



21st CENTURY
实用规划教材

21世纪全国应用型本科计算机案例型规划教材



Visual C++ 程序设计实用案例教程



rogramming

主 编 于永彦 王志坚
 娄渊胜 束玉琴



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国应用型本科计算机案例型规划教材

Visual C++程序设计实用案例教程

主 编 于永彦 王志坚
 娄渊胜 束玉琴



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书为 C++ 程序设计教材的高级程序设计部分, 主要讲述 Visual C++ 基础知识与基本应用, 包括 OOP 概念、Windows 程序原理、MFC 类库及其应用程序框架、数据库应用程序设计、网络应用程序设计和多媒体应用程序设计。全书仍然以一个实用的“简易学生管理系统”为研究载体, 针对每一个具体应用问题设计一个“子工程模型”, 从基本概念入手, 循序渐进, 既有必要的理论知识的铺垫, 又重点突出了对读者实践技能的培养与训练。

本书适用于理工类大中、专院校的 C++ 程序设计课程, 也可供程序设计爱好者和工程技术人员参考使用。

图书在版编目(CIP)数据

Visual C++ 程序设计实用案例教程/于永彦, 王志坚, 姜渊胜, 束玉琴主编. —北京: 北京大学出版社, 2010.2
(21 世纪全国应用型本科计算机案例型规划教材)

ISBN 978-7-301-16597-3

I. V… II. ①于…②王…③姜…④束… 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(2010)第 009482 号

书 名: Visual C++ 程序设计实用案例教程

著作责任者: 于永彦 王志坚 姜渊胜 束玉琴 主编

策划编辑: 孙哲伟 李 虎

责任编辑: 孙哲伟

标准书号: ISBN 978-7-301-16597-3/TP · 1074

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电子邮箱: pup_6@163.com

印 刷 者: 北京大学印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 21.5 印张 492 千字

2010 年 2 月第 1 版 2010 年 2 月第 1 次印刷

定 价: 32.00 元

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

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

21 世纪全国应用型本科计算机案例型规划教材

专家编审委员会

(按姓名拼音顺序)

主 任 刘瑞挺

副主任 陈 钟 蒋宗礼

委 员 陈代武 胡巧多 黄贤英

江 红 李 建 娄国焕

马秀峰 祁亨年 王联国

汪新民 谢安俊 解 凯

徐 苏 徐亚平 宣兆成

姚喜妍 于永彦 张荣梅

信息技术的案例型教材建设

(代丛书序)

刘瑞挺

北京大学出版社第六事业部在 2005 年组织编写了《21 世纪全国应用型本科计算机系列实用规划教材》，至今已出版了 50 多种。这些教材出版后，在全国高校引起热烈反响，可谓初战告捷。这使北京大学出版社的计算机教材市场规模迅速扩大，编辑队伍茁壮成长，经济效益明显增强，与各类高校师生的关系更加密切。

2008 年 1 月北京大学出版社第六事业部在北京召开了“21 世纪全国应用型本科计算机案例型教材建设和教学研讨会”。这次会议为编写案例型教材做了深入的探讨和具体的部署，制定了详细的编写目的、丛书特色、内容要求和风格规范。在内容上强调面向应用、能力驱动、精选案例、严把质量；在风格上力求文字精练、脉络清晰、图表明快、版式新颖。这次会议吹响了提高教材质量第二战役的进军号。

案例型教材真能提高教学的质量吗？

是的。著名法国哲学家、数学家勒内·笛卡儿(Rene Descartes, 1596—1650)说得好：“由一个例子的考察，我们可以抽出一条规律。(From the consideration of an example we can form a rule.)”事实上，他发明的直角坐标系，正是通过生活实例而得到的灵感。据说是在 1619 年夏天，笛卡儿因病住进医院。中午他躺在病床上，苦苦思索一个数学问题时，忽然看到天花板上有一只苍蝇飞来飞去。当时天花板是用木条做成正方形的格子。笛卡儿发现，要说出这只苍蝇在天花板上的位置，只需说出苍蝇在天花板上的第几行和第几列。当苍蝇落在第四行、第五列的那个正方形时，可以用(4, 5)来表示这个位置……由此他联想到可用类似的办法来描述一个点在平面上的位置。他高兴地跳下床，喊着“我找到了，找到了”，然而不小心把国际象棋撒了一地。当他的目光落到棋盘上时，又兴奋地一拍大腿：“对，对，就是这个图”。笛卡儿锲而不舍的毅力，苦思冥想的钻研，使他开创了解析几何的新纪元。千百年来，代数与几何，井水不犯河水。17 世纪后，数学突飞猛进的发展，在很大程度上归功于笛卡儿坐标系和解析几何学的创立。

