法和事件。用作其他构件的容器,放置其他构件。Form 的 Caption 属性是显示程序运行时的标题,在实用的程序中,这个标题是很重要的,它可以直接告诉用户当前操作的是程序的那个界面,Caption 属性可以是任何可显示的字符,Form 的 Name 属性是窗体的名字,Name 属性只能设置为 C++ 的有效标识符。设置 Form 的 Caption 属性和 Name 属性尽管都很简单,但也很重要。大多数的构件都有自己的 Name 属性,更改构件的 Name 属性一般要在属性栏中改,不提倡在程序运行时改构件的名称,关于这点将来还要讲到。

用鼠标点击 Form1 下方的标题,当标题栏颜色加深时,用鼠标向下拉,在 Form1 的下方出现代码编辑器,其默认的名字分别是 Unit1.cpp、Unit2.cpp 等,从名字后缀已经感觉到了 C++ 的气息,这就是代码编辑器。程序设计的代码,程序要完成的任务,都将在这里体现。这是一个多页面设置,在代码编辑器下方栏中有 Unit1.h、Unit1.cpp 和 Diagram 页面,这是相应的头文件和 C++ 原文件。能够通过鼠标的单击,在头文件和代码文件之间切换。应用程序的函数、变量、类等的声明都放置在这个头文件中,便于统一管理。如果你熟悉 C++ 面向对象的编程,可以想到头文件在本质上是一个类的声明。Diagram 页面提供了设置可视化构件和非可视化构件在对象观测树中的关系。为了显示这些关系,需要用鼠标点击对象观测数中的构件,拖拽到 Diagram 页面,系统会给出应用程序中个构件之间的关系。在同时设计多个界面时,能够方便地实现多界面的原代码编辑时切换。这里只是做很简单的介绍,要完全熟悉 C++Builder 6.0 的开发界面需要一定的时间和耐心,你尽可以大胆地配置和设计开发环境。再强调一下,这是你的开发环境,你是它的主人,要驾驭开发环境。

1.3 属性、事件和方法

这里所说的属性、事件和方法是对构件而言的。首先了解对象观测器中的属性和事件,然后介绍方法,这是在 C++Builder 环境进行编制程序的基础。

属性(Properties)确定构件的操作方法。例如改变构件的 Color 属性可以改变这个构件的背景颜色,当选中(对构件的一个鼠标单击)窗体上的一个构件时,对象观测器就显示这个构件的属性,不同的构件属性也不相同。但是构件往往有一些共有的属性(如 Width 和 Height 属性)。属性分设计属性和运行属性,即有些属性只能在设计时设定,有些属性只能在运行时设定,有些属性既可以在设计时设定,也可以在运行时改变。对于简单演示的程序,在设计时就可以直接设置属性完成程序的功能。例如,选择窗体(窗体也是构件,只要在构件上单击就是选择构件),在对象观测器中找到 Color 属性,点击 Color 属性右边的矩形,并作一些选择,看看窗体的颜色变化。实际应用程序的开发,需要编制大量代码改变构件的设计属性。例如 Button 构件的属性如下:其中设计属性就是在属性栏上显示的属性值,用黑体字表示出来,其余的为运行属性。

Action、Align、AlignDisabled、Anchors、BiDiMode、BoundsRect、Brush、Cancel、Caption、ClientHeight、ClientOrigin、ClientRect、ClientWidth、ComObject、ComponentCount、ComponentIndex、Components、ComponentState、ComponentStyle、Constraints、ControlCount、Controls、ControlState、ControlStyle、Cursor、Default、DesignInfo、DockClientCount、DockClients、DockOrientation、DoubleBuffered、DragCursor、DragKind、DragMode、Enabled、Floating、FloatingDockSiteClass、Font、Handle、Height、HelpContext、HelpKeyword、HelpType、Hint、HostDockSite、Left、LRDockWidth、ModalResult、Name、Owner、Parent、ParentBiDiMode、ParentFont、ParentShowHint、ParentWindow、PopupMenu、ShowHint、Showing、TabOrder、TabStop、Tag、TBDockHeight、Top、UndockHeight、UndockWidth、VCLComObject、Visible、VisibleDockClientCount、Width、WindowProc。

对于构件的简单操作,只要相应地设置设计属性的值就可以完成任务,设计属性值就是直接在对象观测器中设置属性的值,运行属性需要编制代码,完成属性值的设置。关于属性有下列几个内容(见表 1.1)。

