



天津市高校“十五”规划教材

新编计算机类本科规划教材

Visual C++的使用及 面向对象软件工程实例分析

贺怀清 刘浩瀚 郭 航 编著

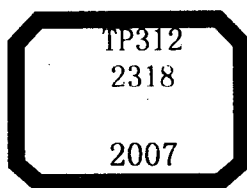
<http://www.phei.com.cn>



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

天津市高校“十五”规划教材
新编计算机类本科规划教材



Visual C++的使用及 面向对象软件工程实例分析

贺怀清 刘浩翰 郭 航 编著



电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书内容分为两部分：第1部分是 Visual C++ 的使用，第2部分是基于 Visual C++ 的面向对象软件工程实例分析。

在 Visual C++ 的使用部分，全面系统地讲述了运用 Visual C++ 进行程序设计的方法，内容涉及基础知识、常用控件的使用、基于对话框的应用程序设计、文档/视图结构、SDI 和 MDI 编程、键盘与鼠标的输入处理、数据库编程和网络编程等。

在基于 Visual C++ 的面向对象软件工程实例分析部分，从面向对象方法、统一建模语言 UML 和运用 UML 的软件开发过程三个方面概述了面向对象软件工程，在此基础上，列举实例进行分析，讲述了在基于 Visual C++ 的软件开发过程中，运用 UML 进行面向对象分析与设计的方法。

本书提供免费电子课件，请登录华信教育资源网下载 (<http://www.huaxin.edu.cn>)。

本书不仅可作为高等学校计算机及相关专业本科学生 Visual C++ 课程的教材，而且可作为那些希望学习和使用 Visual C++ 进行软件项目开发的研究生和工程技术人员学习与参考的教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

Visual C++ 的使用及面向对象软件工程实例分析 / 贺怀清, 刘浩瀚, 郭航编著.

北京: 电子工业出版社, 2007.3

新编计算机类本科规划教材

ISBN 978-7-121-03956-0

I. V… II. ①贺… ②刘… ③郭… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 029286 号

责任编辑: 冉 哲

印 刷: 北京市通州大中印刷厂

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 26.5 字数: 670 千字

印 次: 2007 年 3 月第 1 次印刷

印 数: 4000 册 定价: 34.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

本书是在作者多年从事 Visual C++ 语言的教学和软件项目开发实践基础上编写的, 书中总结了教学和项目开发中的经验和成果, 按照我国学生的学习思维习惯进行内容编排。

Visual C++ 是微软公司在其 Windows 操作系统下进行应用程序开发的最重要的产品, 是系统级应用开发工具, 是运用面向对象思想进行软件项目开发的得力工具。它使得软件开发能够在源码级、类级、控件级等多个级别上重用, 不仅加快了软件开发效率, 而且提高了软件的可重用性和顽健性。因此, 计算机和相关本、专科专业的学生需要学习和掌握它。然而, 由于 Visual C++ 高度封装、高度抽象的特点, 使得学生在学习过程中往往摸不到头绪, 入门很困难。

目前, 市面上有关 Visual C++ 的书很多, 面向的使用对象从初级、中级到高级各不相同, 每本书都有其独特之处。但在教学和学习过程中, 我们发现这样一个现象: 一个知识点的掌握往往需要参考几本书, 而有时几本书的讲解都是一样的, 真正需要的内容找不到, 只能自己去琢磨。鉴于此, 本书在内容讲述上将围绕知识点展开: 首先追根溯源, 将原理讲清楚; 接着, 给出示例, 详细解释知识点在示例中的体现, 并结合示例将编程时需注意的和需掌握的方法渗透给读者; 最后再辅以习题, 习题覆盖知识点, 读者通过思考 and 上机实现习题要求, 达到巩固和掌握知识点的目的。

另外, 目前的 Visual C++ 书在写作侧重点上是不同的, 主要体现在两方面: 或者注重 Visual C++ 本身的使用, 或者注重 Visual C++ 的实例讲解。本书则将 Visual C++ 的使用与运用 Visual C++ 进行软件项目开发的方法结合起来, 因为, 实际上, 能够在面向对象软件工程思想的指导下进行软件开发是学习 Visual C++ 的最终目的。

