

普通高校本科计算机专业特色教材精选·算法与程序设计

MFC Windows 应用程序设计 (第3版)

任 哲 等编著

清华大学出版社



01333434

TP312C-43
03-3

普通高校本科计算机专业特色教材精选·算法与程序设计

MFC Windows 应用程序设计(第3版)

任 哲 等编著



TP312C-43
03-3

清华大学出版社
北京



北航

C1640000

内 容 简 介

本书为《MFC Windows 应用程序设计》的第3版。本书沿袭前一版的编写原则和风格,在读者学习了C/C++的基础上,从Windows SDK程序设计出发,用C++类对Windows应用程序进行了模拟封装,重点介绍MFC的应用程序框架。随后以精练的语言介绍Windows的图形图像处理、应用程序界面、资源、文件、动态链接库、进程与线程管理等知识。最后,介绍组件对象模型(COM)、ActiveX技术、数据库和异常处理的基础知识。

本书内容丰富、结构合理、重点突出、注重应用、兼顾说理性和实用性、文字通俗易懂,以适当的方法加强了对MFC重点技术核心内容的介绍,使读者可以在有限的文字篇幅情况下获得尽可能大的信息量,适合作为普通高等院校的教学及教学参考用书,也可以作为其他计算机开发应用人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

MFC Windows 应用程序设计/任哲等编著. —3版. —北京:清华大学出版社,2013.3

普通高校本科计算机专业特色教材精选·算法与程序设计

ISBN 978-7-302-31273-4

I. ①M… II. ①任… III. ①C语言—程序设计—高等学校—教材 ②Windows操作系统—程序设计—高等学校—教材 IV. ①TP312 ②TP316.7

中国版本图书馆CIP数据核字(2013)第008367号

责任编辑:汪汉友

封面设计:常雪影

责任校对:白蕾

责任印制:何芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:28.25

字 数:636千字

版 次:2004年6月第1版

2013年3月第3版

印 次:2013年3月第1次印刷

印 数:1~3000

定 价:49.00元

产品编号:047602-01

出版说明

INTRODUCTION

在我国高等教育逐步实现大众化后，越来越多的高等学校将会面向国民经济发展的第一线，为行业、企业培养各级各类高级应用型专门人才。为此，教育部已经启动了“高等学校教学质量和教学改革工程”，强调要以信息技术为手段，深化教学改革和人才培养模式改革。如何根据社会的实际需要，根据各行各业的具体人才需求，培养具有特色显著的人才，是我们共同面临的重大问题。具体地说，培养具有一定专业特色的和特定能力强的计算机专业应用型人才则是计算机教育要解决的问题。

为了适应 21 世纪人才培养的需要，培养具有特色的计算机人才，急需一批适合各种人才培养特点的计算机专业教材。目前，一些高校在计算机专业教学和教材改革方面已经做了大量工作，许多教师在计算机专业教学和科研方面已经积累了许多宝贵经验。将他们的教研成果转化为教材的形式，向全国其他学校推广，对于深化我国高等学校的教学改革是一件十分有意义的事情。

清华大学出版社在经过大量调查研究的基础上，决定组织出版一套“普通高校本科计算机专业特色教材精选”。本套教材是针对当前高等教育改革的新形势，以社会对人才的需求为导向，主要以培养应用型计算机人才为目标，立足课程改革和教材创新，广泛吸纳全国各地的高等院校计算机优秀教师参与编写，从中精选出版确实反映计算机专业教学方向的特色教材，供普通高等院校计算机专业学生使用。

本套教材具有以下特点：

1. 编写目的明确

本套教材是在深入研究各地各学校办学特色的基础上，面向普通高校的计算机专业学生编写的。学生通过本套教材，主要学习计算机科学与技术专业的基本理论和基本知识，接受利用计算机解决实际问题的基本训练，培养研究和开发计算机系统，特别是应用系统的基本能力。

2. 理论知识与实践训练相结合

根据计算学科的三个学科形态及其关系,本套教材力求突出学科的理论与实践紧密结合的特征,结合实例讲解理论,使理论来源于实践,又进一步指导实践。学生通过实践深化对理论的理解,更重要的是使学生学会理论方法的实际运用。在编写教材时突出实用性,并做到通俗易懂,易教易学,使学生不仅知其然,知其所以然,还要会其如何然。

