

C++ Builder 5

编程实例与技巧

朱时银 马承志 等 编著
杨 飞 王 华



机械工业出版社
China Machine Press

软件开发技术丛书

C++Builder 5 编程 实例与技巧

朱时银 马承志 杨 飞 王 华 等编著



机械工业出版社
China Machine Press

C++Builder 是由Inprise公司推出的面向对象的可视化软件开发工具。C++Builder 5 是 C++Builder 的最新版本。

本书共分六个部分, 主要介绍了C++Builder 5集成开发环境、C++语言关键技术、组件、通用编程技术、动态链接库、多线程、多媒体编程、数据库编程、网络编程和Windows API编程等。适用于C++Builder 5的各级用户。

版权所有, 侵权必究。

JS476/08

图书在版编目(CIP)数据

C++Builder 5编程实例与技巧 / 朱时银等编著. - 北京: 机械工业出版社, 2001.2
(软件开发技术丛书)

ISBN 7-111-07239-1

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

中国版本图书馆CIP数据核字(2000)第55456号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 赵阿玲 金 锋

北京昌平第二印刷厂印刷 · 新华书店北京发行所发行

2001年2月第1版第1次印刷

787mm × 1092mm 1/16 · 42.75印张

印数: 0 001-5 000册

定价: 79.00元 (附光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

前 言

2000年上半年, Inprise公司(原Borland公司)推出了C++Builder的最新版本: C++Builder 5。C++Builder 5继承了以前版本的优点, 并在此基础上增添了许多新的功能, 比如在ADO、数据模块设计、IBX、MIDAS、CORBA、程序调试、VCL、窗体框架和桌面定义等方面都增强了许多, 可以极大地提高软件系统的设计、管理和开发的速度。此外, 由于C++Builder在学习过程中很容易上手, 而且功能也非常强大, 尤其在界面设计、Windows编程、数据库编程和网络编程等方面更具独到之处, 是其他一些软件开发工具无法比拟的。

本书分六个部分, 共25章, 主要介绍了C++Builder集成开发环境、C++语言关键技术、组件、通用编程技术、动态链接库、多线程、多媒体编程、数据库编程、网络编程和Windows API编程等。

参加本书编写工作的主要有: 朱时银、马承志、杨飞和王华等。此外, 梁志刚、张延鑫、秦宇海、陈学林、柳海军、叶爱亮、石玉华、倪凤月、詹武平、武少青和郑再军等广泛地进行了资料收集、例程验证, 并做了细致的校对。在本书的编写过程中, 曹凌云、祁立学、黄家贵、王成、赵雷鸣、刘存根、马兴义和李刚等也提供了许多帮助。尤其感谢张世芳每天打来的电话。机械工业出版社华章公司在本书的出版过程中给予了大力的支持与帮助, 在此一并表示衷心的感谢。

由于作者水平有限, 书中难免存在错误和疏漏之处, 恳请读者批评、指正。

编著者
2000年11月

导 读

本书特色

本书具有下面的一些特色：

(1) 例程精练、实用、完整，易于查找

本书的例程丰富，短小精练。在每章的前面部分，以一定的篇幅对基本理论、知识要点等进行论述；然后以经典的例程加以说明；最后对例程进行分析和总结。

书中的例程完整，并全部按照章号进行了编号。在随书的光盘中有这些例程的源代码和编译、链接后的可执行文件。读者可以使用随书光盘中的“复制”程序(可执行文件为D5Samples.exe)将例程复制到硬盘的指定位置，也可以通过资源管理器进行复制。

另外，对于每个例程都指定了若干个关键字，可以根据关键字查找相关知识点的例程。

(2) 编程技巧丰富

作者将自己在开发程序过程中总结、收集的经验和教训、一些读者在编程过程中遇到的问题等融合在本书中，并且将这些编程技巧尽量独立，便于读者快速查找。

(3) 内容广泛，兼容性强

本书几乎涉及了C++Builder编程的所有领域，不仅适用于C++Builder 5，而且也适用于C++Builder 4和C++Builder 3等以前的版本。

