

C++ Builder 4

应用程序开发

肖勇 龙玺 徐健 张翼飞 等编著



人民邮电出版社

C++ Builder 4 应用程序开发

肖 勇 龙 玺
徐 健 张翼飞 等编著

人 民 邮 电 出 版 社

内 容 提 要

本书是一本专门介绍 C++ Builder 4 应用程序开发技术的图书, 主要内容包括: C++ 语言简介、安装和卸载 C++ Builder 4、C++ Builder 4 的集成开发环境、设计窗体和菜单、VCL 构件、创建 C++ Builder 4 应用程序、调试工具动态链接库、程序装饰、创建构件、C++ Builder 4 数据库结构、SQL 语言基础、建立数据库应用程序、快速报表、图形编程、多媒体编程、网络编程、C++ Builder 4 程序打包等等。

本书内容丰富、实用性强, 可供广大计算机用户和从事编程工作的人员阅读参考。

155/06

C + + Builder 4 应用程序开发

- ◆ 编 著 肖 勇 龙 玺 徐 健 张翼飞 等
责任编辑 王 晓明
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 27.5
字数: 678 千字
印数: 1—6 000 册
- 2000 年 1 月第 1 版
2000 年 1 月北京第 1 次印刷
- ISBN 7-115-08347-9/TP·1489

定价: 41.00 元

前 言

Borland C++ Builder 4 是 Borland 公司更名为 Inprise 后推出的一个具有战略意义的产品,它不仅秉承了 Inprise (Borland) 公司原来的拳头产品——Delphi 在数据库开发方面的强大功能的优势,而且对目前十分流行的面向对象的设计方法和 C++ 语言进行了全面集成,从而结束了长期以来广大 C++ 程序员没有快速应用开发(PAD)工具可用的局面。正是由于这些特点,使得 C++ Builder 4 已成为当今最热门的编程开发工具之一。

本书全面、深入地介绍了 Borland C++ Builder 4 的各种功能和新增特性,并介绍了一般的 Windows 应用程序开发、数据库开发、多媒体开发以及网络开发等方面的知识和技术,基本涵盖了 Windows 编程领域内的所有高级特性。在本书的编写过程中,作者力求深入浅出地对每一部分进行介绍,目的是使读者不仅能够熟练地掌握编程技巧,而且能够对 Windows 编程的精髓有所了解,从而对 Borland C++ Builder 今后的新版本或其它编程工具能够触类旁通。另外,为了帮助读者加深对本书内容的理解,书中提供了大量实例,这些实例中的程序都是直接从实际开发项目中精选出来的,有很强的实用性。作者希望本书无论是对那些想了解 Windows 编程美妙世界的初学者,还是对已有多年 C++ 编程经验的程序员,甚至对已有较丰富的 Delphi 编程经验的程序开发者来说,都能有一定的参考价值。

本书主要由肖勇、龙玺、徐健和张翼飞撰写,参加本书编写工作的还有冯峰、姜汉卿、郑卫国、刘树勋、赵作智、万民、文小江、姜浩、罗志云、张宏声、柏岩、葛行飞、刘宁、林东海、杨超、李华、宋建宏、贺晓辉、刘俐玲、马英杰、熊克艳、朱家权、景兴碧、李勇军、吴德刚等。

最后需要说明的是,由于计算机应用软件技术发展得太快,加之作者水平有限,因此书中难免出现错误和不妥之处,在此敬请读者批评指正。

作 者

1999 年 11 月

目 录

第一章 C++语言简介	1
1.1 C++语言初步	1
1.1.1 C++语言概览	1
1.1.2 if 与 switch 条件表达式	8
1.1.3 switch 条件表达式	10
1.1.4 循环控制	10
1.1.5 do-while 循环	11
1.1.6 new 和 delete 操作符	12
1.1.7 指针与引用	13
1.2 C++ 类	14
1.2.1 类的定义	14
1.2.2 类的解释	15
1.2.3 继承	30
1.2.4 多重继承	30
1.2.5 基本文件输入/输出	31
1.3 C++ 类库和可视化构件模型	31
1.3.1 C++类库	31
1.3.2 可视构件库	32
1.3.3 C++ Builder 和 VCL	34
第二章 安装和卸载 C++ Builder 4	35
2.1 系统要求	35
2.2 安装 C++ Builder 4	35
2.2.1 安装程序的启动	35
2.2.2 安装主界面	36
2.2.3 安装信息	37
2.2.4 安装确认	38
2.2.5 安装选项	40
2.2.6 安装路径和文件夹选择	43
2.2.7 复制文件	44
2.3 运行 C++ Builder 4	46
2.3.1 创建 BCB4 快捷方式	46
2.3.2 运行 C++ Builder 4	47
2.4 卸载 C++ Builder 4	47

