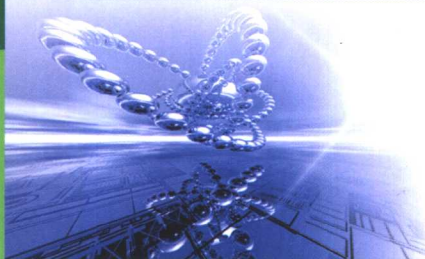


程序员指南丛书



C++ Builder 6 编程基础

- C++ Builder 6 扩展并增强了包括对分布式应用程序、Web 应用程序和数据库技术等支持的新特性。
- 用 Delphi 编写的任何代码几乎都可用在 C++ Builder 中，同样用 C++ Builder 编写的部分代码也可以用在 Delphi 中。
- 本书采用大量实例系统地介绍了 C++ Builder 6 在开发 Windows 应用程序、Web 应用程序以及数据库程序等方面的应用。

肖建等编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



C++ Builder 6 编程基础

肖建等编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书结合大量实例,系统地介绍了 C++ Builder 6 在开发 Windows 应用程序、Web 应用程序和数据库程序等方面的应用。首先介绍了 C++ Builder 6 的新特性、开发环境及 C++ 语言的编程基础与技巧;然后介绍 Windows 程序开发所必需的基本组件的使用技巧以及如何实现对文件属性的管理和操作;接着以专题的形式介绍 C++ Builder 6 在图像处理、多媒体、数据库、网络等方面的应用;最后简要介绍如何在 C++ Builder 6 中实现 COM 对象的开发和使用,并介绍了如何开发 OLE 自动化程序。

本书可以作为广大编程爱好者、IT 从业人士等的学习指导用书,也可以作为相关领域的培训教材。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无防伪标签者不得销售。

图书在版编目(CIP)数据

C++ Builder 6 编程基础/肖建等编著. —北京:清华大学出版社,2002.11

(程序员指南丛书)

ISBN 7-302-05916-0

I. C... II. 肖... III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 073861 号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编:100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 胡先福

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 19.75 字数: 493 千字

版 次: 2002 年 11 月第 1 版 2002 年 11 月第 1 次印刷

书 号: ISBN 7-302-05916-0/TP·3513

印 数: 0001 ~ 5000

定 价: 28.00 元

前 言

C++ Builder 6 在 Borland 公司一直倍受称誉的 ANSI C++ RAD 工具的基础上,扩展并增强了一系列新的属性,包括对分布式应用程序、Web 和数据库技术这些领先技术的支持。这些新属性主要体现在以下几个方面:

- e-Business 时代 C++ 快速开发工具
- 对 Web Services 技术的支持
- 便捷强大的 Web 应用程序开发能力
- 建构高效能多层数据库 Web Services 应用系统
- 对 Linux 操作系统的支持
- 强大的除错(Debug)功能
- 方便快捷的集成开发环境
- 精简快速的 InterBase 数据库

从上面这些新特性可以看出,Borland 公司一直紧密地跟随着 e 时代发展的脚步,当 Microsoft 公司推出 .NET 体系时,Borland 公司也就开始了支持 .NET 体系的工作。Web Services 和 Linux 是今后计算机领域发展的主题和趋势,C++ Builder 6 能够在这两个方面有长足的进步,让广大的 C++ Builder 爱好者更增强了信心。

C++ Builder 的界面与它的同类产品 Delphi 有着很大的相似之处。Delphi 使用的是 Object Pascal 语法,它具有易学易用的特性,能使初学者迅速编写出表现力极强的程序;而 C++ Builder 对于高水平的程序员来说无疑是优秀的开发工具,它最大程度体现了 C++ 语言的灵活性和适用性,大大提高了程序的运行效率。如果能够把这两种语言结合使用势必会大大提高程序的开发效率。

用户完全可以在 C++ Builder 和 Delphi 之间随心所欲地进行多平台程序开发。Delphi 和 C++ Builder 之间具有十分密切的关系,它们在性能上非常相似:用 Delphi 编写的任何代码几乎都可用在 C++ Builder 中,同样用 C++ Builder 编写的部分代码也可以用在 Delphi 中。这样在编写程序时,程序员们就可以根据自己的喜好和特长决定使用哪种语言,而不必担心使用一种语言编写的代码不被另一种语言接受,编写好之后只要遵循一定的规则把它们连接在一起即可使用。