(4) 体现了面向对象的技术和软件工程化的思想

本书从总体上体现了面向对象程序设计的方法，同时将软件工程化的思想融于其中。

内容编排

本书主要分为下面的六个部分：

第一部分 C++Builder编程基础

C++Builder编程基础包括：C++Builder集成开发环境的介绍和C++语言的关键技术。

第二部分 C++Builder编程通用技术

C++Builder编程核心技能包括：窗体和组件、菜单、工具栏和状态栏、键盘和鼠标、文件、目录和驱动器以及打印等。

第三部分 C++Builder高级编程技术

C++Builder 高级编程技术包括：数据共享、操作注册表、帮助系统的制作、安装与卸载、多线程应用程序、动态链接库和图形与多媒体等技术。

第四部分 数据库编程

数据库编程部分包括：关系数据库、数据库组件、InterBase服务器、OLE DB/ADO和Web与数据库等。

第五部分 组件对象模型(COM)

组件对象模型(COM)包括: COM和OLE自动化等内容。

第六部分 分布式程序设计

分布式程序设计包括: DCOM、MIDAS、ActiveForm、CORBA与分布式应用程序, 以及其他的一些编程方法与技巧。

阅读建议

对于C++Builder的初学者, 可以按顺序进行阅读。如果读者对C++语言有一定的了解, 也可以直接从第二部分开始, 待使用C++Builder编程有了一定的经验后, 再回过头来看看第一部分的内容。

对于C++Builder的中级用户, 可以按章节顺序阅读、上机, 也可以直接挑选感兴趣的章节学习。值得注意的是, 第一部分的内容对使用C++语言进行软件开发非常有用, 一定要掌握好。

对于C++Builder的高级用户, 可以直接挑选感兴趣的章节阅读。

目 录

前言
导读

第一部分 C++Builder编程基础

第1章 使用C++Builder进行软件开发	1
1.1 软件开发概况	1
1.1.1 计算模式的发展概况	1
1.1.2 程序设计方法的发展概况	3
1.1.3 软件开发与软件开发人员	4
1.2 集成开发环境简介	6
1.3 例程：第一个Windows应用程序	8
1.4 集成调试器	10
1.5 开发环境的优化	12
1.5.1 自定义工具栏	12
1.5.2 编程环境设置	13
1.5.3 编辑环境设置	14
1.5.4 工程设置	15
1.6 方法与技巧	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	17
1.6.4 设置IDE桌面	18
第2章 C++语言的关键技术	19
2.1 普通注释与特殊格式的注释	19
2.2 const与volatile的用法	20
2.2.1 const	20
2.2.2 volatile	22
2.3 sizeof与#pragma pack的用法	22
2.4 引用	24
2.4.1 引用变量	24
2.4.2 引用参数传递	25
2.5 名空间	26
2.5.1 namespace	27

2.5.2 using	28
2.6 new与delete运算符	30
2.7 void指针	31
2.8 typeid运算符	32
2.9 函数	33
2.9.1 内联函数	34
2.9.2 指定函数形参的默认值	34
2.9.3 参数个数不定的函数	35
2.9.4 模板函数	37
2.9.5 重载函数	39
2.9.6 指向函数的指针	40
2.10 类与对象	41
2.10.1 类的封装	41
2.10.2 类的派生与继承	42
2.10.3 构造函数和析构函数	42
2.10.4 对象的引用参数传递	45
2.10.5 静态成员变量与静态成员函数	47
2.10.6 const类型成员函数与mutable	48
2.10.7 对象的初始化与初始化行	50
2.10.8 拷贝构造函数	52
2.10.9 模板类	54
2.11 友元函数与友元类	56
2.12 运算符重载	57
2.13 派生和继承	59
2.13.1 派生与继承	59
2.13.2 多重、多层派生与继承	63
2.13.3 虚基类	64
2.14 多态性和虚函数	65
2.15 集合	67
2.15.1 集合的基本概念	68
2.15.2 集合的操作	68