2.4.1	启动卸载工具	47
2.4.2	卸载 C++ Builder 4	48
第三章	C++ Builder 4 的集成开发环境	49
3.1	C++ Builder 4 IDE 概述	49
3.1.1	IDE 定义	49
3.1.2	IDE 结构	49
3.2	C++ Builder 4 项目	50
3.2.1	项目的文件	50
3.2.2	C++ Builder 4 主菜单和工具栏按钮	51
3.3	项目组管理	52
3.3.1	项目组	52
3.3.2	建立项目组	53
3.3.3	项目选项	53
3.4	代码编辑器	57
3.4.1	基本操作	57
3.4.2	几个编辑器特性	59
3.4.3	编辑器选项	59
3.5	构件面板使用技巧	61
3.5.1	修改构件面板	61
3.5.2	放置多个构件	62
3.6	类观察器	63
3.6.1	类观察器显示	63
3.6.2	类观察器的操作	64
3.7	对象观察器	65
3.7.1	构件选择器	65
3.7.2	属性页	66
3.7.3	事件页面	67
3.8	C++ Builder 窗体	67
3.8.1	窗体背景知识	67
3.8.2	窗体的读写属性	68
3.8.3	窗体的只读属性	68
3.8.4	常用的窗体方法	69
3.8.5	常用窗体事件	69
第四章	设计窗体和菜单	70
4.1	设计窗体	70
4.1.1	概述	70
4.1.2	窗体弹出菜单	70
4.1.3	放置和选择构件	71
4.1.4	移动和缩放构件	72

4.1.5	对齐构件	73
4.1.6	设置跳表顺序	74
4.2	设计菜单	75
4.2.1	概述	75
4.2.2	设计主菜单	75
4.2.3	编写代码	78
4.2.4	弹出式菜单的设计	78
4.2.5	制作菜单模板	79
第五章	VCL 构件	80
5.1	VCL 构件的公用属性	80
5.1.1	引言	80
5.1.2	Name 属性	80
5.1.3	Color 属性	81
5.1.4	光标 (Cursor) 属性	81
5.1.5	Enabled 属性	82
5.1.6	Font 属性	82
5.1.7	Hint 属性	83
5.2	VCL 构件的主要方法和事件	83
5.2.1	VCL 构件的主要方法	83
5.2.2	VCL 构件的主要事件	84
5.3	标准的 Windows 控件构件	85
5.3.1	编辑构件	85
5.3.2	ListBox 和 ComboBox 构件	87
5.3.3	按钮	88
5.3.4	单选钮与复选钮	91
5.3.5	Label 构件	92
5.3.6	Panel 构件	92
5.4	对话框构件	92
5.4.1	Execute 方法	93
5.4.2	File Open 和 File Save 对话框	93
5.4.3	图形对话框	95
5.4.4	Find 和 Replace 对话框	95
5.4.5	Print 和 Printer Setup 对话框	96
5.5	多页面构件	97
5.5.1	多页面界面概述	97
5.5.2	TabControl 构件	98
5.5.3	PageControl 构件	99
5.5.4	Notebook 构件和 TabSet 构件	101
5.5.5	TabbedNoteBook 构件	104

