



C++Builder 6

总结实例



中国计算机学会 编

本书导读

学习 C++Builder6.0 高级语言的目的是使用最快最好的语言工具开发高效的应用程序，能够完全地支持 C++ 语言的规则，Borland 公司向读者真诚推荐 C++ Builder 6.0 这一最新应用程序开发工具，它具有像 VB 一样易使用的优点，同时又是一个可视化 C++ 语言，是适用于众多使用过 C++ 语言的读者。

应用程序的开发领域是多方面的，本书由浅入深地向读者详细地介绍 C++ Builder 6.0 的使用和 Windows 的编程技术，重点向读者展示使用 C++ Builder 开发应用程序的方法和技巧，并对文件管理和多媒体应用程序的开发也进行了简要的介绍。

第一章简要介绍 C++ Builder 的安装与开发过程、集成开发环境和其所使用的 C++ 语言在 Windows 中的使用，详细介绍 C++ Builder6.0 的工作环境和组成；

第二章向读者介绍一些 C++ Builder 6.0 可视化部件、名空间、运算符、指针、类与对象、集合、函数等，通过这些读者可以学到很多编写实际应用程序时所需要掌握的问题；

第三章主要介绍了窗体和组件，并分别配以例程加以说明；

第四章主要介绍了 C++Builder 6.0 基本编程方法，包括条件分支语句、循环语句的使用、函数和类方法的定义、指针的使用、动态内存的分配等；

第五章介绍了键盘响应和鼠标响应事件在 C++Builder 中的使用；

第六章介绍了一些实现文件、目录和驱动器的操作的编程方法；

第七章介绍了如何在 C++Builder 中撰写一个完整的绘图程序；

第八章介绍了如何以 C++Builder 来处理 Windows 讯息；

第九章介绍了 Object Pascal 语言的特征、结构和概念以及如何在 C++ Builder 中实现以支持 VCL；

第十章介绍了 C++Builder 中包和组件的使用；

第十一章介绍如何编写计划发布到国际市场的应用程序；

第十二至十三章详细地介绍了图形图像编程以及关系数据库的基本概念和简单数据库应用程序的开发方法。

本书追求的目标是使读者从入门到精通 Windows 应用程序的开发，解决中文应用程序开发的疑难，使每一个使用本书的读者都能掌握 C++ Builder 6.0 的使用技巧，从中得到真正的收获和体会。

本书内容由浅入深，语言清晰明朗，适合所有 C++ Builder 初学者和中级读者阅读。

内容简介

C++ Builder 6.0 是 Borland 公司最新推出的功能强大的面向对象、可视化的应用程序开发软件。本书着重介绍了 C++ Builder 6.0 编程的基本知识及在 Windows 应用程序的原理和方法，着重引导用户由浅入深的开发应用程序。内容包括 BCB 语言简介、文本编辑器、图形图像开发、文件管理、多媒体和数据库应用程序的开发等。

本书通过大量的实例，介绍 C++ Builder 6.0 各种控件的应用，解决中文应用程序开发的疑难问题，介绍开发应用程序的技巧。

本书内容条理分明，语言清晰明朗，适合所有 C++ Builder 初学者和中级读者阅读。

目 录

第 1 章 C++Builder 软件开发.....	1
1.1 软件开发概况	2
1.1.1 计算模式的发展	2
1.1.2 程序设计方法的发展	3
1.1.3 软件开发与软件开发人员	4
1.1.4 C++Builder 6.0 的安装	6
1.2 C++ Builder 的集成开发环境	7
1.3 Windows 应用程序实例	9
1.4 集成调试器	12
1.5 开发环境的优化	13
1.5.1 自定义工具栏	13
1.5.2 设置编程环境	14
1.5.3 设置编辑环境	15
1.5.4 设置工程	15
1.6 总结实例	16
1.6.1 使用代码浏览器	16
1.6.2 使用代码编辑器	17
1.6.3 使用帮助系统	18
1.6.4 设置 IDE 桌面	19
第 2 章 C++ Builder 的可视化编程环境	20
2.1 C++ Builder 基本概念	21
2.1.1 基本形式	21
2.1.2 面向对象编程的概念	21
2.2 C++Builder 6.0 快速入门	22
2.2.1 进入 C++Builder 6.0 环境	22
2.2.2 可视化编程环境介绍	23
2.3 C++Builder 6.0 可视化控件的使用	26
2.3.1 文本相关控件	26
2.3.2 按钮和检查框控件	28
2.3.3 分组、分界控件	28
2.4 引用	29
2.4.1 引用变量	29
2.4.2 引用参数传递	30