这个故事，听起来与阿基米德在浴池洗澡而发现浮力原理，牛顿在苹果树下遇到苹果落到头上而发现万有引力定律，确有异曲同工之妙。这就证明，一个好的例子往往能激发灵感，由特殊到一般，联想出普遍的规律，即所谓的“一叶知秋”、“见微知著”的意思。

回顾计算机发明的历史，每一台机器、每一颗芯片、每一种操作系统、每一类编程语言、每一个算法、每一套软件、每一款外部设备，无不像闪光的珍珠串在一起。每个案例都闪烁着智慧的火花，是创新思想不竭的源泉。在计算机科学技术领域，这样的案例就像大海边上的贝壳，俯拾皆是。

事实上，案例研究(Case Study)是现代科学广泛使用的一种方法。Case 包含的意义很广：包括 Example 例子，Instance 事例、示例，Actual State 实际状况，Circumstance 情况、事件、境遇，甚至 Project 项目、工程等。

我们知道在计算机的科学术语中，很多是直接来自日常生活的。例如 Computer 一词早在 1646 年就出现于古代英文字典中，但当时它的意义不是“计算机”而是“计算工人”，

即专门从事简单计算的工人。同理, Printer 当时也是“印刷工人”而不是“打印机”。正是由于这些“计算工人”和“印刷工人”常出现计算错误和印刷错误,才激发查尔斯·巴贝奇(Charles Babbage, 1791—1871)设计了差分机和分析机,这是最早的专用计算机和通用计算机。这位英国剑桥大学数学教授、机械设计专家、经济学家和哲学家是国际公认的“计算机之父”。

20 世纪 40 年代,人们还用 Calculator 表示计算机。到电子计算机出现后,才用 Computer 表示计算机。此外,硬件(Hardware)和软件(Software)来自销售人员。总线(Bus)就是公共汽车或大巴,故障和排除故障源自格瑞斯·霍普(Grace Hopper, 1906—1992)发现的“飞蛾子”(Bug)和“抓蛾子”或“抓虫子”(Debug)。其他如鼠标、菜单……不胜枚举。至于哲学家进餐问题,理发师睡觉问题更是操作系统文化中脍炙人口的经典。

以计算机为核心的信息技术,从一开始就与应用紧密结合。例如,ENIAC 用于弹道曲线的计算,ARPANET 用于资源共享以及核战争时的可靠通信。即使是非常抽象的图灵机模型,也受到二战时图灵博士破译纳粹密码工作的影响。

在信息技术中,既有许多成功的案例,也有不少失败的案例;既有先成功而后失败的案例,也有先失败而后成功的案例。好好研究它们的成功经验和失败教训,对于编写案例型教材有重要的意义。

我国正在实现中华民族的伟大复兴,教育是民族振兴的基石。改革开放 30 年来,我国高等教育在数量上、规模上已有相当的发展。当前的重要任务是提高培养人才的质量,必须从学科知识的灌输转变为素质与能力的培养。应当指出,大学课堂在高新技术的武装下,利用 PPT 进行的“高速灌输”、“翻页宣科”有愈演愈烈的趋势,我们不能容忍用“技术”绑架教学,而是让教学工作乘信息技术的东风自由地飞翔。

本系列教材的编写,以学生就业所需的专业知识和操作技能为着眼点,在适度的基础知识与理论体系覆盖下,突出应用型、技能型教学的实用性和可操作性,强化案例教学。本套教材将会有机融入大量最新的示例、实例以及操作性较强的案例,力求提高教材的趣味性和实用性,打破传统教材自身知识框架的封闭性,强化实际操作训练,使本系列教材做到“教师易教,学生乐学,技能实用”。有了广阔的应用背景,再造计算机案例型教材就有了基础。

我相信北京大学出版社在全国各地高校教师的积极支持下,精心设计,严格把关,一定能够建设出一批符合计算机应用型人才培养模式的、以案例型为创新点和兴奋点的精品教材,并且通过一体化设计、实现多种媒体有机结合的立体化教材,为各门计算机课程配齐电子教案、学习指导、习题解答、课程设计等辅导资料。让我们用锲而不舍的毅力,勤奋好学的钻研,向着共同的目标努力吧!