第六章 创建 C++ Builder 4 应用程序	105
6.1 对象库	105
6.1.1 对象库概述	105
6.1.2 对象库页面	105
6.1.3 对象库视图	108
6.1.4 生成新对象	109
6.1.5 添加对象	110
6.1.6 管理对象库	111
6.2 窗体和应用程序向导	112
6.2.1 向导概述	112
6.2.2 对话框向导	112
6.2.3 应用程序向导	113
6.3 添加函数和数据成员	116
6.3.1 C++ Builder 中的类声明	116
6.3.2 添加函数	118
6.3.3 添加数据成员和删除代码	119
6.4 构件模板	119
6.4.1 构件模板概述	119
6.4.2 生成构件模板	119
6.5 资源文件	120
6.5.1 资源文件分类	120
6.5.2 使用资源文件	121
6.5.3 一个实例	121
6.6 包 (Packages)	129
6.6.1 包 (Packages) 概述	129
6.6.2 静态链接和动态链接	129
6.6.3 使用运行包	130
第七章 调试工具	132
7.1 调试工具	132
7.1.1 调试工具概述	132
7.1.2 弹出菜单的调试选项	132
7.1.3 主菜单中的调试选项	132
7.2 设置断点	133
7.2.1 设置和取消断点	133
7.2.2 断点列表窗口	134
7.3 变量跟踪	135
7.3.1 Watch List 列表框	135
7.3.2 使用 Watch List 列表框	136
7.4 使用 Debug Inspector	137

7.4.1	启动 Debug Inspector	137
7.4.2	使用 Debug Inspector	138
7.5	其他调试工具	139
7.5.1	Evaluate/Modify 调试工具	139
7.5.2	调用堆栈	140
7.5.3	浏览 CPU	140
7.5.4	使用 Go to Address	140
7.5.5	DLL 查错	141
7.5.6	事件日志 (Event Log)	141
7.5.7	模块视图 (Modules)	142
7.6	调试技术	142
7.6.1	单步调试	142
7.6.2	访问失败错误分析	143
7.6.3	调试提示	143
7.6.4	设置调试选项	144
第八章	动态链接库	147
8.1	DLL 概述	147
8.1.1	DLL 定义及分类	147
8.1.2	DLL 的优点	147
8.2	DLL 中的函数	148
8.2.1	装入 DLL	149
8.2.2	DLL 中的函数调用	149
8.3	生成 DLL	150
8.3.1	输入和输出函数与类	150
8.3.2	生成 DLL	151
8.3.3	生成输入库文件	154
8.4	DLL 调用应用程序的生成	155
8.4.1	生成应用程序	155
8.4.2	加进 DLL 头文件	155
8.4.3	加进输入库文件	155
8.5	DLL 中的窗体	156
8.5.1	C++ Builder 应用程序调用	156
8.5.2	调用 DLL 中的 MDI 窗体	156
8.5.3	非 C++ Builder 应用程序调用	158
8.6	DLL 中的资源	158
8.6.1	生成资源 DLL	159
8.6.2	使用资源 DLL	159
第九章	程序装饰	161
9.1	窗体装饰技术	161

9.1.1	工具栏的使用	161
9.1.2	状态栏的使用	165
9.2	应用程序的打印	169
9.2.1	窗体的 Print()方法	169
9.2.2	RichEdit 构件的 Print()方法	169
9.2.3	Tprinter 类和 Printere()方法	169
9.2.4	打印位图	170
9.3	光标的使用	171
9.3.1	光标概述	171
9.3.2	固有光标	171
9.3.3	定制光标	172
9.4	上下文帮助	172
9.4.1	编写帮助文件	173
9.4.2	情境标识符和 HelpContext 属性	173
9.4.3	生成上下文帮助	174
9.4.4	帮助头文件	175
9.4.5	实例	175
9.5	异常处理技术	176
9.5.1	异常处理关键字	176
9.5.2	catch 剖析	177
9.5.3	未处理异常	178
9.6	消息处理	179
9.6.1	消息概述	179
9.6.2	消息发送	180
9.6.3	消息处理	180
9.6.4	消息处理函数	181
9.6.5	自定义消息	182
9.7	使用 Windows 注册表	183
9.7.1	注册表键	183
9.7.2	注册表数据类型	185
9.7.3	使用 TRegistry	186
第十章	创建构件	188
10.1	创建新构件	188
10.1.1	New Component 对话框	188
10.1.2	生成 FlashingLabel 构件	189
10.2	编写构件属性和方法	191
10.2.1	属性	191
10.2.2	属性可以直接访问	193
10.2.3	编写构件方法	195