本书系统地介绍了 C++ Builder 6 在开发 Windows 应用程序、Web 应用程序以及数据库程序等方面的应用,书中采用了大量的实例进行讲解,这些实例基本上能够体现 C++ Builder 6 的新特性,以及它区别于其他开发工具的独特性能。

全书内容安排如下:

第 1 章简要介绍了 C++ Builder 6 的新特性及开发环境。

第 2 章介绍了 C++ 语言的编程基础与技巧。

第3~5章从C++ Builder 6的组件角度出发,介绍了Windows应用程序开发所必需的基本组件的使用技巧。

第6章结合API函数介绍如何在C++ Builder 6中实现对文件属性的管理和操作,同时还开发了一个类似于Windows写字板的文字处理工具,对前面几章所学知识进行总结。

第7~10章以专题的形式介绍C++ Builder 6在图像处理、多媒体、数据库、网络等方面的应用。这些章节都是C++ Builder 6最让人感兴趣的地方,也是其有别于其他开发工具,突显其便捷性、实效性的最好体现。

第11章简要介绍如何在C++ Builder 6中实现COM对象的开发和使用,并介绍了如何开发OLE自动化程序。

除封面署名外,参与本书编写、校对等工作的还有王玉珍、杜薇、赵大龙、孙文军、蔡亚胜、刘翠华、杨乃平、李晓琳、叶清、张瑞年等人。由于水平所限,书中难免会有不足和纰漏之处,希望广大读者批评指正。

作者
2002年9月

目 录

第 1 章 C++ Builder 6 简介	1
1.1 C++ Builder 6 新特性	1
1.2 对象和事件的基本概念	4
1.2.1 对象	4
1.2.2 属性	5
1.2.3 事件	6
1.2.4 方法	6
1.3 C++ Builder 6 集成开发环境	7
1.3.1 C++ Builder 6 主窗口	8
1.3.2 代码编辑窗口	9
1.3.3 窗体设计窗口	11
1.3.4 对象查看窗口	13
1.3.5 项目管理窗口	14
1.4 简单的 C++ Builder 示例	15
1.4.1 程序设计基本方法	15
1.4.2 提交数据的示例	16
第 2 章 C++ 编程基础与技巧	21
2.1 C++ 的文件结构	21
2.1.1 头文件的结构	21
2.1.2 定义文件的结构	22
2.1.3 头文件的作用	22
2.1.4 目录结构	22
2.2 表达式和基本语句	23
2.2.1 运算符的优先级	23
2.2.2 复合表达式	24
2.2.3 if 语句	24
2.2.4 循环语句的效率	26
2.2.5 for 语句的循环控制变量	27
2.2.6 switch 语句	28
2.2.7 goto 语句	28
2.3 常量	29

2.3.1	为什么需要常量	29
2.3.2	const 与 # define 的比较	29
2.3.3	常量定义规则	30
2.3.4	类中的常量	30
2.4	函数设计	31
2.4.1	参数的规则	31
2.4.2	返回值的规则	32
2.4.3	函数内部实现的规则	34
2.4.4	使用断言	35
2.4.5	引用与指针的比较	36
2.4.6	指针与数组的对比	37
2.5	函数重载和内联	39
2.5.1	函数重载的概念	39
2.5.2	成员函数的重载与覆盖	41
2.5.3	参数的默认值	42
2.5.4	运算符重载	43
2.5.5	函数内联	44
2.6	类的构造函数、析构函数与赋值函数	45
2.6.1	构造函数与析构函数的起源	45
2.6.2	构造函数的初始化表	46
2.6.3	构造和析构的次序	48
2.6.4	String 类的构造函数与析构函数	48
2.6.5	String 类的拷贝构造函数与赋值函数	48
2.6.6	在派生类中实现类的基本函数	50
2.7	类的继承与组合	51
2.7.1	继承	51
2.7.2	组合	52
2.8	C++ 的 I/O 流	53
第 3 章	窗体设计与基本组件	55
3.1	组件及对象	55
3.1.1	组件	55
3.1.2	组件的分类	57
3.1.3	C++ Builder 的标准组件	57
3.1.4	对象	58
3.2	Form(窗体)组件	58
3.3	按钮类组件	61
3.4	标签和编辑组件	64
3.4.1	用标签和编辑框设计窗体	64