- (1)属性显示为类的数据成员;
- (2)属性通常有确省值;
- (3)属性可以写入(属性赋值)、也可以读出;
- (4)属性可以分为只读、只写或运行属性;
- (5)运行属性不出现在对象观测器中,只能由程序代码改变;
- (6)属性可以包含简单数据类型、数组、集合(属性值前有+号)、枚举(属性值前有+号)。

表 1.1 构件常用属性

属性	功能
Left、Top、Height、Width	构件在窗体上的位置
Align Alignment	构件在父窗体上的对齐方式
Color	设置构件的背景颜色(文字颜色用 Font 属性设置)
Enabled	使能或失能构件,即构件能否起作用
Hint	设置构件的提示文本
ParentColor、ParentCtl3D、ParentFont、ParentShowHint	这些属性设置为 true 时构件继承父构件的 Color、Ctl3D、Font、ShowHint
Tag	存放构件的类指针、索引或其他数字
BorderStyle	构件和背景关系
Caption	构件显示名字(可以为任何可显示字符)
Name	构件在代码中的名字(只能为 C++标识符)
Ctl3D	构件是否要三维边框
HelpContext	与构件相联系帮助文件的索引号
PopupMenu	指定用户右单击按钮时显示的弹出菜单
TabOrder	设置构件的跳表顺序,即按 Tab 键焦点移动次序
Visible	构件是否显示

事件(Events)是用来实现构件与用户或构件与 Windows 交互作用。例如单击构件时会产生单击事件,可以编写响应单击事件的代码,在发生事件时,用代码完成相应的任务。在事件页面中的事件处理器上,双击后鼠标自动定位到事件处理代码的编辑器上,有关的事件声明也已经加入到相应的头文件中,这里的事件相当于 C++中类的成员函数(后边将要介绍)。例如 Button 构件的事件如下:OnActivate、OnCanResize、OnClick、OnClose、OnCloseQuery、OnConstrainedResize、OnContextPopup、OnCreate、OnDblClick、OnDeactivate、OnDestroy、OnDockDrop、OnDockOver、OnDragDrop、OnDragOver、OnEndDock、OnGetSiteInfo、OnHelp、OnHide、OnKeyDown、OnKeyPress、OnKeyUp、OnMouseDown、OnMouseMove、OnMouseUp、OnMouseWheel、OnMouseWheelDown、OnMouseWheelUp、OnPaint、OnResize、OnShortCut、OnShow、OnStartDock、OnUnDock。这是事件的默认名称。

毫无疑问,事件处理器函数的代码需要程序员编写,这是完全的 C++编程工作,也是 C++Builder 程序设计、调试工作的主要场所。关于构件的事件有下列几个内容(见表 1.2)。

- (1)可以根据程序的需要响应构件的任何事件;
- (2)不必要响应系统定义中的所有事件;
- (3)事件的响应是由事件处理函数(用户编写代码)处理的;
- (4)几个构件可以响应同一个事件(如菜单、快捷键和快捷按钮);
- (5)事件处理器的名字可以取默认值也可以改变事件处理器的名字(只能在对象观测器的事件页修改,原因和构件的 Name 属性相同);
- (6)事件处理器的 Sender 参数可以区分事件是由哪个构件产生的事件;
- (7)双击事件处理器可以打开相应的代码编辑器,并定位到输入的代码开始处;
- (8)事件处理器中已经包含了该事件所要求的函数参数;
- (9)事件处理器中可以调用用户编写的函数。

表 1.2 构件的主要事件

事件	功能
OnChang	构件以某种方式改变时发生此事件
OnCilck	在鼠标单击构件时发生此事件
OnDdlClick	在鼠标双击构件时发生此事件
OnEnter	焦点在构件上,按 Enter 键时发生此事件
OnExit	构件失去焦点时发生此事件
OnKeyDown	用户按下任一键时发生此事件
OnKeyUp	放开任一键时发生此事件
OnKeyPress	用户按下一键时发生此事件
OnMouseDown	鼠标放在构件上,按下鼠标键时发生此事件
OnMouseMove	鼠标移动到构件上时发生此事件
OnMouseUp	放开鼠标键时发生此事件
OnPain	需要重画构件时发生此事件