2.16 异常处理	72
2.16.1 C++的异常处理	72
2.16.2 BCB中的异常类	75

第二部分 C++Builder编程通用技术

第3章 窗体和组件	83
3.1 窗体和组件	83
3.1.1 窗体	84
3.1.2 组件	86
3.1.3 在窗体中使用组件	87
3.2 文本组件	90
3.3 特殊输入组件	92
3.4 按钮及其分类组件	94
3.5 列表组件	95
3.6 分组组件	98
3.7 信息反馈组件	100
3.8 表格显示组件	101
3.9 图形显示组件	102
3.10 开发MDI应用程序	104
3.11 例程：常用组件的用法	106
3.11.1 例程S03B	106
3.11.2 例程S03C	109
3.11.3 例程S03D	112
3.11.4 例程S03E	113
3.12 方法与技巧	118
3.12.1 窗体、组件的使用原则	118
3.12.2 使用信息对话框	119
3.12.3 使用组件的Hint属性	122
3.12.4 使用Edit组件的IME属性	123
3.12.5 将系统字体添加到ComboBox 组件中	124
3.12.6 在RichEdit组件中存取文件	124
3.12.7 设置模态对话框的返回值	125
3.12.8 使用事件处理过程中的Sender 参数	126
3.12.9 为控件生成多行提示信息	126
3.12.10 生成非矩形窗口	126

3.12.11 移动无标题栏窗口	129
3.12.12 制作动态字幕	130
3.12.13 在窗体上动态地设置背景 画面	131
3.12.14 Owner和Parent的区别	132
第4章 菜单、工具栏和状态栏	134
4.1 菜单	134
4.1.1 主菜单	135
4.1.2 鼠标右键弹出式菜单	136
4.1.3 使用菜单模板	136
4.2 工具栏	137
4.2.1 ToolBar	137
4.2.2 CoolBar	137
4.2.3 ControlBar	138
4.3 状态栏	138
4.4 例程S04A	139
4.5 方法与技巧	144
4.5.1 动态管理菜单	144
4.5.2 定制系统菜单	148
4.5.3 制作图形菜单项	149
4.5.4 在状态栏中添加进度条等其他 组件	152
4.5.5 MDI应用程序中的菜单融合	154
4.5.6 制作可四处拖放的工具栏	158
4.5.7 将菜单项移到菜单栏的最右边	158
4.5.8 运行时控件的移动	159
第5章 键盘和鼠标	161
5.1 键盘事件	161
5.2 鼠标事件	162
5.2.1 常用鼠标事件	162
5.2.2 拖曳事件	162
5.3 例程S05A	163
5.4 方法与技巧	164
5.4.1 检测Shift、Alt和Ctrl键是否按下	164
5.4.2 屏蔽系统功能键	166
5.4.3 模拟按下键盘上的某个键	166
5.4.4 限制鼠标移动的范围	167