2.5 名空间.....	31
2.5.1 namespace.....	31
2.5.2 using.....	33
2.6 运算符.....	34
2.6.1 new 与 delete 运算符.....	34
2.6.2 typeid 运算符.....	35
2.7 void 指针.....	36
2.8 函数.....	38
2.8.1 内联函数.....	38
2.8.2 指定函数形参的默认值.....	39
2.8.3 参数个数不定的函数.....	40
2.8.4 模板函数.....	41
2.8.5 重载函数.....	43
2.8.6 指向函数的指针.....	45
2.9 类与对象.....	46
2.9.1 类的封装.....	46
2.9.2 类的派生与继承.....	47
2.9.3 构造函数和析构函数.....	47
2.9.4 对象的引用参数传递.....	50
2.9.5 静态成员变量与静态成员函数.....	52
2.9.6 const 类型成员函数与 mutable.....	54
2.9.7 对象的初始化与初始化行.....	55
2.9.8 拷贝构造函数.....	57
2.9.9 模板类.....	59
2.10 友元函数与友元类.....	61
2.11 运算符重载.....	63
2.12 派生和继承.....	65
2.12.1 派生与继承.....	65
2.12.2 多重、多层派生与继承.....	69
2.12.3 虚基类.....	69
2.13 多态性和虚函数.....	71
2.14 集合.....	73
2.14.1 集合的概念.....	74
2.14.2 集合的操作.....	74
第3章 C++Builder 6.0 窗体和组件.....	78
3.1 窗体和组件.....	79
3.1.1 窗体.....	79
3.1.2 组件.....	82

3.1.3 在窗体中使用组件	83
3.2 文本组件	86
3.3 特殊输入组件	88
3.4 按钮组件	90
3.5 列表组件	92
3.6 分组组件	96
3.7 信息反馈组件	98
3.8 表格显示组件	99
3.9 图形组件	100
3.10 常用组件的用法实例	102
3.11 总结实例	105
3.11.1 使用原则	105
3.11.2 设置动态的背景图	106
3.11.3 Owner 和 Parent 的区别	107
第 4 章 C++Builder 6.0 基本编程方法	109
4.1 条件分支语句	110
4.1.1 布尔类型	110
4.1.2 if 语句	111
4.1.3 switch 语句	113
4.2 循环语句	114
4.2.1 goto 语句	114
4.2.2 for 循环语句	115
4.2.3 While/do.....while 循环语句	116
4.2.4 break/continue 语句	117
4.2.5 使用注意事项	118
4.3 函数和类方法	118
4.3.1 定义	118
4.3.2 内联函数	120
4.3.3 形式参数和实际参数	121
4.3.4 函数和类方法的返回值	122
4.3.5 使用数组作为参数	123
4.4 指针	124
4.4.1 指针的定义	124
4.4.2 指针的运算	126
4.4.3 多级指针	128
4.4.4 指针和数组	129
4.4.5 指针和动态内存分配	132
4.4.6 引用	134

4.4.7 类型定义	137
4.4.8 指针作函数的参数	138
第5章 键盘和鼠标	142
5.1 键盘事件	143
5.2 鼠标事件	144
5.2.1 常用鼠标事件	144
5.2.2 拖曳事件	144
5.3 拖动操作的实现	145
5.4 总结实例	146
5.4.1 检测 Shift、Alt 和 Ctrl 键是否按下	146
5.4.2 屏蔽系统功能键	148
5.4.3 模拟按下键盘上的某个键	149
5.4.4 限制鼠标移动的范围	149
5.4.5 自定义鼠标	150
5.4.6 设置光标闪烁的速度	152
第6章 文件、目录和驱动器	154
6.1 文件	155
6.1.1 C 标准文件类型	155
6.1.2 C++ 流式文件类	156
6.1.3 非缓冲文件	156
6.2 文件对话框组件	157
6.3 文本编辑器编制实例	158
6.4 Win 3.1 相关组件	160
6.5 常用文件管理函数	160
6.5.1 文件操作函数	160
6.5.2 目录操作函数	166
6.5.3 驱动器操作函数	168
6.5.4 文件名操作函数	168
6.6 统计源文件字节数实例	170
6.7 数字有效位转换实例	172
6.8 总结实例	173
6.8.1 获取文件的日期信息	173
6.8.2 检测软盘或光盘是否曾变化	175
6.8.3 检测驱动器容量	176
6.8.4 拷贝整个目录	178
6.8.5 将文件删除到回收站中	179
6.8.6 检测驱动器是否就绪	180