3.4.2	动态设置编辑框属性	66
3.4.3	掩码编辑框	67
3.4.4	Memo 组件和 RichEdit 组件	69
3.5	单选按钮和复选框	73
3.5.1	单选按钮和复选框	73
3.5.2	单选按钮组	75
3.6	列表框和组合列表框	77
3.6.1	列表框	77
3.6.2	自绘列表框	82
3.6.3	组合列表框	84
第 4 章	菜单设计	89
4.1	MainMenu Designer	89
4.2	动态改变菜单	93
4.2.1	菜单项的有效和无效	93
4.2.2	复选标记	94
4.2.3	隐藏菜单	95
4.2.4	单选菜单项	96
4.2.5	动态创建菜单项	97
4.3	打开和存储文件的菜单项	100
4.4	设计快捷菜单	101
4.5	使用菜单模板	104
4.6	图形菜单项	105
4.6.1	定制图形复选框标记	105
4.6.2	位图菜单项	108
第 5 章	特殊窗体与对话框设计	112
5.1	利用模板生成窗体	112
5.2	多页窗体	114
5.2.1	使用 TPageControl 建立多页窗体	114
5.2.2	使用 TTabControl 建立多页窗体	122
5.2.3	使用 TNotebook 和 TTabSet 建立多页窗体	124
5.2.4	使用 TTbbedNotebook 建立多页窗体	129
5.3	建立分割窗体	130
5.4	模式对话框	131
5.5	无模式对话框	136
5.6	消息框	139
5.7	通用对话框组件	141
5.7.1	文件对话框组件	141

5.7.2	字体对话框组件	144
5.7.3	颜色对话框组件	145
5.7.4	打印和打印设置对话框组件	146
5.7.5	查找及替换对话框组件	146
第 6 章	文件操作	151
6.1	文件属性	151
6.1.1	文件管理组件	151
6.1.2	文件属性管理器	152
6.2	文件复制	158
6.2.1	文件流类	158
6.2.2	内存流类	159
6.2.3	文件复制器	161
6.3	写字板程序	165
6.3.1	界面设计	165
6.3.2	编写程序代码	169
第 7 章	图形图像编程	180
7.1	图形设计综述	180
7.2	基本图形绘制	181
7.2.1	直线的绘制	181
7.2.2	椭圆的绘制	183
7.2.3	多边形绘制	185
7.2.4	矩形的绘制	187
7.3	图像特效	189
7.3.1	镜像效果	189
7.3.2	推进、交错、百叶窗等效果	190
7.4	图像的特殊效果处理	199
7.4.1	新建工程与预处理	200
7.4.2	图像的柔化处理	201
7.4.3	图像的锐化处理	202
7.4.4	图像的浮雕效果	204
7.5	OpenGL 编程	205
7.5.1	Win32 下 OpenGL 的程序设计技巧	205
7.5.2	新建工程与预处理	210
7.5.3	编写程序代码	211
第 8 章	多媒体编程	215
8.1	多媒体组件	215

8.1.1	TMediaPlayer 组件	215
8.1.2	TMediaPlayer 与 TrackBar 组件结合	221
8.2	音频播放器	223
8.2.1	新建工程	223
8.2.2	编写程序代码	224
8.3	CD 播放器	228
8.4	使用 MCI 函数播放 CD	231
8.4.1	Windows 低级音频函数介绍	231
8.4.2	MCI 简介	232
8.4.3	使用 MCI 编程实例	233
第 9 章	数据库应用	235
9.1	数据库管理系统	235
9.2	C++ Builder 数据库体系结构	236
9.3	基本数据库组件	236
9.3.1	TTable 和 TQuery 组件	236
9.3.2	TDataSource 组件	241
9.3.3	TDataBase 组件	242
9.3.4	TStoredProc 组件	243
9.3.5	TSession 组件	244
9.3.6	TDBText 组件	244
9.3.7	TDBEdit 组件	244
9.3.8	TDBGrid 组件	245
9.3.9	TDBNavigator 组件	247
9.3.10	TField 组件	247
9.4	使用 TTable 组件编程	252
9.5	使用 TQuery 组件编程	256
9.5.1	SQL 属性	256
9.5.2	动态 SQL 表达式	260
第 10 章	网络编程	265
10.1	Socket 编程	265
10.1.1	TClientSocket 组件	265
10.1.2	TServerSocket 组件	267
10.1.3	聊天程序	268
10.2	TNMPUDP 组件	273
10.2.1	TNMPUDP 组件介绍	273
10.2.2	远程屏幕抓取程序	275
10.3	TIdTrivialFTP 组件	281