为此, 本书内容分为两部分: 第 1 部分介绍 Visual C++ 的使用, 第 2 部分是基于 Visual C++ 的面向对象软件工程实例分析。

第 1 部分由第 1~9 章组成。

第 1 章 基础知识。本章是学习 Visual C++ 程序设计的基础, 着重讲述了 Windows 消息机制和 MFC 消息机制, 简要说明了 Windows 应用程序的特点, 介绍了 Visual C++ 集成开发环境的部分用法和概念。

第 2 章 基于对话框界面的应用程序设计。基于对话框的编程是 Visual C++ 最简单但最常用的编程方式。本章通过一个示例的设计与实现讲解对话框的工作原理。

第 3 章 常用控件的使用。Visual C++ 提供了丰富的控件, 它们是应用程序常见的界面元素。本章对控件工具箱提供的最常用的 16 种控件给予了原理描述和示例实现, 它们是: 静态文本、文本编辑框、复选框、单选按钮、分组框、按钮、组合框、列表框、微调控件、进度条、滑杆、热键、列表控件、树形视图控件、标签控件和动画控件。通过对控件的讲解, 使读者进一步熟悉和掌握基于对话框的编程方法。

第 4 章 文档/视图结构。Visual C++ 应用程序的核心是文档/视图结构。本章着重阐释了文档/视图的内在关系, 并通过一个简单示例引出单文档编程的方法。

第 5 章 单文档程序编程。这一章既是对单文档编程方法的进一步讲解, 也是对 Visual C++ 所具有的丰富功能的一点触及。本章以示例的方式着重讲解了在文档/视图结构中通过串行化保存和读取数据的方法, 同时讲解了 Visual C++ 中字体的使用、绘图的实现、无模式对话框的使用以及动态菜单和静态菜单的实现。

第6章 多文档程序编程。多文档编程体现了 Visual C++文档/视图结构的灵活多样的功能。本章以示例方式着重讲解多文档编程方法,总结了 SDI 和 MDI 程序访问当前活动视图和活动文档的方法,介绍了视图分割的概念与方法。

第7章 键盘与鼠标的输入处理。键盘与鼠标是最常用的数据输入和人机交互方式,它们的输入在 Windows 中是以消息的形式出现的。前面几章的示例中涉及了对键盘与鼠标的输入处理,本章则较全面地讲述了键盘与鼠标的消息及相关处理函数,并通过示例强化对这部分知识的学习与理解。

第8章 使用 MFC 编制数据库管理程序。Visual C++提供了多种访问数据库的方法,本章讲述了使用 ODBC 和 DAO 连接数据库的方法,着重讲解了 MFC 数据库封装类及其使用,并给出了数据库编程示例。

第9章 使用 MFC 进行网络编程。本章对 MFC Socket 编程技术进行了较为详细的讲解,包括 Socket 通信接口的使用方式和 MFC 对 Socket 接口进行封装的工作类,并通过示例说明了相关方法的实现。

第2部分由第10~12章组成。

第10章 面向对象软件工程概述。本章从面向对象方法、统一建模语言 UML 和运用 UML 的软件开发过程三个方面概述了面向对象软件工程,并给出了运用 UML 进行面向对象分析与设计的简单示例。

第11章 实例分析1: 人体运动仿真实验系统

本章取材于作者的部分研究工作结果,该实例简要说明了人体运动仿真的背景,着重阐述了运用 UML 进行该系统分析与设计的方法,并给出了部分编码实现。

第12章 实例分析2: 迷宫绘制系统

本章提供了面向对象软件工程应用的第二个实例。在实例分析中,在说明了运用 UML 进行系统分析与设计的方法后,详细给出了系统的编码实现。

本书的所有示例和实例代码均在 Windows XP 平台下的 Visual C++6.0 环境下通过了编译、链接和调试,并正常运行。

本书通俗易懂,由浅入深,循序渐进,理论讲解与示例描述、实例分析相结合,便于学生接受和掌握。

本书不仅可作为高等学校计算机及相关专业本科学生 Visual C++课程的教材,而且可作为那些希望学习和使用 Visual C++进行软件项目开发的研究生和工程技术人员的参考书。