6.8.7 获取应用程序的文件名	181
6.8.8 操作临时文件	182
第 7 章 绘图程序的设计	184
7.1 关于鼠标事件 (Mouse Event)	185
7.2 Windows 内鼠标相关 Message	186
7.3 鼠标事件的处理	186
7.4 OnMouseDown 事件的处理	187
7.5 OnMouseUp 事件的处理	187
7.6 OnMouseMove 事件的处理	188
7.7 鼠标处理的加强	188
7.8 绘图物件的定义	190
7.9 小画家使用的工具列	197
7.10 工具列的事件处理函数	200
第 8 章 Windows 讯息的处理	202
8.1 自定标题棒的制作	204
8.2 自定标题的绘制	206
8.3 旋转文字的输出	207
8.4 程式的改进	209
8.5 在程式中使用材质背景	210
8.6 以材质图案填满画面	211
第 9 章 VCL 的 C++ 语言支持	212
9.1 C++ 和 Object Pascal 对象模型	213
9.1.1 对象本身和实例	213
9.1.2 VCL 类的对象构造	215
9.1.3 在基类构造函数调用虚拟方法	215
9.1.4 对象析构	218
9.1.5 AfterConstruction 和 BeforeDestruction	219
9.1.6 类虚拟函数	220
9.2 Object Pascal 数据类型和语言概念	220
9.2.1 Typedefs	220
9.2.2 支持 Object Pascal 语言的类	221
9.2.3 Object Pascal 语言的 C++ 语言对应	221
9.2.4 动态数组	221
9.2.5 定义不同的类型	224
9.2.6 资源字符串	225
9.2.7 缺省参数	225

9.2.8 运行时类型信息	226
9.2.9 未映射的类型	226
9.2.10 关键字扩展	227
9.2.11 __declspec 关键字扩展	229
第 10 章 包和组件的使用	231
10.1 使用包	232
10.2 运行时包	232
10.2.1 在应用程序中使用运行时包	233
10.2.2 动态装载包	233
10.2.3 运行时包的选择	233
10.2.4 定制包	234
10.3 设计时包	234
10.4 创建和编辑包	235
10.4.1 创建包	235
10.4.2 编辑包	236
10.4.3 包源文件和工程选项文件	236
10.4.4 包的结构	237
10.4.5 编译包	238
10.5 发布包	240
10.5.1 发布使用包的应用程序	240
10.5.2 将包发布给其他开发者	240
10.5.3 包集文件	240
第 11 章 创建国际化的应用程序	242
11.1 国际化和本地化	243
11.1.1 国际化	243
11.1.2 本地化	243
11.2 国际化应用程序	243
11.2.1 应用程序代码	243
11.2.2 设计用户界面	247
11.2.3 分离资源	248
11.2.4 创建资源 DLL	248
11.2.5 使用资源 DLL	249
11.2.6 动态切换资源 DLL	250
11.3 本地化应用程序	250
第 12 章 图形图像编程	251
12.1 常用图形对象及简单应用	252

12.1.1	画布对象 (TCanvas Object)	252
12.1.2	画笔对象 (TPen Object)	255
12.1.3	画刷对象 (TBrush Object)	258
12.1.4	TColor 类型	259
12.2	图形程序的开发	260
12.2.1	响应鼠标事件	260
12.2.2	简单的画圆程序	263
12.2.3	画板程序	268
12.3	动画绘图效果	280
12.4	常用图像对象	283
12.4.1	TGraphics 类	283
12.4.2	TPicture 类	283
12.4.3	TBitmap Object (位图对象)	283
12.4.4	TImage 控件	285
12.5	总结实例	286
12.5.1	图像浏览器制作实例	286
12.5.2	百叶窗效果制作实例	293
12.5.3	积木效果制作实例	294
第 13 章	设计数据库应用程序	296
13.1	使用数据库	297
13.1.1	数据库类型	297
13.1.2	数据库安全	298
13.1.3	事务	299
13.1.4	数据字典	299
13.1.5	引用完整性、存储过程和触发器	300
13.2	数据库结构	300
13.2.1	可扩展性规划	301
13.2.2	一级数据库应用程序	302
13.2.3	二级数据库应用程序	302
13.2.4	多级数据库应用程序	303
13.3	设计用户接口	304
13.3.1	显示单条记录	304
13.3.2	显示多条记录	304
13.3.3	分析数据	305
13.3.4	选择要显示的数据	305
附录	C++Builder 6.0 经验总结	307