5.4.5 自定义鼠标	168	7.6 例程S07B.....	216
5.4.6 设置光标闪烁的速度	169	7.7 方法与技巧	218
5.4.7 不同程序间的拖曳操作	171	7.7.1 获取默认打印机的信息	218
第6章 文件、目录和驱动器	173	7.7.2 获取打印机队列的状态信息	219
6.1 文件	173	7.7.3 打印窗体	221
6.1.1 C标准文件类型	173		
6.1.2 C++流式文件类	174	第三部分 C++Builder高级编程技术	
6.1.3 非缓冲文件	174	第8章 数据共享	223
6.2 文件对话框组件	175	8.1 使用剪贴板	223
6.3 Win3.1相关组件	176	8.1.1 组件对剪贴板操作的支持	223
6.4 常用文件管理函数	176	8.1.2 使用剪贴板类	225
6.4.1 文件操作常用函数	176	8.1.3 操作图形	227
6.4.2 目录操作常用函数	182	8.1.4 操作组件	229
6.4.3 驱动器操作常用函数	184	8.1.5 操作定制格式的数据	230
6.4.4 文件名操作常用函数	184	8.2 动态数据交换(DDE)	231
6.5 例程S06A	186	8.2.1 例程S08F	232
6.6 例程S06B.....	188	8.2.2 操作程序管理器	235
6.7 例程S06C.....	190	8.3 内存映射文件	237
6.8 例程S06D	191	8.3.1 Windows 95/NT内存管理	237
6.9 方法与技巧	193	8.3.2 内存映射文件	239
6.9.1 获取驱动器类型信息	193	8.3.3 映射文件的共享方式	240
6.9.2 操作INI文件.....	194	第9章 操作注册表	246
6.9.3 获取文件的日期信息	196	9.1 TRegistry类	247
6.9.4 检测软盘或光盘是否有过变化	198	9.2 例程S09A	247
6.9.5 检测驱动器容量	199	9.3 与注册表相关的API函数	251
6.9.6 拷贝整个目录	200	9.4 TRegIniFile组件	256
6.9.7 将文件删除到回收站中	202	第10章 帮助系统的制作	258
6.9.8 检测驱动器是否就绪	203	10.1 RTF文件	258
6.9.9 获取应用程序的文件名	203	10.2 HPJ文件与HLP文件	261
6.9.10 操作临时文件	204	10.3 启动帮助	266
第7章 打印	206	第11章 安装与卸载	269
7.1 一些简单的打印输出	206	第12章 多线程应用程序	280
7.2 TPrinter对象和TScreen对象	207	12.1 进程与线程	280
7.3 TPrintDialog、TPrinterSetupDialog 组件	208	12.2 TThread类	281
7.4 打印操作常用API函数	211	12.3 线程的同步	283
7.5 例程S07A	213	12.4 线程的优先级	284
		12.5 例程S12A	284

第13章 动态链接库	291
13.1 DLL基本理论	291
13.1.1 导入与导出匹配	292
13.1.2 隐式链接和显式链接	293
13.1.3 符号链接和序号链接	293
13.1.4 客户程序如何查找DLL	293
13.2 创建动态链接库	294
13.3 使用动态链接库	296
13.4 DLL中包含VCL组件	298
13.5 方法与技巧	301
13.5.1 如何调试动态链接库	301
13.5.2 在DLL中使用MessageBox 代替ShowMessage	301
13.5.3 如何动态地装入DLL	302
13.5.4 如何定制不同语言版本的应用 程序	303
第14章 图形与多媒体	305
14.1 图形与图像	305
14.1.1 图形对象	306
14.1.2 图形类	306
14.1.3 图形对象、类和组件之间的关系	307
14.1.4 画布类	308
14.1.5 橡皮条例程 (S14A)	310
14.2 多媒体	316
14.2.1 概述	316
14.2.2 多媒体应用领域	317
14.2.3 多媒体软件开发工具	318
14.3 图像格式的处理	320
14.3.1 位图	320
14.3.2 JPEG文件	324
14.4 特殊显示效果	326
14.5 动画	334
14.5.1 文字动画	334
14.5.2 图像动画	334
14.6 音频、视频	342
14.7 方法与技巧	346

14.7.1 使用MessageBeep与Beep	346
14.7.2 使用PlaySound	347
14.7.3 使用mciSendString	349

第四部分 数据库编程

第15章 关系数据库	353
15.1 数据库技术的发展	353
15.1.1 数据管理的发展	353
15.1.2 数据库技术的发展	353
15.2 关系数据库基础	354
15.2.1 表	354
15.2.2 字段	355
15.2.3 数据类型	355
15.2.4 记录	356
15.2.5 关键字与索引	356
15.2.6 完整性	357
15.2.7 Paradox数据库的关键字、索引 和引用完整性规则	358
15.2.8 存储过程	358
15.2.9 触发器	359
15.2.10 结构化查询语言(SQL)	359
15.3 数据库设计	368
15.3.1 数据库规范化	368
15.3.2 设计工具	369
15.4 C++ Builder提供的数据库辅助工具	369
15.4.1 SQL Explorer	370
15.4.2 Database Desktop	372
15.4.3 BDE Administrator	374
第16章 数据库组件	377
16.1 设计数据库应用程序	377
16.1.1 使用数据库	377
16.1.2 数据库应用程序结构	379
16.1.3 设计用户界面	382
16.2 常用数据库组件	384
16.2.1 例程S16A	385
16.2.2 数据访问组件	393
16.2.3 数据控制组件	402