10.3.1 TIdTrivialFTP 组件介绍	281
10.3.2 TFTP 客户端程序	283
10.4 TIdTelnet 组件	284
第 11 章 简单 COM 编程	288
11.1 COM 技术基础	288
11.1.1 COM 应用的组成	289
11.1.2 COM 扩展	291
11.1.3 通过向导实现简单的 COM 对象	292
11.2 使用 COM 对象编程	292
11.2.1 使用向导创建 COM 对象	292
11.2.2 基于 COM 对象的服务器程序	295
11.2.3 基于 COM 对象的客户端程序	298
11.3 OLE 自动化	300
11.3.1 通过引入类型库创建一个自动化控制器	301
11.3.2 通过变体(Variants)类型创建一个自动化控制器	302

第 1 章 C++ Builde 6 简介

随着时代的发展和计算机技术的进步,人和计算机之间的距离越来越小,计算机和用户的界面也越来越亲切。大部分人只要经过短时间的学习就能够掌握操作计算机的基本技能,这在很大程度上归功于可视化操作系统的出现,例如 Windows、Linux 等操作系统。目前 Windows 为最广泛使用的系统,因此本书介绍的是基于 Windows 操作系统的应用程序开发。

在 Windows 操作系统出现的早期,想要在 Windows 下写出一个漂亮的 Windows 应用程序,对绝大部分人来说都只能是一个梦想,只有少数一批优秀的程序设计师才有能力完成这些工作。经过软件公司不断的改进和努力,引进了 RAD(Rapid Application Development,快速应用程序开发环境)程序开发工具,并集成在 IDE(Integrated Develop Enviroment,集成开发环境)中。配合可视化开发工具,程序设计师能够在 Windows 环境下轻松快速地开发出 Windows 应用程序,而一些普通的计算机爱好者,也能够在经过短时期的学习之后,就掌握编写基于 Windows 程序的方法。

Borland 公司在这方面一直走在世界的前列,它所开发的 Delphi、C++ Builde 以及 JBuilder 等程序设计软件都一直为世人所称道。本书介绍的就是 Borland C++ Builde 最新版本 6 的基本知识。首先本书将比较全面系统地介绍有关 Borland C++ Builde(BCB) 6 的新特性和在 BCB 6 环境下开发 Windows 应用程序的一般过程。

1.1 C++ Builde 6 新特性

C++ Builde 6 在 Borland 公司一直倍受称誉的 ANSI C++ RAD 工具的基础上,扩展并增强了一系列新的特性,包括对分布式应用程序、Web 和数据库技术这些领先技术的支持。这些新特性包括以下几个方面:

e-Business 时代的 C++ 快速开发工具

主要功能:

- 支持 Web Services,简化企业与企业(B2B)间应用整合;
- 提升 Web 应用程序开发效率;
- 构建存取数据库并提供 Web Services 的高效能中介软件(Middleware);
- 支持企业标准 XML、SOAP、WSDL 与 XSL;
- 轻松开发跨越 Windows/Linux 平台的 C++ 应用程序;
- 高效能 32 位 C++ 编译器;
- 支持 Microsoft Access、IBM DB/2 与 Informix、FoxPro、Borland InterBase、Microsoft SQL Server、MySQL、Borland MyBase、Oracle、Paradox 以及 Sybase 等各种数据库。

C++ Builder 6 提供了一个稳固、高效率的 C++ 开发环境,能帮助用户运用新兴 Web Services 技术,开发高效率的 e-Business 应用系统。C++ Builder BizSnap 技术是一个以 Web Services 为基础的开发平台,它运用企业标准 SOAP/XML Web Services,将用户企业内的应用系统与下游客户及上游产业供应商间天衣无缝地整合,大幅简化了传统企业对企业(Business to Business, B2B)间应用系统整合的复杂度。C++ Builder WebSnap 技术是针对 Web 应用所提出的开发平台,通过运用新式软件组件技术,能有效地简化开发过程中各环节的复杂度,帮助开发人员快速开发出功能丰富、高效能的 Web 应用程序。C++ Builder DataSnap 技术能提供一个高效能的资料存取解决方案,帮助用户整合企业内各类作业流程以及企业外众多合作伙伴。