10.3	向 FlashingLabel 构件增加功能	195
10.3.1	添加属性、方法和事件	195
10.3.2	ComponentState 属性	200
10.4	测试构件	200
10.4.1	测试程序设计	200
10.4.2	添加新构件	201
10.5	把新构件加入构件面板中	202
10.5.1	加入构件面板的步骤	202
10.5.2	为新构件定制位图	203
10.6	编写构件的事件	203
10.6.1	事件概述	204
10.6.2	编写构件的自定义事件	204
10.6.3	改写基础类事件	206
10.7	程序清单	207
第十一章	C++ Builder 4 数据库结构	214
11.1	数据库基础	214
11.1.1	数据库概念	215
11.1.2	本地数据库	215
11.1.3	客户机/服务器数据库	215
11.1.4	单层、双层和多层数据库结构	215
11.2	Borland 数据库引擎	216
11.2.1	BDE 驱动器	216
11.2.2	BDE 别名	216
11.2.3	创建 BDE 别名	217
11.2.4	SQL Links	219
11.3	Local InterBase	220
11.4	C++ Builder 数据库构件	220
11.4.1	数据库构件概述	221
11.4.2	TDataSet 类	222
11.4.3	Table 构件	225
11.4.4	Query 构件	232
11.4.5	StoredProc 构件	234
11.4.6	UpdateSQL 构件	235
11.4.7	DataSource 构件	236
11.4.8	Session 构件	236
11.4.9	Database 构件	239
11.4.10	BatchMove 构件	243
11.4.11	TField 类	243
11.5	客户机/服务器数据库构件	245

11.5.1	TClientDataset 构件概述	245
11.5.2	浏览和编辑数据	246
11.5.3	索引	247
11.5.4	计算字段	248
11.5.5	统计值	248
11.5.6	数据包	249
11.5.7	与应用服务器通信	250
11.5.8	在文件中存储数据	251
11.6	C++ Builder 数据构件	252
11.6.1	数据构件的共同属性	252
11.6.2	DBGrid 构件	252
11.6.3	DBNavigator 构件	254
11.6.4	DBText 和 DBEdit 构件	255
11.6.5	DBMemo 和 DBRichEdit 构件	255
11.6.6	DBImage 构件	255
11.6.7	DBListBox 和 DBComboBox 构件	256
11.6.8	DBLookupListBox 和 DBLookupComboBox 构件	257
11.6.9	DBCheckBox 和 DBRadioGroup 构件	257
11.6.10	DBCtrlGrid 构件	258
11.6.11	其它数据构件	260
11.7	数据库应用程序的体系结构	260
11.7.1	设计数据库应用程序	260
11.7.2	数据库应用程序的体系结构	261
11.7.3	Provider	264
11.7.4	与应用服务器连接	267
11.7.5	调用服务器上的接口	269
第十二章 SQL 语言基础		270
12.1	结构化查询语言	270
12.2	用 select 检索数据	271
12.2.1	从表中选择数据	271
12.2.2	where 的使用	274
12.2.3	查询结果的排序	277
12.2.4	检索集合数据	278
12.2.5	连接	278
12.2.6	子查询	279
12.2.7	union	280
12.3	select 和 into 的联合使用	280
12.4	insert 的使用	281
12.4.1	插入指定的值	281

12.4.2 插入多行	281
12.4.3 省略列列表	281
12.4.4 通过存储过程来插入	282
12.5 update 的使用	282
12.6 delete 的使用	283
第十三章 建立数据库应用程序	284
13.1 数据库窗体向导	284
13.1.1 生成简单窗体	284
13.1.2 生成主/细目窗体	288
13.2 手工生成数据库窗体	290
13.3 非图形数据库编程	292
13.3.1 读数据库	292
13.3.2 生成数据库表格	296
13.4 数据模块的使用	298
第十四章 快速报表	301
14.1 报表构件	301
14.1.1 QuickRep 构件	301
14.1.2 QRBand 构件	304
14.1.3 报表设计元素	304
14.2 生成报表	305
14.2.1 生成简单报表	305
14.2.2 生成报表应用程序	307
第十五章 图形编程	309
15.1 图形构件	309
15.1.1 Shape 构件	309
15.1.2 Image 构件	310
15.1.3 PainBox 构件	311
15.2 TCanvas 类	312
15.2.1 两种绘图方法的比较	312
15.2.2 TCanvas 类	313
15.3 图形设备接口	314
15.3.1 画笔、画刷和字体	314
15.3.2 位图与调色板	317
15.3.3 剪取区域	318
15.4 绘图操作	319
15.4.1 绘制位图	319
15.4.2 绘制文本	320
第十六章 多媒体编程	324