这里要说明的是:构件的事件处理器是相互关联的。例如,在鼠标单击构件产生 OnCilck 事件处理器,实际上系统还要产生 OnMouseMove、OnMouseDown 和 OnMouseUp,即鼠标在单击构件时要产生鼠标移动、鼠标按下和鼠标放开事件,它们产生的顺序是: OnMouseMove、OnMouseDown、OnCilck 和 OnMouseUp。同样。双击构件时产生事件处理器的顺序是 OnMouseMove、OnMouseDown、OnCilck、OnDdlClick 和 OnMouseUp, 单行编辑构件(Edit 构件)的键盘输入时会产生 OnKeyDown、OnKeyPress、OnChang 和 OnKeyUp。作为初级编程人员先不要对这些细节作太多考虑,只要了解有这种情况存在,将来在实践中慢慢体会。

方法(Method)是 C++Builder 提供的库函数。系统提供了大量的方法,对 VCL 构件来讲,它是让构件完成某个动作的函数。例如所有图形构件都用 Show()方法显示构件,相当于在属性中设置 Visible 属性为 true;用 Hide()方法隐藏构件,相当于在属性中设置 Visible 属性为 false,调用这些方法和在 C++中的类成员函数完全相同,不同的构件有不同的方法,对于程序员来说,掌握构件的方法是快速编写程序的捷径。大量的构件,每一个构件又有大量的方法,对程序设计人员来说使用系统提供的方法是简单和方便的,正像前面所说的那样,系统提供构件的方法很多。因此了解系统为程序员提供了那些方法,是设计人员的任务之一。获得系统提供的方法是利用 C++Builder 提供的在线帮助工具。一般地说,系统提供方法时,程序员不要再编写代码(当然也可以编写代码),直接利用就可以,因为系统提供的方法往往比单个的程序员编写的代码质量要高。例

如 Button 构件的方法如下(见表 1.3):

AfterConstruction、ArrangeIcons、Assign、BeforeDestruction、BeginDrag、BringToFront、Broadcast、CanFocus、Cascade、ClassInfo、ClassName、ClassNameIs、ClassParent、ClassType、CleanupInstance、ClientToParent、ClientToScreen、Close、CloseQuery、ContainsControl、ControlAtPos、CreateParentedControl、DefaultHandler、DefocusControl、DestroyComponents、Destroying、DisableAlign、DisableAutoRange、Dispatch、Dock、DockDrop、DragDrop、Dragging、DrawTextBiDiModeFlags、DrawTextBiDiModeFlagsReadOnly、EnableAlign、EnableAutoRange、EndDrag、ExecuteAction、FieldAddress、FindChildControl、FindComponent、FlipChildren、FocusControl、Focused、Free、FreeInstance、FreeNotification、FreeOnRelease、GetControlsAlignment、GetFormImage、GetInterface、GetInterfaceEntry、GetInterfaceTable、GetNamePath、GetParentComponent、GetTabOrderList、GetTextBuf、GetTextLen、HandleAllocated、HandleNeeded、HasParent、Hide、InheritsFrom、InitiateAction、InitInstance、InsertComponent、InsertControl、InstanceSize、Invalidate、IsImplementorOf、IsRightToLeft、IsShortCut、MakeFullyVisible、ManualDock、ManualFloat、MethodAddress、MethodName、MouseWheelHandler、NewInstance、Next、PaintTo、ParentToClient、Perform、Previous、Print、Realign、ReferenceInterface、Refresh、Release、RemoveComponent、RemoveControl、RemoveFreeNotification、Repaint、ReplaceDockedControl、SafeCallException、ScaleBy、ScreenToClient、ScrollBy、ScrollInView、SendCancelMode、SendToBack、SetBounds、SetFocus、SetFocusedControl、SetSubComponent、SetTextBuf、Show、ShowModal、TForm、Tile、Update、UpdateAction、UpdateControlState、UseRightToLeftAlignment、UseRightToLeftReading、UseRightToLeftScrollBar、WantChildKey

表 1.3 构件的主要方法

方法	功能
Show	显示构件
Hide	隐藏构件
Broadcast	向所有窗口构件发送消息
ClientToScreen	将窗口坐标变成屏幕坐标
ContainsControl	如果构件是窗体或构件的子构件,返回 true
HandleAllocated	检查句柄 Handle 是否存在
Invalidate	请求重画构件
Perform	不通过 windows 系统直接将消息发给构件
Refresh	立即重画构件,并在重画构件前擦去构件
Repaint	立即重画构件,并在重画构件前不擦去构件
SetBounds	允许一次性设置 Left、Top、Height、Width
SetFocus	设置构件焦点并成为激活构件
Update	强制重画构件