本书由贺怀清、刘浩瀚和郭航共同编著,第1、6、8、9章由郭航编写,第2、3、5、12章由刘浩瀚编写,第4、7章由刘浩瀚与郭航共同编写,第10、11章由贺怀清编写。全书由贺怀清统一修改定稿。此外,王海波提供了第3、4、5章的示例代码,邱智芝参与了资料的收集,调试验证了书中的全部示例代码,贺怀龙绘制了书中部分插图,参与了后期的文字加工和排版工作。

本书得到了天津市教委的大力支持,被列为天津市高校“十五”规划教材——普通高等网络课程建设项目,并得到了中国民航大学机电工程学院副院长罗云林教授的大力支持和热情帮助,还得到了中国民航大学教务处的经费支持,对此作者表示由衷的感谢。

由于作者水平所限,书中难免会有缺点和错误之处,诚恳希望读者不吝赐教。

本书提供免费电子课件,请登录华信教育资源网下载(<http://www.huaxin.edu.cn>),或直接与作者联系(Liuhhq@263.net)。

作者
于天津

目 录

第 1 章 基础知识	1
1.1 Windows 应用程序的特点	1
1.2 深入理解 Windows 消息机制	3
1.2.1 消息基本概念	3
1.2.2 Windows 的消息机制与工作原理	4
1.3 Visual C++ 开发应用程序的特点	12
1.3.1 应用程序工程 (Project)	12
1.3.2 应用程序的总体框架	13
1.4 MFC 的消息映射机制	14
1.4.1 消息映射	14
1.4.2 MFC 处理的三类消息	14
1.4.3 MFC 消息映射的实现方法	15
习题 1	21
第 2 章 基于对话框界面的应用程序设计	22
2.1 对话框的基本概念	22
2.1.1 什么是对话框	22
2.1.2 对话框的组成	22
2.2 对话框的设计与实现	22
2.3 对话框的工作原理	31
2.3.1 对话框的初始化	31
2.3.2 对话框的数据交换机制	31
2.3.3 对话框的运行机制	32
2.3.4 处理控件通知消息	32
2.4 进一步了解工程	33
2.4.1 打开工程	33
2.4.2 浏览工程的内容	33
习题 2	35
第 3 章 常用控件的使用	36
3.1 控件的基本概念	36
3.1.1 什么是控件	36
3.1.2 控件类型	36

3.1.3 控件类	37
3.1.4 传统控件的消息传递	37
3.1.5 控件的使用	38
3.2 静态文本控件	39
3.2.1 静态文本控件类 CStatic 的说明	39
3.2.2 示例 StaticBox 的设计与实现	41
3.3 文本编辑框 (Edit Box)	46
3.3.1 文本编辑框类 CEdit 的说明	47
3.3.2 示例 EditBox 的设计与实现	51
3.4 复选框和单选按钮	54
3.4.1 CButton 类的说明	55
3.4.2 示例 RadioCheck 的设计与实现	60
3.5 组合框	63
3.5.1 组合框控件类 CComboBox 的说明	63
3.5.2 示例 ComboBox 的设计与实现	69
3.6 列表框	73
3.6.1 列表框控件类 CListBox 的说明	73
3.6.2 示例 ListBox 的设计与实现	79
3.7 进度条	83
3.7.1 进度条控件类 CProgressCtrl 的说明	83
3.7.2 示例 Progress 的设计与实现	85
3.8 微调控件	88
3.8.1 微调控件类 CSpinButtonCtrl 的说明	88
3.8.2 示例 Spin 的设计与实现	91
3.9 滑杆	95
3.9.1 滑杆控件类 CSliderCtrl 的说明	95
3.9.2 示例 Slider 的设计与实现	97
3.10 列表控件	100
3.10.1 与列表控件有关的一些数据类型	100
3.10.2 列表控件类 CListCtrl 的说明	102
3.10.3 示例 ListControl 的设计与实现	106
3.11 树形视图控件	114
3.11.1 与树形视图控件有关的一些数据类型	114
3.11.2 树形视图控件类 CTreeCtrl 的说明	116
3.11.3 示例 TreeCtrl 设计与实现	120
3.12 动画控件	126
3.12.1 动画控件类 CAnimateCtrl 的说明	127
3.12.2 示例 AnimateHotkey 的设计与实现	129
3.13 标签控件	135