3. 注意培养学生的动手能力

每种教材都增加了能力训练部分的内容,学生通过学习和练习,能比较熟练地应用计算机知识解决实际问题。既注重培养学生分析问题的能力,也注重培养学生解决问题的能力,以适应新经济时代对人才的需要,满足就业要求。

4. 注重教材的立体化配套

大多数教材都将陆续配套教师用课件、习题及其解答提示,学生上机实验指导等辅助教学资源,有些教材还提供能用于网上下载的文件,以方便教学。

由于各地区各学校的培养目标、教学要求和办学特色均有所不同,所以对特色教学的理解也不尽一致,我们恳切希望大家在使用教材的过程中,及时地给我们提出批评和改进意见,以便我们做好教材的修订改版工作,使其日趋完善。

我们相信经过大家的共同努力,这套教材一定能成为特色鲜明、质量上乘的优秀教材。同时,我们也希望通过本套教材的编写出版,为“高等学校教学质量和教学改革工程”做出贡献。

清华大学出版社

前言

CONTENTS

不经意间，距本书第1版出版已经有近十年光景了，而微软的 Visual C++ 6.0，特别是其核心项目 MFC，自其 1992 年初发布的 MFC 1.0（即 MS C/C++ 7.0）开始也已经在业界驰骋拼杀 20 个年头了。本书的不断改版，固然是读者的抬爱，对于 MFC 来说，则不能不说是一个奇迹，因为迄今为止，在软件业发展史上除了 MFC 还没有一款软件受到如此瞩目。对之大加赞扬，乃至成为其忠实粉丝的拥趸者有之，而对其投以藐视、甚至鄙视的目光，称其代码丑陋无比而大加挞伐的也为数众多，但这种毁誉参半的评价却未妨碍它的广泛应用。这究竟是 Visual C++ 6.0 代码并不像藐视者所说的那么丑陋，还是因其代码效率高而没有太在意它的丑陋而使人们勉为其难地在使用它？不得而知。尽管在有了 Java 和 .NET 的今天，以 MFC 为主的 Visual C++ 6.0 仍然在桌面系统程序开发上以其独有的高效率优势而占有一定市场份额。

随着 IT 业的发展，新程序开发工具的不断涌现，不知从何时起，关于 MFC 又多个了一个话题：“MFC 是否已经过时？”有人说它从出生那天起就过时了，也有人说它始终没过时，甚至时不时地有人在预测它什么时候过时以及将由什么工具来代替。一直到微软公司自己推出了自认为可代替 MFC 的 .NET 之后，人们认为 MFC 这回真的是要寿终正寝了，谁知到了 .NET 已经推出了 Visual Studio 2010 的当下，MFC 仍然像幽灵一样徘徊于业界各个角落。

作为本书的作者，经常收到一些读者来信询问诸如上述的 MFC 会不会过时，学习 MFC 是不是还有意义之类的问题。对于头一个问题，我的回答是，不知道，真不知道。对于第二个问题我的回答是，有意义，非常有意义，因为在应用程序框架领域，MFC 是一款具有历史意义的软件，它之所以能够在质疑和争论中被应用到今天，那就是因为它有巨大成功之处，也有因软件工程发展阶段限制的失误及失败之处，这一切都是软件业的财富。对这种财富视而不见，不学习，不继承，岂不是傻瓜？

众所周知,自面向对象程序设计思想出现之后,C++ 是第一个借助 C 的优势而真正在业界广泛推广了起来的面向对象编程语言,而 MFC 则是第一个使用 C++ 编写的 Windows 应用程序框架,正是因为有了它(当然还有 VB),那些深陷于 Windows 数千个 API 函数中的程序员才被解放了出来,也才有了以后 Windows 应用程序的蓬勃发展。