16.1	文字效果	324
16.1.1	文字概述	324
16.1.2	TFont 类	325
16.1.3	文字的显示	327
16.1.4	文字动画	329
16.2	图像动画	333
16.3	音乐效果	337
16.3.1	WAV 和 MIDI 文件简介	337
16.3.2	TMediaPlayer 构件	338
16.3.3	播放音乐实例	340
16.4	播放视屏文件	343
16.4.1	视屏文件概述	343
16.4.2	用 C++ Builder 4 播放 DAT	344
第十七章	网络编程	348
17.1	分布式组件对象模型 DCOM	348
17.1.1	DCOM 概述	348
17.1.2	用 Borland C++ Builder 4 进行 DCOM 编程	355
17.2	Internet 编程	365
17.2.1	Borland C++ Builder 提供的 Internet 控件	365
17.2.2	WinSock 编程	366
17.2.3	建立 Web 浏览器	381
17.2.4	使用 FTP 控件	393
第十八章	C++ Builder 4 程序打包	411
18.1	打包准备工作	411
18.1.1	InstallShield Express 概述	411
18.1.2	InstallShield Express 环境	411
18.2	程序打包	412
18.2.1	整理档案	412
18.2.2	打包过程	413

第一章 C++语言简介

C++是一种功能强大的计算机编程语言，自问世以来已受到广大计算机用户的欢迎，并被用于许许多多的编程和开发工作中。C++源于C语言，建立在C语言的基础之上，它不同于C语言的主要特点是带有类。

本章将对C++语言作一些基本介绍，主要涉及如下几方面的内容：

- C++ 语言初步
- C++ 类
- C++ 类库和可视化构件模型

1.1 C++语言初步

我们首先介绍C++语言概览、条件表达式、循环控制、new和delete操作符、指针和引用等有关内容，这些知识适用于所有基于C++语言的编程工具。

1.1.1 C++语言概览

C++语言的一个特点就是允许用户最大限度地利用面向对象编程(OOP)。OOP不仅操作方便，还允许用户创建对象。这些对象既可以在当前程序中使用，也可以在以后的程序中使用。对象只向用户(使用对象的程序员)展示它自身所需要的部分，以便简化其使用，用户可以不必知道其内部的机制。

1. 变量

变量代表存储单元的名字，一旦声明了一个变量就可以用它在存储器中操作数据，已声明但没赋初值的变量包含随机值。下面几行代码用了两个变量，在每行代码的结束后是注释，说明执行该行代码的结果：

```
int x;           //变量 x 声明为整型变量
x=10;           //把值 10 赋给变量 x
int y=15;        //声明变量 x 为整型并初始化其值为 15
x+=y;           //x 和 y 的值相加后赋给 x
x++;            //x 的值加 1
```

变量名可以是混合大小写字母，可以包含数字和下划线(_)，但不能包含空格或其它特殊字符。变量名必须是字符或下划线开始，但一般来说，变量名由下划线开始并不可取，因为编译器常由下划线打头的特殊变量或函数开始。变量名的最大允许长度视编译器而定，如果变量名不超过31个字符，那将是安全的，实际上，超过20个字符的变量名就显得比较长。

2. C++的数据类型

在C++中，使用变量前必须声明变量类型。在C++的一些程序语言中，可以给一个变

量赋任意类型的值，如 BASIC 代码。当用户错误地使用数据类型时会导致系统编译错误，这时，系统会提示用户进行改正。C++ 中的数据类型可以是带符号的也可以是不带符号的。带符号的数据类型可包含正数或负数，而不带符号的数据类型只能包含正数。表 1-1 列出了 C++ 使用的基本数据类型、所需的存储空间以及该数据类型的取值变化范围。