16.2.4 表的连接	403	17.4.2 触发器	453
16.2.5 计算字段	407	17.5 本地缓存	454
16.2.6 查找字段	409	第18章 OLE DB/ADO	461
16.2.7 DBGrid组件	411	18.1 数据库客户技术概述	461
16.3 查询记录	412	18.1.1 ODBC	461
16.3.1 使用GotoKey	413	18.1.2 DAO	461
16.3.2 使用FindKey	414	18.2 OLE DB/ADO原理	462
16.3.3 使用模糊查询	415	18.3 ADO组件	462
16.3.4 使用Locate	416	18.3.1 连接到一个ADO数据存储	463
16.4 修改记录	417	18.3.2 使用ADO数据集组件	468
16.5 插入和删除记录	418	18.3.3 使用ADO命令组件	475
16.6 数据模块	420	18.3.4 例程S18A	476
16.7 数据字典查看器	422	第19章 Web与数据库	481
16.8 事务	425	19.1 数据库与Internet	481
16.9 处理数据库异常	431	19.1.1 Web浏览器与Web服务器	481
16.10 报表组件	436	19.1.2 Web文档	482
16.10.1 QuickRep组件	438	19.1.3 统一资源定位符URL	482
16.10.2 Band组件	439	19.1.4 CGI概况	483
16.10.3 创建一个简单的报表(例程 S16P)	440	19.1.5 ISAPI/NSAPI概况	483
16.10.4 QuickReport向导	443	19.2 CGI编程	484
第17章 InterBase 服务器	445	19.2.1 普通CGI编程	484
17.1 设置InterBase	445	19.2.2 CGI数据库编程	489
17.1.1 设置本地InterBase	445	19.2.3 普通ISAPI编程	489
17.1.2 设置InterBase别名	446	19.3 C++Builder对CGI与ISAPI/NSAPI的 支持	489
17.2 客户/服务器的概念	447	19.3.1 TWebModule	490
17.2.1 客户	447	19.3.2 TWebResponse	491
17.2.2 InterBase 客户例程库	447	19.3.3 TWebRequest	491
17.2.3 InterBase服务器	447	19.4 创建Web应用程序	492
17.2.4 Borland数据库引擎	448	19.4.1 CGI类型服务器程序	492
17.3 数据库的管理和维护	448	19.4.2 ISAPI/NSAPI类型服务器程序	494
17.3.1 创建数据库	448	19.4.3 表单	500
17.3.2 创建表	450	19.4.4 把信息保存到数据库	502
17.3.3 数据库安全	451	19.5 HTML生成器组件	506
17.3.4 备份与恢复	451	19.5.1 PageProducer组件	506
17.4 服务器端程序	453	19.5.2 QueryTableProducer组件	508
17.4.1 存储过程	453	19.5.3 TDataSetTableProducer组件	510

19.5.4 DataSetPageProducer组件	512
------------------------------------	-----

第五部分 组件对象模型 (COM)

第20章 COM理论与应用	515
20.1 COM基础理论	515
20.1.1 软件组件的特点	516
20.1.2 DLL的局限性	516
20.1.3 COM的特性	517
20.2 COM对象和接口	517
20.2.1 COM对象	517
20.2.2 COM接口	518
20.2.3 C++抽象基类	521
20.2.4 自定义模拟接口	521
20.2.5 IUnknown接口	524
20.3 COM客户与服务器	524
20.3.1 COM客户	524
20.3.2 COM服务器	524
20.3.3 类厂	526
20.3.4 COM服务器的定位	526
20.4 创建COM应用程序	529
20.4.1 类型库	529
20.4.2 类型库编辑器	529
20.4.3 创建COM服务器程序	534
20.4.4 创建COM客户程序	539
第21章 OLE自动化	545
21.1 自动化概述	545
21.1.1 派发接口	545
21.1.2 双接口	546
21.1.3 早期绑定和后期绑定	547
21.1.4 Variant类型	547
21.2 创建OLE自动化服务器	553
21.3 创建OLE客户程序	555
21.4 完成Word和Excel自动化	557
21.4.1 添加/删除服务器组件	557
21.4.2 启动Word和Excel	560
21.4.3 Word和Excel自动化对象	561
21.5 复合文档	566