正因为这两个第一次,即两个首次,所以 C++ 语言的稚嫩和不完善(例如不合逻辑的多继承,以及还没有接口的概念等),以及 MFC 设计者们对面向对象程序设计思想理解得不深刻,使得 MFC 的有些代码看起来确实丑陋,例如它那难于理解的六组宏,层次过多的类继承关系,等等。但就当时的条件,事实求是地说,微软已经做得足够好了,更何况在它发展 ATL 时很快就做了方向正确的改进,这也可能是 Visual C++ 6.0 能坚持到今天的原因之一吧。

总之,放着眼前这个透漏了微软前期技术、发展轨迹以及其成功和失误(这些东西都在后来极大地影响了 Java 和 .NET 的 C#)的现成实例不借鉴、不学习,是不明智的。所以作者始终认为,即使将来 MFC 最终退出了历史,作为专攻计算机技术的人,特别是有志于计算机软件技术的学生,任何时候都要学习 MFC,必要时还要对其成功和失败之处进行研究探讨。

为了在一定程度上把 MFC 与 .NET 联系起来,第 3 版改写了第 1 章并添加了第 18 章。第 1 章中,在介绍 Windows 应用程序结构的同时,把微软在操作系统(平台)的发展中所涉及的 DOS、Windows 和 .NET 之间的关系做了尽可能简单的介绍。在新添加的第 18 章中,为了说明 MFC 在 .NET 中的地位,也为了读者以后学习 .NET 的方便,在较为详细地介绍了 .NET 这个平台的相关技术之后介绍了非托管的 MFC 和托管的 C++ /CLI,企图使读者对 Visual C++ 在 Windows 中的应用有一个相对清晰的了解和认识,从而不至于让微软的那些让人眼花缭乱的术语弄得晕头转向。

因本书是为普通高等学校计算机或计算机软件工程专业学生编写的教学用书,所以对于那些需要着重学习编程技巧的读者并不适合,请慎重购买。

本书在编写过程中参阅了大量参考文献和网上资料,在此谨对这些文献、资料的作者及提供者表示衷心的感谢。同时也对本书的前两版提出意见、建议以及来信对作者给予鼓励和支持的读者表示诚挚的感谢。当然,还要感谢清华大学出版社。

参加本书编写的有房红征、陈涛、李益民、车进辉和任哲,由任哲负责统稿。

希望读者喜欢本书。如对本书有意见和建议,请与作者联系。作者电子邮箱为 renzhe71@sina.com。

最后,希望本书能给各位带来好心情!

作者

2013 年 2 月

第2版 前言

由于广大读者的抬爱及清华大学出版社的鼓励，本书有幸再版。在此首先对广大读者和出版社表示衷心的感谢。自本书第1版出版以来，作者陆续收到一些读者来信，针对这些来信，作者准备谈以下两个问题。

1. 为什么要学习 MFC

作者在第1版前言中就说过，通过对MFC的学习，除了可以使学生掌握一种Windows应用程序设计的基本方法之外，还可以使他们进一步全面、深刻地理解面向对象程序设计的思想，从而把握程序设计方法的发展方向。对此，曾经有读者提出过疑问：Visual Basic、Delphi、C#.NET、Java不都是面向对象的吗？为什么非要学习目前看起来似乎有些落后的MFC呢？

诚然，从时间上来看，MFC是微软较早开发的Windows应用程序框架，历史比较悠久。但是，请读者注意，历史悠久与落后绝不是同一概念。当然也不意味着只要学会了MFC就可以打遍天下，解决所有问题，但就MFC所蕴涵及所展露的程序设计思想、代码实现技巧，则是其他开发工具所不及的。

作为世界上第一个应用程序框架类库的MFC，是在C++尚未完善，尚不支持某些功能的时候产生的，MFC设计者们就是使用这种当时还比较低级的C++实现了这个庞大程序框架的各种复杂功能，因此MFC的各个角落都充满了这些设计者的智慧和技巧，这些智慧和技巧都体现了当今软件工程技术的重要思想和发展方向，以至于对C++的发展和随后出现的Java、C#技术等，都产生了极为深刻的影响，甚至有些就是MFC技术的直接应用。作者认为，迄今为止还没有一个应用程序框架类库能具有如此之大的信息量，所以它是计算机专业教育和教学的绝好素材。