前面曾经介绍过,不同的构件有不同的属性、事件和方法,这里只是给出最常见的属性、事件和方法的功能,通过这些概念,就可以练习在 C++Builder IDE 中编制程序。

现在已经介绍了 C++Builder 的基本特点。要记住,C++Builder 的代码是 C++,全世界的人都说 C++不是好学的,C++Builder 不能将 C++变成容易学习的语言。但是,C++Builder 确实能将 C++语言学习变为有趣的计算机语言,它将枯燥的计算机语言变得亲切友好,因为只要设计几行代码,就可以看到程序的表现,这对于一个程序设计人员来讲,每天都有一种成就感、自豪感。编者在这里强调:老师布置的作业或老板安排的任务是程序设计人员的压力,成就感和自豪感是一个程序员勇往直前的动力。学习程序设计和程序员设计应用程序需要的是压力和动力的平衡。

现在设计一个简单的 C++Builder 程序,体会在 C++Builder 环境下程序设计的魅力。这里利用构件属性、

方法和事件的程序。

步骤如下：

(1) 打开 C++Builder 开发系统, 单击构件板 Standard 页面, 鼠标单击带“A”的构件, 再在窗体单击, 可以看到构件 Label 已经放在单击窗体上相应的位置, 在对象观测器和对象观测树中都能够看到它的名字是 Label1。单击构件 Label1, 拖动鼠标, 移动到适当的位置。在构件上单击构件就是选中这个构件。相应地在对象观测器和对象观测树中显示的就是当前选中的构件的属性, 当鼠标在这个构件上停留的时候会出现一个浅黄色的框, 闪现这个构件的名称、所属类、在窗体上的位置和其他属性。可以比较一下浅黄色的框中的内容和属性栏中的属性值。在窗体上放置一个构件就是在构件板中单击这个构件, 再在窗体上单击。

(2) 选中 Label1 (在构件上的一次单击选中该构件), 在对象观测器 (Object Inspector) 属性栏 (Properties) 找到 Caption 属性, 输入“解放军信息工程大学”。Caption 属性是在程序运行时构件要显示的内容。

(3) 设置构件 Label1 的 Font 属性, 单击 Font 属性旁边的带有省略号“...”的按钮, 出现字体设置对话框, 设置构件 Label1 将要显示的字体、字号及颜色。在对象观测器中带有省略号“...”的按钮说明单击这个按钮可以打开一个对话框, 通过对话框要完成某种操作。Font 对话框就是字体设置对话框, 和 offices word 中字体设置基本相同。熟悉 windows 编程的人员能够想到这是 C++Builder 开发系统调用了 windows 中的 API 函数。

(4) 在窗体上放两个 Button 按钮 (在构件板 Standard 页面, 带有“OK”字样), 将两按钮的 Caption 属性分别改为“显示”、“隐藏”, 也可以改变字体的大小 (在这里不能改变字体的颜色)。设计完成后窗体如图 1.6 所示。



图 1.6 设计完成后的窗体

(5) 在窗体上双击“显示”按钮, 出现代码编辑器, 且光标已经定位在能够输入代码的地方, 输入如下代码。

```
Label1->Show();
```

这就是“显示”按钮的事件处理器, 它的完整形式为:

```
void __fastcall TForm1::Button1Click(TObject *Sender) // Button1 的事件处理器
{
    Label1->Show(); // 输入的代码, 其中调用了 Label1 的 Show() 方法;
}
```

因为“显示”按钮的默认名字为 Button1, 即他的 Name 属性值为 Button1

(6) 单击窗体 (或按 F12 键切换到应用程序窗体, F12 是一个开关键, 连续按下它能够实现在窗体和代码编辑器之间的切换, F11 也是一个开关键, 用来打开和关闭对象观测树和对象观测器), 用同样的方法生成“隐藏”的代码处理器, 它的完整代码为:

```
void __fastcall TForm1::Button2Click(TObject *Sender) // Button2 的事件处理器
{
    Label1->Hide(); // 输入的代码, 其中调用了 Label1 的 Hide() 方法;
}
```

现在保存程序, 可以编译连接, 选择 C++Builder 主菜单的【Run】->【Run】、或快捷按钮上带绿色箭头的按钮、或 F9 (快捷键) 执行程序了, 分别点击两个按钮。当点击“显示”按钮时, 在 Label1 上显示“解放军信息工程大学”, 当点击“隐藏”按钮时, 显示文本消失。在这个简单的程序中, 应用了构件的属性、事件和方法。