21.5.1 为没有类型库的服务器创建 客户应用程序	566
21.5.2 复合文档	567

第六部分 分布式程序设计

第22章 MIDAS和DCOM	577
22.1 分布式系统	577
22.1.1 分布式系统的发展	577
22.1.2 分布式系统使用的技术	579
22.1.3 C++Builder 能开发的分布式 结构	579
22.2 MIDAS	580
22.2.1 MIDAS组件	580
22.2.2 MIDAS应用系统结构	581
22.3 DCOM	582
22.4 开发多层应用系统	582
22.4.1 创建应用程序服务器	583
22.4.2 创建客户应用程序	586
22.4.3 更新数据	588
22.4.4 查询数据	591
22.4.5 参数传递	593
22.4.6 主从约束	595
22.4.7 离线处理	598
22.5 DCOM和多层应用系统的配置	599
22.5.1 客户端应用程序的分发	600
22.5.2 服务器端应用程序的分发	601
第23章 ActiveForm	602
23.1 ActiveX控件	602
23.2 ActiveForm概述	602
23.2.1 如何创建ActiveForm组件	603
23.2.2 注册ActiveX控件	603
23.2.3 在网络上配置ActiveX控件	604
23.2.4 开发在IE中使用的ActiveForm	605
23.3 浏览器中的数据库ActiveForm	609
23.4 技巧: 将Delphi的应用程序 转换成ActiveForm	612
第24章 CORBA与分布式应用程序	613

24.1 CORBA结构	613	25.7 存取计算机名称	647
24.1.1 存根和构架	614	25.8 获取用户注册信息	648
24.1.2 聪明代理	614	25.9 关闭Windows	649
24.1.3 VisiBroker CORBA系统的运行 流程	615	25.10 启动控制面板	649
24.2 编写CORBA应用程序	615	25.11 启动屏幕保护	652
24.2.1 编写CORBA应用程序服务器	616	25.12 使窗体标题栏闪烁	652
24.2.2 编写CORBA客户程序	625	25.13 使窗口背景颜色渐变	653
24.3 编写CORBA数据库应用系统	629	25.14 隐藏/显示桌面上的图标	654
24.3.1 创建对象服务器	629	25.15 获取窗口标题栏中的文字	654
24.3.2 创建客户应用程序	632	25.16 制作软件封面	655
24.4 VisiBroker简介	637	25.17 隐藏/显示Windows任务栏	658
24.4.1 OSAgent	637	25.18 使应用程序不出现在 Windows的任务栏上	658
24.4.2 OSFind	637	25.19 状态栏编程	658
24.4.3 OAD和IREP	638	25.20 存取系统参数信息	665
第25章 其他编程方法与技巧	641	25.21 Windows启动时运行程序	666
25.1 获取Windows版本信息	641	25.22 为程序在启动菜单中创建 快捷方式	667
25.2 获取系统的度量信息和相关 配置信息	642	25.23 在程序中运行其他程序	668
25.3 获取CPU信息	644	25.24 同时只运行程序的一个实例	669
25.4 获取内存信息	645	25.25 关闭运行的其他应用程序	671
25.5 存取系统的颜色信息	646	25.26 显示旋转字体	672
25.6 获取Windows及系统路径	647		