作者认为，通过对MFC的学习至少可以达到如下几个目的：一是可以学习Windows应用程序的特点（图形界面、多任务、事件驱动）、数据和函数的组织方法（资源、动态链接库）和结构（接收、分派消息的主程序模块和处理消息的消息相应程序模块）及其处理思想；二是通过MFC这个大实例，可以学习面向对象程序设计的实际方法（例如文档/视图结

构的构思);三是可以学习 MFC 设计者们的一些巧妙的程序设计方法和技巧(例如消息映射、对象动态创建、序列化);四是可以学习 Visual C++ 这个开发环境的使用,真可谓是一举多得。所以,作为每一个立志从事计算机工作的人来讲,不在一定程度上了解和掌握 MFC 将会是他的终生遗憾。

当然,其他开发工具也有它们各自的特点和优势,也值得学习。例如,Visual Basic 作为微软开发的第一套可视化开发环境,是它把 DOS 程序员从汪洋大海般的 Windows API 函数中解救出来,为 Windows 应用程序的设计和发展做出了不可磨灭的贡献。但是,它使用的并不是非学不可的 Basic 这种非主流语言,所以对于学时极为有限和宝贵的学生来说,是否非得在校学习它就值得考虑。而使用 PASCAL 语言的 Delphi 和使用 C/C++ 语言的 C++ Builder 以其提供的丰富组件,在快速开发应用程序方面确实具有一定的优势,但它们与 Visual Basic 一样,把 Windows 底层封装得太严了,程序员只剩下填空的份儿,以至于使它们成了一种“傻瓜”式的开发工具,这当然不适合以教授和学习程序设计思想和设计方法为目的的学校来使用。

当然,作者也认可业界“聪明人用 Delphi,高手用 VC”之类的说法,因为在实际工程中,为了满足工程期限的需要,而采用 Delphi 之类的快速开发工具是合理并应该的。但并不意味着学生在校期间一定要拿出大量的课时去学习这些快速开发工具(工具无所谓会不会的问题,而是熟不熟的问题),因为高等学校毕竟是教育机构而不是培训机构,不但要让学生知其然也更应该让其知其所以然,从而在此基础上形成较强的软件开发(甚至是研发)能力,而不是只让学生会几门流行的技术而使学生在技术发展背后疲于奔命。

那么学习 Windows SDK 编程又如何呢?作者认为,用函数编程这是 C 程序设计课程解决的问题,学生学习了 C 语言之后只要肯下工夫,多实践,自学 Windows SDK 程序设计是完全可以的,没有必要再浪费宝贵的课时了。

也正因为上述种种原因,所以本书既不以剖析 MFC 源代码为主,也不以介绍 Visual C++ 开发工具的使用为主,而是以重点介绍 MFC 框架的设计思想、理念和方法为基础,兼顾应用,从而使读者通过对 MFC 的学习,除了掌握 MFC 之外,还能进一步深化理解 C/C++、数据结构、操作系统乃至计算机硬件的相关知识,尽快提高程序设计能力并学习其他应用程序设计工具打下良好的基础。

2. 第 2 版的变化

自本书的第 1 版出版以来,笔者陆续接到许多读者的来信,其中既有教师也有学生和程序设计人员,在信中都在对本书褒奖之余对书的内容及编写提出了很好的意见与建议,并希望作者在有机会的条件下进行修改。在此,作者首先对这些关心本书的读者表示由衷的谢意。

综合了读者的意见之后,《MFC Windows 应用程序设计(第 2 版)》特做如下变化:

因为本书旨在说明 MFC 的思想,所以只要能以模拟方式说明其思想的,本书就采用了模拟方法。显然本书第 1 版在消息映射的模拟和叙述上过于潦草,给一部分读者学习这部分时造成了一些不必要的困难。故此,作者在第 2 版中进行了必要的修改,使得这部分的介绍更为合理、清楚。

文档/视图结构是 MFC 应用程序框架的核心。为此, 作者在第 2 版的第 3 章中适当增加了文档/视图结构程序框架各对象的创建顺序、对象之间的联系、消息的传递等内容, 以期使读者对文档/视图结构及其各对象之间的关系有更清楚的认识和了解, 为读者精研 MFC 打下一个更加良好的基础。另外, 在这一章中还加强了对对象动态创建的介绍。