3.13.1	标签控件类 CTabCtrl 的说明	135
3.13.2	示例 TabCtrl 的设计与实现	137
3.14	控件使用方法总结	142
3.14.1	Win32 控件共有的通知消息	142
3.14.2	控件特点总结	145
3.14.3	设计新的控件类	147
习题 3		149
第 4 章	文档/视图结构	151
4.1	文档/视图关系	151
4.2	文档/视图结构的类属关系	155
4.3	集合类	159
4.4	单文档编程范例	160
4.4.1	简单示例 Editor	160
4.4.2	MFC 自动生成的框架程序结构剖析	171
习题 4		173
第 5 章	单文档程序编程	174
5.1	字体	174
5.1.1	创建字体的结构与函数	174
5.1.2	设备上下文类 CDC	176
5.1.3	示例 Font 的设计与实现	180
5.2	图形	185
5.2.1	设备上下文类的派生类	186
5.2.2	MFC 设备上下文类的使用	187
5.2.3	MFC 对图形对象类的封装	188
5.2.4	MFC 绘图对象类的使用	192
5.2.5	示例 Graphics 的设计与实现	193
5.3	添加弹出对话框	199
5.3.1	对话框的种类	199
5.3.2	示例 Dialog 的设计与实现	200
5.4	串行化	209
5.4.1	串行化的概念	209
5.4.2	实现串行化的方法	210
5.4.3	示例 Serialize 的设计与实现	211
5.4.4	关于串行化的小结	220
5.5	菜单	220
5.5.1	静态菜单	220
5.5.2	动态菜单	223

5.5.3 示例 Menu 的设计与实现	225
习题 5	237
第 6 章 多文档程序编程	239
6.1 多文档窗口特点	239
6.2 MDI 示例程序 Draw 的设计与实现	240
6.2.1 首先用 AppWizard 生成示例程序的工程框架	240
6.2.2 设计示例程序的文档类	243
6.2.3 设计示例程序的视图类	249
6.3 访问当前活动视图和活动文档的方法	256
6.3.1 SDI 程序	257
6.3.2 MDI 程序	257
6.4 视图的分割	257
6.4.1 什么是视图分割	257
6.4.2 视图分割的方法	258
6.4.3 支持多个文档类型	260
习题 6	262
第 7 章 键盘与鼠标的输入处理	263
7.1 鼠标消息与处理函数	263
7.1.1 鼠标消息	263
7.1.2 鼠标消息的处理函数	264
7.1.3 捕捉鼠标	267
7.1.4 改变鼠标光标	268
7.2 鼠标消息处理示例 Mouse	269
7.2.1 创建工程框架	269
7.2.2 添加自定义类	269
7.2.3 文档设计	272
7.2.4 视图设计	273
7.3 键盘消息与处理函数	281
7.3.1 键盘消息	281
7.3.2 键盘消息响应函数	282
7.3.3 键盘插入符	282
7.4 键盘消息处理示例 TestKeyboard	283
7.4.1 创建工程框架	283
7.4.2 为示例添加变量和类内函数	283
7.4.3 为示例添加类	284
习题 7	286

第 8 章 使用 MFC 编制数据库管理程序	287
8.1 数据库概论	287
8.1.1 数据库的基本概念	287
8.1.2 使用 ODBC 连接数据库	290
8.2 使用 MFC 对数据库编程	294
8.2.1 MFC 数据库封装类	294
8.2.2 MFC 数据库类的使用	295
8.3 数据库编程示例 Enroll	306
8.3.1 注册数据源	306
8.3.2 Enroll 工程的设计与实现	307
8.3.3 使用 ODBC 的 API 函数直接修改数据	310
8.4 DAO 简介	319
8.4.1 什么是 DAO	319
8.4.2 DAO 和 ODBC	320
8.4.3 DAO 的特色	321
8.4.4 用 DAO 创建表	322
习题 8	323
第 9 章 使用 MFC 进行网络编程	324
9.1 网络编程概论	324
9.1.1 网络编程与 Socket 接口	324
9.1.2 WinSock API 的使用方法	326
9.2 MFC 对 Socket 的封装	329
9.2.1 CAsyncSocket 类	329
9.2.2 CSocket 类	331
9.2.3 CsocketFile 类	332
9.3 使用 WinSock 进行通信	333
9.3.1 有连接的通信	333
9.3.2 无连接的通信	335
9.4 网络编程示例	337
9.4.1 Server 端的实现	337
9.4.2 Client 端的实现	345
习题 9	351
第 10 章 面向对象软件工程概述	352
10.1 面向对象方法	352
10.1.1 面向对象思想	352
10.1.2 面向对象软件开发基本方法与步骤	353
10.2 统一建模语言 UML	354