第 1 章 C++Builder 软件开发



你 将 会 学 到



1. 软件开发概况
2. C++ Builder 的集成开发环境
3. Windows 应用程序实例
4. 集成调试器
5. 开发环境的优化
6. 总结实例

本章重点介绍了 C++Builder6.0 集成开发环境，以便读者在创建应用程序时能充分利用其强大的功能，从而达到事半功倍的效果。

对于 C++Builder6.0 的初学者来说，可能暂时用不到本章介绍的部分内容，所以在阅读的时候可以跳过去，等有了一些编程经验以后，再回过头来阅读本章，收获将会更多一些。

1.1 软件开发概况

1.1.1 计算模式的发展

计算模式的发展概况如图 1-1 所示，具体过程说明如下：

1. 单机计算模式

在计算机最初被应用的时候，往往是为了加快计算的速度，功能类似于现在通常所说的“计算器”。这时的数据量不大，数据类型也比较单一，所以计算模式采用的大都是单机计算模式。这一阶段的程序员在本机通过编写代码对数据进行管理和操作。形象描述可参考图 1-1 左上角的阶段 1。

2. 单机数据库计算模式

随着数据量的增加、数据复杂度的提高而出现了数据库技术，这时的数据库一般被称为桌面数据库或本地数据库。在这一阶段，已经将数据的管理和数据的计算分开，也就是数据库完成数据的管理功能，而程序员编写一定的代码去对数据进行操作。形象描述可参考图 1-1 右上角的阶段 2。

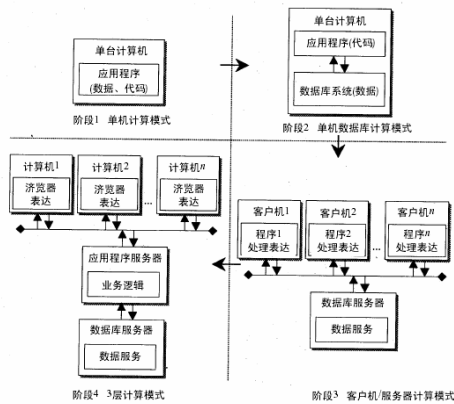


图 1-1 计算模式的发展概况图

目前仍然经常使用的桌面数据库有 Access、Paradox 和 Visual FoxPro 等。

3. 客户机/服务器计算模式（C/S）

随着数据量、数据复杂度和计算量的增加以及用户的需求难度的提高和网络的应用，出现了客户机/服务器计算模式，也就是 Client/Server 模式。这时已经将数据的管理与操作进一步分离：



通常服务器上的数据库系统对数据进行管理,并给网络上的其他计算机提供数据服务;客户机通过网络从数据库服务器上获取数据,实现对数据的处理。由于这时客户机上的程序功能比较齐全,不但完成数据的处理工作,还要实现数据的表达工作,比如人机交互的界面,所以显得程序比较庞大,通常将这一阶段的客户程序称为“胖客户”程序。形象描述可参考图 1-1 右下角的阶段 3。

这一阶段已经可以实现多台计算机间的数据共享,也出现了并行计算,但网络一般局限于企业的内部网 (Intranet)。常用的数据库系统已经在功能上和性能上比桌面数据库有了极大的提高,具体有: InterBase、Oracle 和 SQL Server 等。

4.3 层/ n 层计算模式 ($3 \sim tiers/n \sim tiers$)

随着 Internet 技术的普及和用户需求的进一步提高,出现了目前最为流行的 3 层计算模式,也就是将应用程序分成 3 个明确的层:显示层、逻辑层和数据层(形象描述可参考图 1-1 左下角的阶段 4)。这时不仅“数据”与“处理”是明显分开的,而且“处理”与“表达”也明显地分开了。

这里的“数据处理”也就是通常所说的“业务规则”,已经被单独封装在“应用程序服务器”中了。而表达层指的是用户接口,比如 Windows 操作系统中的图形用户界面等。通常使用的“浏览器”就是一个标准通用的表达层软件。