参考读者建议, 作者把第 9 章的内容在顺序上做了改变: 先介绍相对比较容易的 CFile 类, 后介绍 CArchive 类, 最后介绍文档序列化, 并且加强了序列化部分的说理性。

异常及异常处理是一个健壮的应用程序必备特性, 这方面的内容的缺失不能不说是第 1 版的一个缺憾, 这次在第 2 版中把它补齐了。

当然, 为追求内容的完整性, 本书至少还应该补上 MFC 在网络上的应用 (这也是读者的建议), 但作者思虑再三后认为, 作为教材本书还是不包含这部分内容为好, 一是因为 MFC 在网络程序方面并不占优势, 二是因为各个学校都是在 C#.NET 或 Java 等课程中处理该问题, 如果本书包括这部分内容的话, 在教学上会造成内容的重叠。

另外, 正如一些读者所指出的, 第 14 章“组件对象模型 COM 基础”出现在本书中显得比较生硬: 第一, 组件对象模型的内容对于本科生来讲似乎是难了一些; 第二, 这本书讲的是 MFC, 而这一章讲的是 ATL, 确实与本书不配套。关于第一点, 作者认为不是问题, 学生觉得难, 一是因为没有好的教材, 二是学生原来接触这方面的技术比较少, 但不管是什么原因, 在组件应用如此广泛的今天, 计算机专业特别是以软件设计为培养目标的专业不向学生讲授这方面的知识是无论如何不行的, 但是, 现在大多数本科学校又没有相应的课程来介绍这方面的内容, 再者组件技术把抽象类的应用发挥到了淋漓尽致的程度, 不学它是不会深切领会抽象类的精神实质以及它的应用的, 所以本书就勉为其难地编写了这部分的入门知识。为了降低学习这部分内容的台阶, 作者重写了这一章。至于第二点, 生硬就生硬了吧, 因为 ATL 在组件设计方面确实要比 MFC 强许多。

正如前面所讲, MFC 信息量巨大, 涉及程序设计的方方面面, 因此需要读者在阅读本书时不但要不断地复习 C/C++ 的内容, 还要不时地阅读其他书籍, 尤其是数据结构、操作系统等。

另外, 还要多找一些实例上机做实验 (提供这样例题的书籍及网上资料很多)。

本书在编写过程中参考了大量的相关文献和网上的资料, 并引用了其中的一些例题、文字和插图, 在此对这些文献和资料的作者表示诚挚的谢意。同时也对关心本书的读者表示万分的感谢。

参加本书编写工作的有任哲、房红征、李益民、车进辉。另外, 刘泰辉博士对于本书的章节及内容的安排提出的良好意见和建议, 也使本书增色不少。

将信息量如此巨大的 MFC 在一本 50 万字的教学用书中进行介绍, 对水平有限的作者来说实在是一件难事, 所以在此恳请读者对书中的缺点和错误提出批评和指正。同

时,也希望阅读本书的读者或使用本书作为教材的教师能与作者共同探讨,以期对本书作进一步的改进。欢迎读者及时与作者沟通。

通信地址: 北华大学计算机科学技术学院

邮编: 132021

电子信箱: renzhe71@sina.com.cn

祝各位教学有成!

作者

2007 年 7 月

第1版 前言

PREFACE

这些年来, Windows 一直是一个主流操作系统, 市场上对开发 Windows 应用程序的需求与日俱增。随着 Windows 操作系统的发展, Windows 应用程序变得越来越复杂, 使得单纯使用 Windows API 来开发 Windows 应用程序变得越来越困难。于是, 一些可视化的 Windows 开发工具(如 Visual Basic、Delphi、C++ Builder 等)如雨后春笋般地涌现出来, 这无疑给 Windows 应用的开发带来了极大的方便, 同时也大大地提高了开发效率。但是, 由于这些开发工具对应用程序框架的代码封装得过于严密, 因而使得对 Windows 低层应用的开发几乎成为不可能。而 Microsoft 基础类库(Microsoft foundation classes, MFC)只是使用 C++ 类对 Windows API 进行了封装, 它一方面较好地屏蔽了 Windows 编程的复杂性, 使得 Windows 应用程序的设计变得简单起来, 另一方面仍然允许有经验的程序员使用 Windows 的底层功能开发高效的 Windows 应用程序, 再加上与 MFC 配套的开发工具 Visual C++ 提供的功能强大的各种 Wizard(向导), 使得 MFC 成为了一个强有力的 Windows 应用程序的开发工具。