第一部分 C++Builder编程基础

第1章 使用C++Builder进行软件开发

在本章的一开始将对当前计算机技术的发展，尤其是软件开发技术的发展进行概括介绍，使得读者可以从总体上把握计算机软件开发现状，并且有助于读者确定一个中短期的学习计划，甚至对将来的就业、发展等问题也有一些参考价值。此外，在项目开发的时候，这一部分的内容对选择合适的软件开发工具也有一定的帮助。

本章重点介绍了C++Builder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。

另外，在本章的最后，对使用C++Builder编程过程中常用的一些操作步骤进行了统一的介绍，一方面可以简化后面对例程的描述，使本书的内容更加紧凑而物有所值，另一方面可以告诉读者，要实现一个目的，可以有多种方法，在具体的情况下，应选择一种最好的方法。

对于C++Builder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 计算模式的发展概况

计算模式的发展概况可以参考图1-1，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。

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。

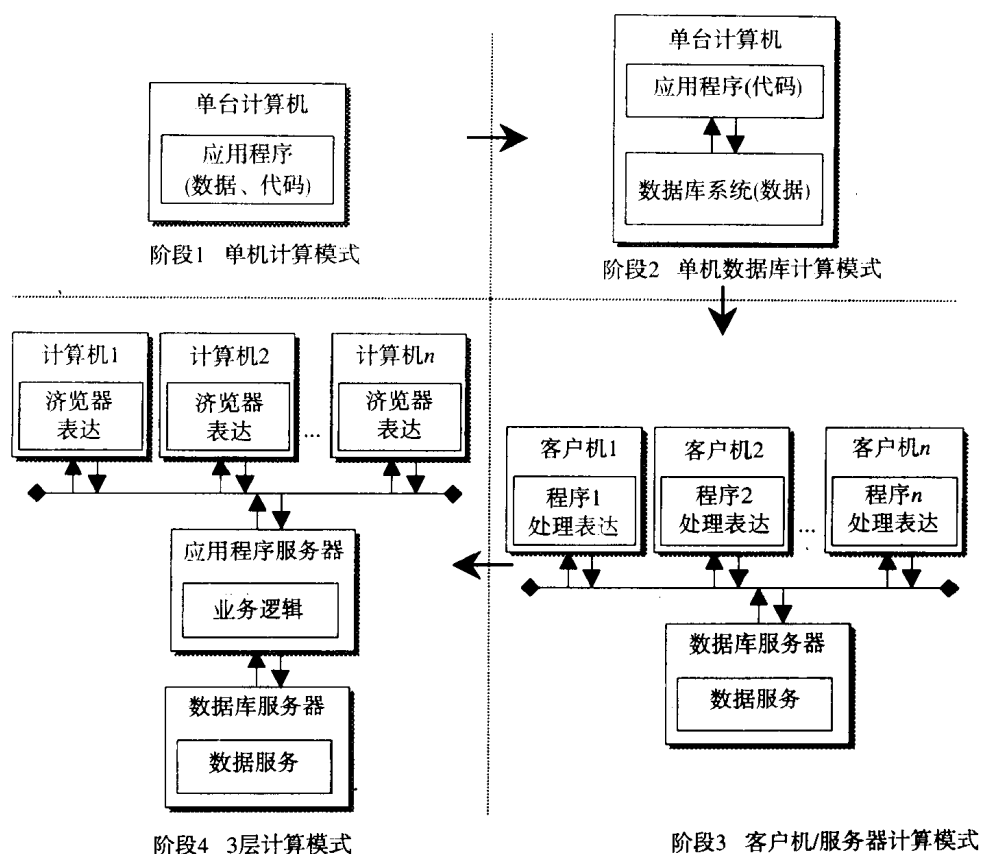


图1-1 计算模式的发展概况图

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~tiers/n~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。

在后面的章节中还将继续深入地介绍关于3层计算模式的众多优点。

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概况

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。本书的第一部分“通用软件开发技术”将对开发此类软件的基本方法进行介绍，比如软件界面的设计、文件操作和打印等。

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。在本书的最后一章有关于对硬件进行操作的一些方法和技巧。

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了。

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，其中很大的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