在 3 层计算模式中,将“表达”、“业务规则”和“数据服务”分离开来将带来非常多的好处。

比如,在 Internet 环境中实现电子商务的时候,将用户和商品的信息保存在公司的数据库服务器中,而在公司的应用程序服务器中按一定的规则实现对这些信息的处理:接收来自浏览器的请求,然后访问公司的数据库服务器,接着对获取的信息进行处理,最后将结果返回给浏览器。

如果公司的业务规则随着时间发生了变化,我们只要改变中间应用程序服务器上的程序就可以了,而无需修改数据库服务器上的数据逻辑和大量存在于 Internet 空间的浏览器。这时,应用程序要解决的问题对于数据来源和结果表达是不关心的,于是它将随着时间而保持良好的可扩展性。

当软件需求的复杂性不断增加后,有时将把一个特殊的层分成两个或更多的层,比如将应用程序服务器层分成 WWW 服务器层和业务操作层,这就导致了多层或 n 层体系,而实际上 n 层的基础还是 3 层体系。

1.1.2 程序设计方法的发展

随着计算机技术的发展,计算模式在不断地发展,同时,程序的设计方法也在不断地发展。

具体来讲,主要可以分为下面的三个不同的程序设计方法:

1. 结构化程序设计 (Structured Programming)

结构化程序设计采用的是由上而下的程序设计方法:先从大的方面考虑问题,然后按一定的顺序依次解决,接着对各个方面逐步细化。结构化程序设计的代表语言是 C 语言和 C++ 语言等。

2. 面向对象程序设计 (Object-Oriented Programming)

面向对象程序设计实现了对现实世界的反映,引入了“类”与“对象”的概念。面向对象程序设计的代表语言是 C++ 语言和对象 Pascal 语言。

3. 组件对象模型程序设计 (COM /CORBA)

组件对象模型程序设计是在面向对象程序设计技术的基础上发展起来的,它可以实现软件的功能模块化、编程语言的无关性、操作系统的无关性等,极大地提高了代码的可重用性、软件的可扩展性等。

同 COM 技术并行的另一个技术就是 CORBA (Common Object Request Broker Architecture) 技术,即公共对象请求代理体系结构,发展前景也非常广阔。

CORBA 的目的是为了简化开发分布式应用系统的复杂性以及减少需要花费的成本。CORBA 使用了面向对象和组件的设计结构,允许软件对象在不同的操作系统平台和应用程序之中重复使用。

上面的三个程序设计的方法并不是一个取代另一个的关系,而是包含与被包含的关系:在进行面向对象程序设计的时候,需要用到结构化程序设计;而进行组件对象模型程序设计的时候,也一定要用到面向对象程序设计和结构化程序设计的技术。

目前国内的等级考试、学校教学等只涉及到结构化程序设计,而一般的软件系统在开发过程中使用的则是面向对象程序设计的方法,组件对象模型程序设计的方法也正在被广泛地应用,比如 Word、Access、IE 等都采用了 COM 技术,而微软公司的 Windows 2000 实际上也就是一个 COM 的大集合。

1.1.3 软件开发与软件开发人员

现在,计算机技术已经被纳入信息技术 (IT: Information Technology) 的领域。在这一领域,需求决定了技术,技术领导着方向。

下面针对目前的几种主要类型的计算机软件说明软件开发的情况:

1. 通用软件的开发

通用软件的种类很多,应用范围非常广。比如字处理软件、词典软件、杀毒软件和游戏软件等。这类软件针对一些具体应用编写,总体上开发技术不难,但可能会含有一些关键技术。

C++ Builder、Delphi、Visual C++ 和 Visual Basic 等软件开发工具都可以很好地开发通用软件。

2. 控制软件的开发

控制软件一般和相关硬件紧密接合,实现对具体硬件的实时控制。比如,可以实现精密加工的数控机床、一些家用电器和便携式仪器等,采用了中央处理器芯片进行控制。在大中型设备上,普遍采用微机进行控制;在小型设备和微型设备上,一般采用单片机(如 8096/8098 系列)进行控制。

以前,开发实时控制的程序往往采用汇编语言。现在,也可以使用高级语言进行编程,或者使用汇编语言和高级语言进行混合编程。C++ Builder、Delphi 等开发工具都提供了这些功能。