目前, 各高等学校已经把 C/C++ 列为理工科专业学生的必修课程, 这门课程使学生初步了解和掌握了面向对象程序设计的思想和方法, 这无疑为学生学习和掌握 MFC 打下了良好的基础, 同时对 MFC 的学习, 除了可以使学生掌握 Windows 应用程序设计的基本方法之外, 又可以使学生更进一步深刻、全面地理解面向对象程序设计的思想, 从而把握程序设计方法的发展方向。

基于上述这些理由, 各个高校都在 MFC 的教学方面进行了积极探索。本书是作者积数年教学经验, 为高等学校理工科学生编写的一部应用 MFC 进行 Windows 应用程序设计的入门教材。

全书共有 16 章, 参考教学时数 50 学时。

本书的特点是遵循循序渐进的教学原则, 以较大的篇幅从 Windows SDK 程序入手, 在介绍 Windows 应用程序一般特点的基础上, 重点介绍 Windows 的消息机制和 Windows 程序的基本结构, 以 C++ 类对其进行模拟

封装使学生迅速建立起 MFC 应用程序框架的基本概念; 然后以精练的语言介绍 Windows 的图形图像处理、应用程序界面、资源、动态链接库、进程与线程管理等知识。最后, 为使学生对软件工程的发展有一个初步的了解又介绍了组件对象模型 (COM)、ActiveX 技术和数据库的基本常识。

为配合教学和学习, 本书为每一个知识点都配以必要的实例, 力求通过实例让读者掌握 MFC 程序设计的特点及方法, 所有实例都在 Windows 98 和 Visual C++ 6.0 环境中测试通过。并且, 为配合教师教学, 本书为使用本教材的教师免费提供源代码和电子教案。

本书在编写过程中得到了吉林大学张长海教授及其他教师的悉心指导, 在本书的内容、章节编排等方面提出了宝贵意见和建议, 在此对他们的帮助表示衷心的感谢。本书在编写中还参考了大量的相关文献, 并引用了其中的一些例题和内容, 在此也对这些文献的作者表示诚挚的谢意。

参加本书编写工作的有任哲、李益民、车进辉。由于作者水平有限, 恳请读者对书中的缺点和错误提出批评和指正, 欢迎读者与作者联系。

通信地址: 北华大学计算机科学技术学院

邮 编: 132021

电子信箱: renzhe71@sina.com.cn

作 者

2004 年 4 月

目 录

第 1 章 Windows 程序基础	1
1.1 Windows 应用程序的基本概念	1
1.1.1 一般概念	1
1.1.2 Windows 内核、API 和开发工具	7
1.2 Windows 的数据类型	8
1.2.1 Windows 的一个特殊数据类型——句柄	9
1.2.2 窗口类 WNDCLASS	13
1.2.3 Windows 函数的调用说明	14
1.3 窗口的创建和显示	15
1.4 事件、消息循环和窗口函数	18
1.5 Windows 应用程序的结构	20
1.5.1 主函数	20
1.5.2 窗口函数	22
1.5.3 Windows 系统、主函数、窗口函数之间的关系	23
1.6 Windows 程序代码重构	30
1.6.1 用函数封装消息处理代码	30
1.6.2 消息映射表	34
练习题	38
思考题	38
第 2 章 Windows 应用程序的类封装	39
2.1 应用程序主函数的 C++ 类封装	39
2.1.1 窗口类的声明	40
2.1.2 应用程序类的声明	41
2.1.3 主函数封装后的程序	42
2.2 派生类的应用	46
2.2.1 应用程序类的派生类	47

2.2.2 窗口类的派生类	49
2.3 窗口函数的封装——消息映射	53
2.3.1 消息处理函数的简单封装	54
2.3.2 消息映射	56
2.3.3 消息映射表的声明和实现	59
小结	62
练习题	62
思考题	63