10.2.1 UML 的目标与特征	354
10.2.2 UML 的组成结构	355
10.2.3 UML 的模型图	355
10.2.4 结构型的图	356
10.2.5 行为型的图	358
10.2.6 其他特征	362
10.3 运用 UML 的软件开发过程	362
10.3.1 需求分析	362
10.3.2 系统设计	363
10.3.3 系统开发	364
10.3.4 部署	364
10.4 一个简单示例——校园绿化设计系统	365
10.4.1 校园绿化设计系统的需求分析	365
10.4.2 校园绿化设计系统的详细设计	366
习题 10	370
第 11 章 实例分析 1: 人体运动仿真实验系统	371
11.1 系统背景	371
11.2 需求分析	371
11.3 系统设计	372
11.3.1 用例图	372
11.3.2 活动图	372
11.3.3 顺序图	372
11.3.4 类图	374
11.3.5 其他数据结构	374
11.4 重要功能的编码实现	376
11.4.1 人体建模——organ 类	376
11.4.2 运动仿真的组织与实现——action 类	381
11.4.3 各关节运动变化触发函数	386
11.5 系统界面与模拟结果	387
习题 11	388
第 12 章 实例分析 2: 迷宫绘制系统	389
12.1 系统介绍	389
12.2 需求分析	389
12.3 系统设计	390
12.4 关键技术实现	392
12.4.1 使用面向对象方法	392
12.4.2 图形的编辑功能	392

12.4.3	对象在内存中的存储	395
12.4.4	把图形对象存储到磁盘文件中	395
12.4.5	光标形状设置	395
12.5	程序运行过程及原理概述	396
12.6	关键代码讲解	397
12.6.1	实现 CShape 类的关键代码	397
12.6.2	图形类的定义	400
12.6.3	添加到视图类中的成员	401
12.6.4	视图类中主要成员函数的实现	401
12.6.5	添加到文档类的数据成员	407
12.6.6	文档类中主要成员函数的实现	407
习题 12		408
参考文献		409

第 1 章 基础知识

1.1 Windows 应用程序的特点

1. 图形用户界面

Windows 应用程序运行时，会弹出一个窗口。窗口中有工具栏、菜单栏和一些控件，它们组成了应用程序的界面，这就是所谓的图形用户界面（GUI，Graphic User Interface）。它具有用户界面友好，操作简单等特点。用户不需要记住很多命令就能用鼠标完成许多复杂的功能，如图 1.1 所示。

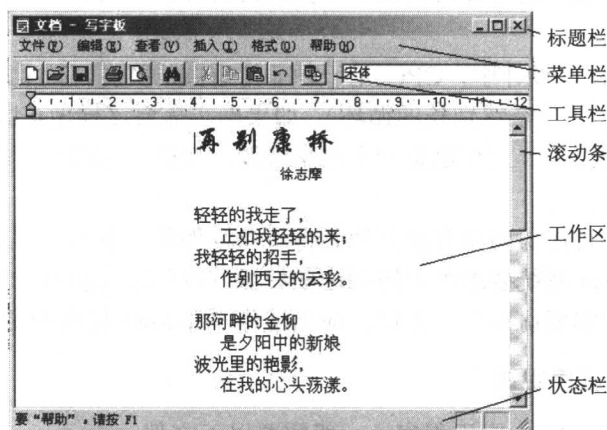


图 1.1 图形用户界面

2. 事件驱动模型

应用程序的运行是靠事件驱动的，鼠标的单击、双击和移动，敲击键盘，窗口建立或撤销，内容改变等都会产生事件。另外，操作系统内部也会产生各种事件。事件产生后会发出相应的消息，这些消息会发送给各个对象，这些对象接收到消息后会执行相应的函数，以完成指定的功能。这些函数叫做消息映射函数。应用程序在运行时就不断地检查是否有消息发送过来，接收到什么样的消息就执行与它相对应的函数，这就是所谓的消息循环机制。