从整体上看,目前国内软件盗版行为非常猖獗。不过,在控制软件方面,由于软硬件结合的紧密性和技术的复杂性,盗版情况就相对比较少。

3. 科学计算软件的开发

计算机技术最初应用在科学计算领域,目前这一领域使用计算机的比例还很大,而且越来越依靠计算机技术的发展。

C++ Builder 非常适合用来开发科学计算方面的软件,其代码执行效率高,开发周期短,支持并行运算和分布式计算。如果您的工作主要在科学计算方面,也可以使用一些专门面向科学计算的软件,如 Mathematica、MatLab 等。这些软件的科学计算功能方面非常强大,开发周期更短,只是灵活性不如 C++ Builder、Delphi 和 Visual C++ 等。

4. 多媒体软件的开发

市场上的多媒体软件越来越多,同时,多媒体与数据库、网络也将紧密地结合在一起。通过数据库实现对多媒体信息的管理、查询,即多媒体数据库的需求也越来越大。互联网上的静态主页正在被动态主页所替代,其中很大一部分就是多媒体技术和数据库技术的应用。

C++ Builder 在多媒体方面具有强大的功能,支持 OpenGL、DirectX 等技术。当然,开发普通的多媒体软件或功能模块可以采用一些专业多媒体开发软件,如 Director、Authorware、Flash Tool Book 等软件。

5. 数据库软件的开发

使用数据库可以有效地管理大量复杂的信息。C++ Builder 支持多种数据库访问接口,如 ODBC、DAO、ADO 和 OLE DB 等,对有些数据库接口还提供了 MFC 支持。另外,C++ Builder 在开发数据库软件的过程中,提供了一个对数据库管理、操作的集成开发环境。

6. 网络软件的开发

Internet/Intranet 技术的发展非常迅猛,尤其和数据库技术结合以后,具有更强大的生命力。

比如,电子商务(ecommerce)可以通过互联网与顾客和商业伙伴进行交易。管理信息系统(MIS)可以有效地实现一个公司、学校或科研单位内部的办公自动化和信息共享。

使用 C++ Builder 在开发网络软件时可以采用的技术很多,如 WinSocket 编程、WinNet 编程和 ActiveX、COM/DCOM、MIDAS 和 CORBA 编程等。

7. 其他软件的开发

其他软件的开发主要有:操作系统的开发、数据库管理系统的开发、硬件设备驱动程序的开发等。在这些领域,C++ Builder 将是一个非常好的工具。

应该提醒大家的是,近几年 Linux 操作系统的广泛应用给软件开发人员提供了许多机遇和挑战。在 Linux 操作系统中,也可以方便地使用 C++ 语言进行软件开发。

总的来说,计算机应用已经渗透到日常生活中的各个方面,计算机软件适用的范围也越来越广。世界范围内存在的软件开发人员匮乏的现状,为我们从事软件开发提供了广阔的空间。

软件开发人员或即将从事软件开发的人员可能会常常听到:现在软件开发领域最热门的是数据库和网络,找工作也最容易。的确,网络和数据库,包括这两个领域的结合部分,是目前软件开发的重点,也是本书的重点。但是具体内容又包括哪些呢?发展方向又如何

呢？

许多从事软件业或将要从事软件业的人在就业前，可能会从将来使用哪一个软件开发工具来考虑，并确定一个短期的学习方向。这种以软件开发工具来进行分类的方法比较普遍，比如在许多招聘广告上会明确说明：需要一个 C++ Builder 程序员或需要一个 C++ 程序员，但是这种分类方法只是表面现象。当一个项目确定下来后，会根据项目实现的目的和一些特殊要求来进行功能划分，然后针对不同的功能划分来选择相应的开发工具和计算机语言，在确定使用的开发工具和计算机语言的时候，开发组成员的素质是需要考虑的重要因素之一。有时，目前的人手可能满足不了需求，就可以采用招聘的方式补充力量。

我们可以从软件实现的功能来考虑软件开发的分类，比如我们要完成一个普通的 3 层计算模式的软件项目，对于底层的数据库服务器，可以选择 SQL Server、Oracle 或 InterBase 等；对于中间层的应用程序服务器，可以使用 C++ Builder、Delphi 等使用支持指针、COM 技术而且运行效率比较高的开发工具来编写软件；对于表达层的软件可以采用最通用的浏览器，具体实现可以使用效率比较低的解释语言，如 HTML、VBScript、JavaScript 等。可以看到，在实现一个具体目的的时候，可以有多种不同的方法。这时，开发工具并不是最重要的，因为采用的具体技术对这些开发工具都是通用的。所以，我们在学习 C++ Builder 的时候，不但要学习如何使用 C++ Builder 进行软件开发，而且要对其中涉及的具体技术（如 Windows SDK 编程、COM 等）进行深入了解。