第3章 MFC 应用程序框架	65
3.1 早期的应用程序框架及其 MFC 类	65
3.1.1 早期的应用程序框架	65
3.1.2 MFC 的窗口类	66
3.1.3 CWinApp 类	67
3.2 最简单的 MFC 程序实例	68
3.2.1 程序的编写	68
3.2.2 程序主函数的代码	70
3.3 应用程序的文档/视图结构	71
3.3.1 文档/视图结构的基本概念	71
3.3.2 单文档界面和多文档界面结构	73
3.4 文档类 CDocument 的派生类	74
3.5 视图类 CView 的派生类	75
3.6 窗口框架类 CFrameWnd 的派生类	76
3.7 文档模板类 CDocTemplate	78
3.8 应用程序类的派生类	79
3.8.1 应用程序类派生类的代码	79
3.8.2 程序员的主要工作	80
3.9 MFC 文档/视图应用程序框架中各个对象的关系	81
3.9.1 应用程序各对象创建的顺序	81
3.9.2 应用程序各对象之间的联系	82
3.9.3 文档/视图应用程序消息的传递	85
3.10 对象的动态创建	85
3.10.1 问题的提出与解决	85
3.10.2 类信息表及其声明和实现	90
3.10.3 对象类信息表的提取	91
小结	92
练习题	92
思考题	92

第 4 章 图形	93
4.1 DC 和 GDI	93
4.1.1 设备描述环境	93
4.1.2 Windows 的 GDI	94
4.2 CDC 类	94
4.3 CPen 类	96
4.4 CBrush 类	99
4.5 绘图模式	101
4.6 文本和 CFont 类	102
4.6.1 显示文本	102
4.6.2 字体和 CFont 类	105
4.7 CDC 的其他派生类	107
4.7.1 窗口用户区设备描述环境 CClientDC 类	107
4.7.2 图元文件设备描述环境 CMetaFileDC 类	108
小结	110
练习题	110
思考题	110
第 5 章 MFC 的通用类	111
5.1 简单数据类	111
5.1.1 点类 CPoint	111
5.1.2 矩形类 CRect	112
5.1.3 尺寸类 CSize	112
5.1.4 字符串类 CString	113
5.2 群体数据类	114
5.3 数据在文档/视图结构程序中的位置	116
小结	118
练习题	118
思考题	118
第 6 章 Windows 应用程序界面的设计	119
6.1 SDI 和 MDI 界面	119
6.2 SDI 界面窗口的样式	120
6.2.1 在 MFC AppWizard 中确定窗口样式	120
6.2.2 在函数 PreCreateWindow 中修改窗口的样式	121
6.2.3 可拆分为小窗口的 SDI 界面	125
6.3 拆分窗口的同步更新及更新效率	126
6.3.1 拆分窗口的同步更新	126

6.3.2 提高拆分窗口更新效率的方法	126
6.4 创建带有滚动条的窗口	131
小结	135
练习题	135
思考题	136
第7章 鼠标和键盘	137
7.1 处理鼠标消息	137
7.1.1 用户区鼠标消息	138
7.1.2 非用户区鼠标消息	142
7.1.3 鼠标消息的捕获	143
7.2 处理键盘消息	144
7.2.1 按键的虚拟码	144
7.2.2 一般按键消息	147
7.2.3 系统按键消息	148
7.3 应用程序窗口的焦点	149
小结	150
练习题	151
思考题	152
第8章 资源	153
8.1 资源和资源文件	153
8.1.1 什么是资源	153
8.1.2 资源头文件	154
8.1.3 资源描述文件	155
8.2 菜单	155
8.2.1 资源描述文件的菜单部分	155
8.2.2 编辑现有的菜单	157
8.2.3 自定义菜单	160
8.2.4 用菜单编辑器编辑菜单	162
8.2.5 菜单命令选项的动态修改	162
8.3 快捷键表	165
8.3.1 资源描述文件的快捷键表部分	165
8.3.2 使用快捷键编辑器编辑快捷键表	166
8.4 图标	166
8.4.1 制作图标	166
8.4.2 资源描述文件的图标部分	167
8.5 位图	168