例如，我们打开系统附件中的计算器应用程序，它的界面如图 1.2 所示。应用程序显示完界面后就开始不断地检查是否有消息发送过来，如果没有消息发送过来它什么也不会做。用户单击菜单中的某一项或单击下面的按钮时都会产生一个消息，应用程序接收到这些消息后会执行相应的消息映射函数以完成某种功能。用户单击计算器上的数字按钮就能将数



图 1.2 计算器

据输入到计算器中。单击其他功能按钮就会执行特定的功能,例如,用户单击求平方根(sqrt)按钮,应用程序就会对编辑框中的数据计算平方根并把结果输出到编辑框中。

3. 资源共享

Windows 是一个多任务的操作系统,各个应用程序共享系统提供的资源,常见的资源包括:设备上下文、画刷、画笔、字体、对话框控制、对话框、图标、定时器、插入符号、通信端口、电话线等。

而 Windows 要求应用程序必须以一种能允许它共享 Windows 资源的方式进行设计,它的基本模式是这样的:

- ① 向 Windows 系统请求资源;
- ② 使用该资源;
- ③ 释放该资源给 Windows 以供别的程序使用。

即使最有经验的 Windows 程序员也常常会忽略第③步。如果忽略了这一步,轻则当时不出错,但过一会儿出现程序运行异常情况,干扰别的程序正常运行;重则立即死机,例如,设备上下文没有释放。

在 Windows 应用程序设计中,CPU 是一种非常重要的资源,因此应用程序应当避免长时间地占用 CPU 资源(如一个特别长的循环)。如果确实需要这样做,也应当采取一些措施,让程序能够响应用户的输入。主存也是一个共享资源,要防止同时运行的多个应用程序因协调不好而耗尽内存资源。

应用程序一般不要直接访问内存或其他硬件设备,例如,键盘、鼠标、计数器、屏幕或串口、并口等。Windows 系统要求绝对控制这些资源,以保证向所有的应用程序提供公平的不中断的运行。如果确实要访问串、并口,应当使用 Windows 提供的函数来安全地访问。

4. Windows 应用程序组成

编写一个典型的 Windows 应用程序,一般需要以下文件。

① .CPP 或.C 源程序文件。源程序文件包含了应用程序的数据、类、功能逻辑模块(包括事件处理、用户界面对象初始化以及一些辅助例程)的定义。

② .H 头文件。头文件中包含了.CPP 或.C 源文件的所有数据、模块、类的声明。当一个.CPP 或.C 源文件要调用另一个.CPP 或.C 中所定义的模块功能时,需要包含那个.CPP 或.C 文件对应的头文件。

③ 资源文件。资源文件中包含了应用程序所使用的全部资源定义,通常以.RC 为后缀名。注意,这里说的资源不同于前面提到的资源,这里的资源是应用程序所能够使用的一类预定义工具中的一个对象,包括:字符串资源、快捷键表、对话框、菜单、位图、光标、工具条、图标、版本信息和用户自定义资源等。

这里需要解释的是资源文件。在 Windows 程序设计过程中,菜单、对话框、位图等可视的对象被单独分离出来加以定义,并存放在资源源文件中,然后由资源编译程序编译为应用程序所能使用的对象的映象。资源编译使应用程序可以读取对象的二进制映象和具体数据结构,这样可以减少为创建复杂对象所需要的程序设计工作。程序员在资源文件中定义应用程序所需使用的资源,资源编译程序编译这些资源并将它们存储于应用程序的可执行文件或动

态连接库中。在 Windows 应用程序中引入资源有以下好处。

① 降低内存需求。当应用程序运行时，资源并不随应用程序一起装入内存，而是在应用程序实际用到这些资源时才装入内存。在资源装入内存时，它们拥有自己的数据段，而不驻留于应用程序数据段中。当内存紧张时，可以废弃这些资源，使其占用的内存空间供他用，而当应用程序用到这些资源时才自动装入。这种方式降低了应用程序的内存需求，使得一次可以运行更多的程序。这也是 Windows 内存管理的优点之一。