对于软件开发人员，如果参加一个具体的软件工程，技术水平可以很快地提高。如果没有现成的工程可以做，也应该善于“无中生有”，自己去寻找一些“工程”来做。当今的世界商机无限，只要认真地去对待，脚踏实地地干，一定会有所收获。目前国内的软件开发人员发展的轨迹通常是：程序员→高级程序员→系统分析员→项目负责人→部门经理。这几个阶段有重复的部分，有时也可以跳过其中的某个阶段。一般来讲，作为公司的高层领导一定要懂技术，最好是具有技术出身的背景；而对于技术人员，比如高级程序员，也不要精力全部集中在技术上，也应该了解一些管理、协调和工程方面的问题。对于软件开发人员，如果将编程等看作一种乐趣，而不是一个谋生的手段，这时达到的将是一个更高的境界，而国内则很少有人愿意一辈子去做编程的工作。不过，值得高兴的是，由于 COM 技术的发展，对于一些从事管理的人员，他们也可以花较少的精力而比较容易地掌握当前流行的编程技术（如 ADO），编写出像模像样的软件，这也是 COM 技术的一个优点。

1.1.4 C++Builder 6.0 的安装

Borland C++ Builder6.0 的安装是很简便的。只需在 C++ Builder6.0 的安装盘中找到 install.exe 文件，并运行之，它就会一步步做你安装的向导。如图 1-2 所示。

首先选择安装 C++ Builder6.0，其它选项您可以选择安装也可以选择安装不安装。接下来您需要接受安装的协议，点击 next 之后，安装程序将请您输入随盘所带的序列号。然后安装程序将让您选择安装类型，您可以选择典型安装(typical)、完全安装(full)、自定义安装(custom)、紧缩安装(compact)。如果您对 C++Builder 的各部分组成非常熟悉，建议您选择自定义安装，如果您不是太熟悉，您可以选择典型安装或完全安装。接下来安装文件将让您选择安装的目录。完全安装大约需要 900 兆的空间。初学者尽量不要改变安装的目



录，否则可能导致使用中出现问题。接下来安装程序将向您的硬盘上复制所需的文件。并同时向您简单介绍 C++ Builder6.0 的功能与使用。重新启动计算机后，您的 C++ Builder6.0 就安装好了。



图 1-2 选择安装 C++ Builder6.0

1.2 C++ Builder 的集成开发环境

C++ Builder 是 Inprise 公司推出的基于 C++语言的、类似于 Delphi 的可视化集成开发工具,同时它又是一个 RAD。利用 C++ Builder 编程,可以快速、高效地开发出基于 Windows 环境的各类程序，尤其在数据库和网络方面， C++ Builder 更是一个十分理想的软件开发平台。

通常，C++ Builder 6.0 集成开发环境的主界面(如图 1-3)包括下面几个部分：

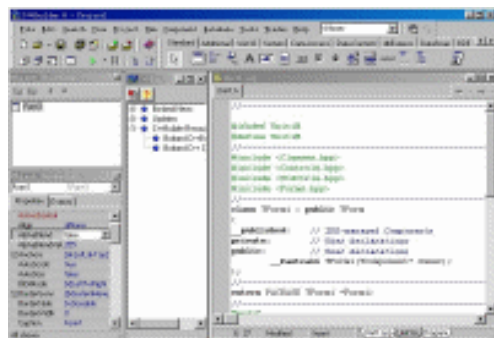


图 1-3 C++ Builder6.0 集成开发环境的主界面

1. 标题栏

标题栏中显示了当前工程的工程名。在图 1-3 中，标题栏位于最上部。

2. 主菜单

通过主菜单可以实现集成开发环境中的绝大部分命令，建议读者在适当的时候逐个地浏览，使用一下主菜单中所有的命令。

另外，在程序开发的过程中，读者可以在不同的情况下，在界面的不同部位点击鼠标右键，看看是否有一些右键弹出菜单出现，通过这些弹出菜单能实现什么新的功能或快捷操作。

C++ Builder 在主菜单栏中将“Component”和“Database”作为独立的菜单项罗列出