对 Web Services 技术的支持

C++ Builder BizSnap 技术提供了完整的 Web Services 开发环境,使进入 e-Business 的新时代一点也不难。BizSnap 相关组件与工具可以轻松转换现有企业信息成为 XML 文件,让企业在 B2B 的潮流中完全不受限制与拘束。不论用户的企业伙伴使用何种应用系统, C++ Builder BizSnap 采用企业标准 SOAP/XML,将企业间的沟通整合,丝毫不因多变、异质 IT 解决方案而受阻。BizSnap 使 C++ Builder 能够借由标准的 SOAP/XML Web Services 与 Microsoft .NET、Microsoft BizTalk 以及 Sun Microsystems ONE 等应用服务相整合。

便捷强大的 Web 应用程序开发能力

C++ Builder WebSnap 技术让用户在开发 Web 应用程序时享有 RAD 工具独有的便利性。WebSnap 的除错服务器,无须借助额外 Web 服务器,即可在 C++ Builder 集成开发环境内提供 Web 应用程序完整除错功能。用户可以使用 Microsoft FrontPage 或 Macromedia Dreamweaver 等网页制作工具制作画面及撰写服务器端脚本语言,借由 WebSnap 相关组件与 VBScript、JScript 等任何 Microsoft Active Script 所支持的脚本语言直接整合,同时获得脚本语言的弹性与 C++ Builder 的执行效率。Web Page 外观设计向导,能够以可视化工具与软件组件结合方式,快速开发整合数据库及动态网页的 Web 应用程序。

构建高效能多层数据库 Web Services 应用系统

C++ Builder DataSnap 技术允许用户运用 SOAP、CORBA、COM、TCP/IP 等分布式运算标准,开发多层分布式数据库应用系统。搭配 DataSnap 所开发出的中介软件(Middleware),使企业内用户端应用程序无需任何组态设定,可降低用户端维护成本;搭配 Web Services 更能够提供企业外的相关厂商,流畅地交换企业信息。DataSnap 中介软件内各项技术能对数据库连接(Connection)进行有效的管理,并对频宽控制达到最佳化的效果,亦为 Web Services 整合各种企业级数据库提供了最佳的途径,将企业内应用系统延伸成为 e-Business 应用系统。

对 Linux 操作系统的支持

兼具高可靠性和高延展性并且免费的 Linux 已成为今日最具竞争力的操作系统。在 Web 等服务器应用中, Linux 是经过市场验证的成熟解决方案;在桌面应用中, Linux 也快速成为 Microsoft Windows 操作系统的替代方案。Borland 计划在 2002 年上半年推出 Linux 平台

的 C++ 快速开发工具,而 C++ Builder 6 将与 Linux 平台 Borland C++ 开发工具采用相同的 CLX 跨平台组件库,两者原始程序代码将完全相容。C++ Builder 6 允许用户运用相同的技巧及知识,即可完成横跨 Linux 与 Windows 两大操作系统的应用程序,保障用户的技术投资带来最大的投资回报。

轻松连接企业内数据库

C++ Builder 6 内建多种数据库存取方式,可跨 Linux 平台的 dbExpress 驱动程序,提供高速存取 Oracle、IBM DB/2 与 Informix、Borland InterBase 及 MySQL 的能力,新版 Borland Database Engine 提供高效能 SQLLink 驱动程序来存取 Oracle、IBM DB/2 和 Informix、Microsoft SQL Server、Sybase 与 Borland InterBase,并对 Oracle 对象关联延伸功能提供完整的支持,dbGo 系列组件可透过 Microsoft ADO 存取各类资料来源。无论用户使用 dbExpress、Borland Database Engine 还是 dbGo,皆可运用 C++ Builder 6 功能丰富的数据感知组件(Data aware component),快速开发出以数据为中心的 Windows 应用程序或 Web 应用程序。