② 便于统一管理和重复利用。例如，如果在应用程序中要多次用到一个代表公司的徽标位图，就可以将它存放在资源文件中，每次用到时再从资源文件中装入。这种方式比将位图放在一个外部文件中更加简单有效。

③ 应用程序与界面有一定的独立性，有利于软件的国际化。由于资源文件独立于应用程序设计，使得在修改资源文件时（如调整对话框大小、对话框控制位置），可以不修改源程序，从而简化了用户界面的设计。另外，目前所提供的资源设计工具一般都是采用“所见即所得”方式，这样就可以更加直观、可视地设计应用程序界面。由于资源文件的独立性，软件国际化工作也非常容易。例如，现在开发了一个英文版的应用程序，要想把它汉化，只需要修改资源文件，将其中的对话框、菜单、字符串资源等汉化即可，无须修改源程序。

但是，应用程序资源只是定义了资源的外观和组织，而不是其功能特性。例如，编辑一个对话框资源，可以改变对话框的安排和外观，但是却没有也不可能改变应用程序响应对话框控制的方式。外观的改变可以通过编辑资源来实现，而功能的改变却只能通过改变应用程序的源代码，然后重新编译来实现。Windows 应用程序的生成也要经过编译、链接两个阶段，只是增加了资源编译过程，基本流程如图 1.3 所示。

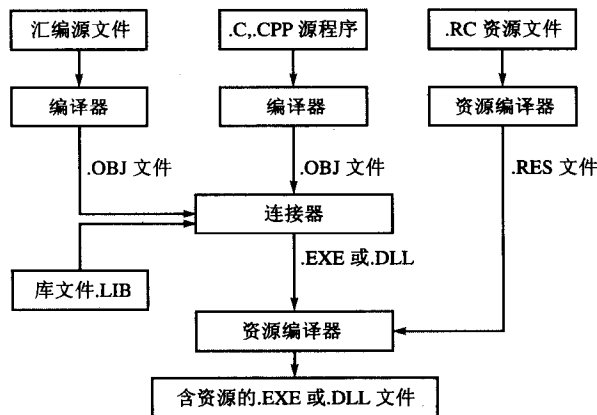


图 1.3 应用程序生成过程

1.2 深入理解 Windows 消息机制

1.2.1 消息基本概念

微软的 Richard Wilton 曾经给出一种定义：“消息的概念可以这样说，在 Windows 环境中，消息是指从 Windows 调用的函数传递给应用程序出口函数的固定不变的数据集。”学过基本 Windows SDK（Standard Develop Kit）编程的人，看到这些话就会很清楚地知道了

消息的概念。如果这个概念看不懂的话，我们可以从另外一个角度，先说一点有关消息的知识。Windows 系统中有一个消息系统，有且只有一个。还有很多个消息队列，消息系统能接收到所有的消息，并且它还能够把消息发给任何应用程序。当然应用程序之间也可以互相发送消息。应用程序一旦接收到消息就需要对所收到的消息进行反应，这种反应包含实际处理与不处理两种。如果要对这个消息真正处理的话，就要针对这个消息的要求调用程序进行处理；否则就不处理，但必须以最快的速度继续传递给其他的应用程序。不同的消息会有不同的处理，例如，WM_PAINT 是 Windows 系统常用的消息，它的作用是通知应用程序要在显示器上输出显示内容。对有些消息，需要添加另外的程序代码来处理；而有些消息，在 Visual C++编程的时候，所产生的程序框架就已经能对它进行处理了，没必要再编程了。

消息系统对于一个 Windows 程序来说十分重要，它是一个程序运行的动力源泉。Windows 消息分为 4 类：输入消息、控制消息、系统消息和用户消息。但 Windows 系统对它们的处理过程是一样的，都是由消息队列管理其接收、处理与发送。归根到底，一个消息是 Windows 系统定义的一个 32 位的值，它唯一地定义了一个事件，向 Windows 发出一个通知，告诉应用程序某个事件发生了。例如，单击鼠标、改变窗口尺寸、按下键盘上的一个键都会使 Windows 发送一个消息给应用程序。