表 1-1 C++ 使用的数据类型(32 位程序)

数据类型	字节大小	取值范围
Char	1	-128 到 128
unsigned char	1	0 到 255
Short	2	-32768 到 32767
unsigned short	2	0 到 65535
Long	4	-2147483648 到 2147483648
unsigned long	4	0 到 4294967295
Int	4	同 long
unsigned int	4	同 unsigned long
Float	4	1.2×10^{-38} 到 3.4×10^{38}
Double	8	2.2×10^{-308} 到 1.8×10^{308}
Bool	1	True 或 false

注意：只有 double 和 float 数据类型使用浮点数，另外的数据类型只处理整型数。尽管可把浮点数赋给整数数据类型，但其小数部分会被抛弃，只有整数部分被赋给了变量。但是，有可能的话，C++ 会在不同的数据类型之间进行转换，如下列代码所示：

```
short result;
long num1=200;
long num2=200;
result=num1*num2;
```

此例试图把两个 long 整数的乘积赋给一个 short 整数。虽然该式混合了两种数据类型，C++ 能够执行转换，但是由于转换的缘故，num1*num2 的值超出了 result 的值域，所以得到的结果不是 40000，而是 -25536。如果 result 定义为 int 数据类型就不会有问题。

在一些情形下，C++ 不能执行转换，就会导致诸如 Cannot convert from X to Y 的编译错误，也会导致一个编译警告：Conversion may lose significant digits。

3. C++ 运算符

C++ 有许多运算符，表 1-2 列出了一些常用的运算符及其意义和例子。运算符用来操作数据，它执行计算、等式验证、赋值、操纵变量及其它大多数程序员永远不会碰到的任务。虽然运算符较多，记住每一个符号比较困难，但是在使用 C++ 的过程中，用户会逐渐掌握所有的运算符的使用。

表 1-2 常用的 C++ 运算符

运算符	说明	例子
数学运算符		
+	加	$X=y+z;$
-	减	$X=y-z;$
*	乘	$X=y*z;$
/	除	$X=y/z;$

续 表

运算符	说明	例子
赋值操作符		
=	赋值	X=10;
+=	赋值并加 1	X+=10;
-=	赋值并减 1	X-=10;
=	赋值并乘	X=10;
/=	赋值并除	X/=10;
&=	赋值并位与	X&=0×02
=	赋值并位或	X =0×02
逻辑操作符		
&&	逻辑与	if(x&&0XFF){...}
	逻辑或	if(x 0XFF){...}
相等操作符		
==	等于	if(x==10){...}
!=	不等于	if(x!=10){...}
>	大于	if(x>10){...}
<	小于	if(x<10){...}
>=	大于等于	if(x>=10){...}
<=	小于等于	if(x<=10){...}
么操作符		
*	间接操作符	int x=*y;
&	地址操作符	int x=&y;
~	位非	x&=~0x02
!	逻辑非	if (! valid){...}
++	增 1 操作符	x++;
--	减 1 操作符	x--;
类和结构操作符		
::	域分别	MyClass::SomeFunction();
->	间接成员关系	MyClass->SomeFunction();
.	直接成员关系	MyClass.SomeFunction();

应该注意的是,有时一个运算符可以作为前增量(++x)或后增量(x++)。前增量表示先增加变量的值,然后再用该变量,而后增量表示先用变量,再增加它的值。还有一些运算符用同一个符号,其意义必须由上下文确定。比如,星号(*)可以用来执行乘法,声明一个指针,或者间接引用一个指针。

4. C++中的函数

函数是与主程序分开的一段代码,在程序中当需要执行特定功能时,就调用这些代码。函数是任何程序语言的重要组成部分,C++也不例外。简单类型的函数不带参数并返回空值(void),其他函数可能带一个或多个参数,也可能返回一个值。

函数的命名规则与变量名的命名规则相同,值得注意的是命名要有一定的意义。函数的结构如图 1-1 所示。

在使用函数前,必须先声明它。函数声名或原型告诉编译器带多少参数,每个参数的数据类型以及函数返回值的数据类型。函数的应用如程序清单 1-1 所示。