企业级 e-Business 解决方案

Sun Microsystems 所提出的服务器端组件架构 Enterprise JavaBean(EJB)是 Java2 Enterprise Edition(J2EE)架构中的核心。Java 跨越操作系统的特性以及自动内存管理特性,使得 Java 在服务器端应用获得企业广泛的支持。C++ Builder 6 能够借由 Borland Enterprise Server SIDL 功能,直接使用挂载于 Borland Enterprise Server 内的 EJB 对象。用户可以运用 C++ Builder 6 开发 Windows 应用程序、Web 应用程序或 SOAP/XML Web Services,桥接符合 Sun Microsystems J2EE 相容认证的 Borland AppServer,将平台无关性、负载平衡、丛集(Clustering)容错等企业级 Java 解决方案直接与微软 Windows 环境相整合。

C++ Builder 6 提供符合 ANSI/ISO 标准、功能强大的 C++ 编译器,增加支持 STLPort,并整合功能完整、高生产力的可视化开发环境,包含画面设计、对象查看器、专案管理员以及图形化除错器。C++ Builder 6 依循面向对象精神,构建以 VCL/CLX 软件组件为基础的开发架构,让程序代码可重复使用的能力大幅提升,缩短了整个软件开发周期的时程。

强大的除错(Debug)功能

C++ Builder 6 强化了以前版本中断点管理的功能,支持多进程(Process)除错和 COM+ 对象除错,内建 Code Guard 向导能自动侦测内存泄漏的错误程序。C++ Builder 6 Enterprise 版针对多层分布式开发环境提供远端除错的能力,开发人员可以通过网络直接对远端的 COM 对象进行除错,让复杂的多层应用系统除错变得更容易。

方便快捷的集成开发环境

C++ Builder 6 大幅提升了 Code Completion 向导执行速度,能自动列出组件属性、方法清单,减少错误发生;Code Insight 向导将常用程序代码定义于程序样板中,按下单一键即可插入程序代码中;全新 Object Tree View Windows 让用户更方便地选取程序中用到的组件;Object Inspector 增添关联功能,若组件某属性值为其他的组件,可直接在 Object Inspector 查看和设定组件的属性值。

我们将会在后面的小节中对 C++ Builder 6 的集成开发环境进行简单的介绍。

精简快速的 InterBase 数据库

C++ Builder 6 内附的 InterBase 6.5 是功能完整的数据库管理系统,支持标准 ANSI SQL,以最少的系统资源提供最佳的执行效能。支持各种操作系统的 InterBase 6.5 几乎不需要专职的数据库系统管理员,简单易用的特色领先众多的数据库管理系统,并且能提供与企业级数据库相同的强大功能。

C++ Builder 6 的各种版本

不论是初学 Windows 程序设计的学生或是开发关键性任务、大型应用系统的企业开发人员,C++ Builder 6 完整的产品线可满足不同类型开发人员的需求。C++ Builder 6 提供以下版本。

- C++ Builder 6 Enterprise 版:适用于企业开发人员、增值软件厂商(VAR)、系统整合公司等需要构建新时代 e-Business 应用系统的技术人员,全新 BizSnap 提供 B2B 电子商务的必要功能,WebSnap 将 RAD 开发工具带入 Web 应用程序开发领域,DataSnap 以多层式架构整合企业信息与电子商务等各种新应用。

- C++ Builder 6 Professional 版:适用于需开发商业数据库应用程序、或欲进入新兴 Linux 市场的专业开发人员,保障用户在 Windows 平台上的开发投资轻易移植至 Linux 平台。

- C++ Builder 6 Personal 版:提供初学者以可视化开发工具,轻松学习 Windows 程序设计,快速开发进入 Windows 程序设计领域。此版本仅授权供学习用途。

C++ Builder 6 Enterprise 版最低系统需求

- Intel PentiumII 400MHz 以上微处理器
- Windows 98、Windows 2000(ServicePack 2 以上)或 Windows XP
- 128MB 内存(建议使用 256MB 以上)
- 完整安装需要 750MB 硬盘空间
- SVGA(800 × 600 分辨率,256 色)或更高级的显示卡
- 鼠标或其他相容设备

1.2 对象和事件的基本概念

面向对象(Object Oriented)就是把现实世界中的事物以及这些事物所特有的一些性质,用更接近自然的形式在计算机世界中反映出来。每一个对象都有自己的属性(Property)和方法(Method),对象和对象之间通过方法来传递对象的消息(Message)。