消息可以由系统或应用程序产生。系统在发生输入事件时产生消息，例如，用户敲键，移动鼠标或者单击控件；系统也产生消息以响应由应用程序带来的变化，如应用程序改变系统字体或改变窗体大小。应用程序可以产生消息使窗体执行任务，或者与其他应用程序中的窗口通信。

产生消息的来源可以大致分为 4 类：

- ① 用户产生消息；
- ② Windows 本身产生消息；
- ③ 应用程序本身产生消息；
- ④ 其他应用程序产生消息。

1.2.2 Windows 的消息机制与工作原理

1. Windows 消息结构

一个消息包含一个消息名称（UINT）和两个参数（WPARAM, LPARAM）。当用户进行了输入或窗口的状态发生改变时，系统都会发送消息到某一个窗口。例如，当选中菜单之后会有 WM_COMMAND 消息发送，WPARAM 的高字中（HIWORD（wParam））是命令的 ID 号，对菜单来讲就是菜单 ID。当然，用户也可以定义自己的消息名称，还可以利用自定义消息来发送通知和传送数据。消息本身是作为一个记录传递给应用程序的，这个记录中包含了消息的类型及其他信息。例如，对于单击鼠标所产生的消息来说，这个记录中包含了单击鼠标时的坐标。这个记录类型叫做 MSG。MSG 含有来自 windows 应用程序消息队列的消息信息，它在 Windows 中声明如下：

```
typedef Struct TagMSG
{ HWND    hWnd;           //接收该消息的窗口句柄
```

UINT message;	//消息常量标识符,也就是我们通常所说的消息号
WPARAM wParam;	//高 32 位消息参数,确切含义依赖于消息值
LPARAM lParam;	//低 32 位消息参数,确切含义依赖于消息值
DWORD time;	//消息创建时的时间
POINT pt;	//消息创建时,鼠标/光标在屏幕坐标系中的位置

}MSG

hWnd 是 32 位的窗口句柄类型 (HWND) 变量。什么是窗口句柄? 它实际就是一个指针。系统通过窗口句柄来在整个系统中唯一地标识一个窗口。发送一个消息时必须指定一个窗口句柄表明该消息由哪个窗口接收。而每个窗口都会有自己的窗口过程,所以用户的输入就会被正确地处理。例如,有两个窗口公用一个窗口过程代码,在窗口 1 中按下鼠标时消息就会通过窗口 1 的句柄被发送到窗口 1 中而不是窗口 2 中。

message 用于区别其他消息的常量值,这些常量可以是 Windows 单元中预定义的常量,也可以是自定义的常量。消息标识符以常量命名的方式指出消息的含义。当窗口过程接收到消息之后,它就会使用消息标识符来决定如何处理消息。例如,WM_PAINT 告诉窗口过程,窗体客户区被改变了需要重绘。符号常量指定系统消息所属的类别,其前缀指明了处理解释消息的窗体的类型。

wParam 通常是一个与消息有关的常量值,也可能是窗口或控件的句柄。

lParam 通常是一个指向内存中数据的指针。

2. Windows 消息标识符的分类和数值范围

系统保留消息标识符的值在 0x0000~0x03ff (WM_USER-1) 范围内,这些值被系统定义消息使用,应用程序不能给自己的消息使用这些值。应用程序消息的范围为 WM_USER (0x0400) ~0x7fff,或 0xc000~0xffff; WM_USER~0x7fff 的消息由应用程序自己使用; 0xc000~0xffff 的消息用来和其他应用程序通信。我们顺便说一下具有标志性的消息值:

WM_NULL~0x0000	空消息
0x0001~0x0087	主要是窗口消息
0x00A0~0x00A9	非客户区消息
0x0100~0x0108	键盘消息
0x0111~0x0126	菜单消息
0x0132~0x0138	颜色控制消息
0x0200~0x020A	鼠标消息
0x0211~0x0213	菜单循环消息
0x0220~0x0230	多文档消息
0x03e0~0x03e8	DDE 消息
0x0400~WM_USER	
0x8000~WM_APP	
0x0400~0x7fff	应用程序自定义私有消息

其实,Windows 中的消息虽然很多,但是种类并不繁杂,大体上有 3 种:窗口消息、命令消息和控件通知消息。