1.2.1 对象

世界上所有的具体或抽象的事物,基本上都可以被视为“对象”。例如,一个人可以被视为一个大的对象,身体中的头部、胸部、四肢等部位可以被视为比较小的对象,四肢又可细分

为手臂、脚等对象,手又可分为上臂、小臂、手掌等更小的对象……这样不断细分。

C++ Builder 中的对象由一些程序代码和数据组成,可以被视为一个完整的单元。对象可以是应用程序的一部分,例如窗体、鼠标、将组件面板中的组件放置到窗体上所产生的组件对象、菜单以及一些不可见组件所生成的组件对象等。

例如,在某个简单的 C++ Builder 应用程序中,我们在窗体中添加了一些组件,这时候在 Object TreeView 窗口就可以看到该应用程序中的所有对象,如图 1-1 所示。

可以看到,在该应用程序中,我们有如下对象:

```
object Form1: TForm1
  object Label1: TLabel
  object Button1: TButton
  object CheckBox1: TCheckBox
  object MainMenu1: TMainMenu
    object File1: TMenuItem
      object Exit1: TMenuItem
    object Project1: TMenuItem
      object Reset1: TMenuItem
    object Run1: TMenuItem
```

这些对象都可以在该应用程序窗体的 dfm 文件中看到。

1.2.2 属性

在 C++ Builder 中,每个对象都有自己的属性,我们可以使用这些属性来设置这个对象的外观。例如,窗口的属性包括窗口的大小、颜色、标题等。请看下面的窗口属性描述代码:

```
object Form1: TForm1
  Left = 192
  Top = 107
  Width = 531
  Height = 328
  Caption = '窗体属性'
  Color = clGradientActiveCaption
  Font.Charset = ANSI_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Italic'
  Font.Style = []
  Menu = MainMenu1
  OldCreateOrder = false
  OnCreate = FormCreate
  PixelsPerInch = 96
  TextHeight = 16
```

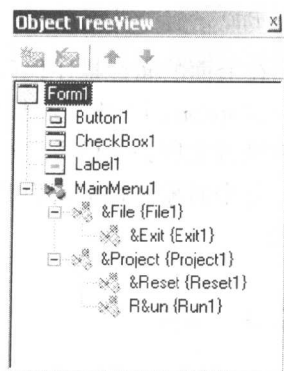


图 1-1 对象

End

在上面的属性描述代码中, Left、Top、Width 和 Height 这四个属性用来控制窗体的位置和大小, Caption 用来设置其标题, Color 用来设置其颜色, Font 用来设置其上文字的字体, Menu 用来设置它所关联的主菜单等。

每个对象属性大都有其默认值,有些对象属性在程序设计阶段或执行阶段都可以进行设置。

1.2.3 事件

在 C++ Builder 中,只要在某个 Enable 属性为 true 的组件对象上单击或进行其他操作,就会触发一个事件。当 C++ Builder 收到属于某个对象的事件时,它就会根据此事件做出相应的处理,其处理的方式就是去执行程序开发者为该对象所编写的相应事件处理程序代码,也可以称之为“事件函数”。

例如,我们编写某个窗体中 Button1 按钮的单击事件处理程序如下:

```
void __fastcall TForm1::Button1Click(TObject * Sender)
{
    if(CheckBox1 -> Checked == true)
        Label1 -> Caption = "hello,BCB!";
    else
        Label1 -> Caption = "hello!";
}
```

运行该程序时,当用户单击了 Button1 按钮之后,程序就会运行上面的代码,检查 CheckBox1 对象是否被选中,然后根据选中与否的状态给出不同的结果。

1.2.4 方法

所谓方法,是指对象完成某件事或某个目标所采取的处理方式。在 C++ Builder 中,每个对象都有一些方法用来改变该对象的外观和内部行为。有了对象的方法,我们可以不需要知道真正的数据结构以及其他相关的事项就可以实现所要完成的任务。

例如,在窗体对象中提供了打印、画线等功能,这些功能统称为方法。将 1.2.3 节所编写的事件处理程序改写如下:

```
void __fastcall TForm1::Button1Click(TObject * Sender)
{
    if(CheckBox1 -> Checked == true)
        Label1 -> Caption = "hello,BCB!";
    else
        Label1 -> Caption = "hello!";
    Form1 -> Canvas -> LineTo(23,120);
}
```