刘瑞挺教授 本系列教材编写指导委员会主任、全国高等院校计算机基础教育研究会副会长、中国计算机学会普及工作委员会顾问、教育部考试中心全国计算机应用技术证书考试委员会副主任、全国计算机等级考试顾问。曾任教育部理科计算机科学教学指导委员会委员、中国计算机学会教育培训委员会副主任。PC Magazine《个人电脑》总编辑、CHIP《新电脑》总顾问、清华大学《计算机教育》总策划。

前 言

一直以来, C++程序设计始终是计算机、通信等相关专业的主修课程, 对于诸如数据结构、操作系统、编译原理等后续课程的学习、专业知识结构的完整构建具有一锤定音的作用。系统掌握 C++语言的基础理论与实用技术, 已成为大多数学生的必修课程与必备技能。目前的一些典型的 C++编译工具, 如 Visual C++、Borland C++等, 虽然较好地实现了功能抽象和数据抽象的有机统一, 一定程度上实现了可视化编程, 但也因此使得所要学习的知识体系变得日益庞大, 往往费了大量时间而达不到预期的学习效果。

古语云: “实践出真知。”在北京大学出版社领导的大力支持下, 编者邀请了河海大学、江苏信息职业技术学院、淮阴工学院等三所高校的部分骨干教师, 联合编写了 C++程序设计教材, 拟包括《C++程序设计基础案例教程》、《Visual C++程序设计实用案例教程》、《C++学习指导》和《C++实践教程》四个分册, 并已于 2009 年 1 月份先期出版了《C++程序设计基础案例教程》一书。从兄弟院校采用情况及反馈意见来看, 既对教材特色和内容组织给予了充分肯定, 也对诸如教材定位、素材取舍与案例编排上提出了宝贵的修改意见, 感激之余, 不免惴惴难安。为此, 本次在组织编写《Visual C++程序设计实用案例教程》过程中, 重新修订了原讲义, 对全书结构作了较大调整, 对实践案例作了重新筛选, 力求数量适当、难度适宜、覆盖面适中。

一定程度上可将本书看作为《C++程序设计基础案例教程》的姊妹篇, 是对前一本教材的深化与升华。第 1 章首先带领读者扼要回顾了面向对象程序设计的基础理论与基本技术, 为后续章节的学习进行有效铺垫。第 2、第 3 章则从 API、MFC 而各方面介绍了设计 Windows 程序的基本原则与基本方法, 所选案例较为经典, 涉及面较广, 需要读者深入领会。第 4、第 5、第 6 章分别基于对话框、文档/视图、数据库等三种技术, 从三个角度各自设计了一个完整的“简易管理系统”, 它们功能上保持一致, 而实现方法与具体过程各有千秋, 其目的就是在对比中学习三种设计方法的优劣性, 为读者的进一步深造提供参考文献。在第 7、第 8 章, 我们分别就 Visual C++在网络和多媒体方面的应用作了一定的探讨, 重点是通过若干个典型案例加以分析, 目的是抛砖引玉, 引导读者加强自学意识。

本书的编写得到了很多热心人士的帮助与支持。首先感谢河海大学计算机与信息工程学院、淮阴工学院计算机工程系、江苏信息职业技术学院的领导及各位同仁, 感谢他们为全书的整体性构思提供了许多建设性的建议, 感谢他们为本书提供了试用平台, 并提出了许多宝贵的修改建议。还有其他为本书进行过文字校对、编辑排版的老师、同学, 此处不再一一列出, 谨一并表示最诚挚的谢意。

虽然编者主观上做了最大的努力, 但由于本身的水平有限, 加上时间仓促, 教学改革的力度又较大, 难免存在一些不足。“他山之石可以攻玉”, 真诚地希望使用或阅读本书的读者给予批评指正, 不吝赐教。

作 者

2009 年 10 月 6 日 于无锡

参 考 文 献

- [1] Microsoft. MFC 与 Windows 编程[M]. 2 版. 影印版. 北京: 北京大学出版社, 2000.
- [2] [美]Charles Petzold. Windows 编程[M]. 5 版. 影印版. 北京: 北京大学出版社, 2002.
- [3] [美]K.Li and M.Wu. Effective GUI Testing Automation: Developing an Automated GUI Testing Tool[M]. 王轶昆, 等译. 北京: 电子工业出版社, 2005.
- [4] 张宏林. Visual C++数字图像模式识别技术及工程实践[M]. 北京: 人民邮电出版社, 2004.
- [5] 刘长明, 等. Visual C++实践与提高——多媒体篇[M]. 北京: 中国铁道出版社, 2003.
- [6] [美]David J. Kruglinski. Visual C++技术内幕[M], 潘爱民, 王国印译. 北京: 清华大学出版社, 1999.
- [7] 魏亮, 李春葆. Visual C++程序设计例学与实践[M]. 北京: 清华大学出版社, 2006.
- [8] 刘瑞, 吴跃进, 王宗越. Visual C++项目开发实用案例[M]. 北京: 科学出版社, 2006.
- [9] 孙鑫, 余安萍. VC++深入详解[M]. 北京: 电子工业出版社, 2006.
- [10] 陈清华, 等. Visual C++课程设计案例精选与编程指导[M]. 南京: 东南大学出版社, 2004.
- [11] 严华峰, 等. Visual C++课程设计案例精编 [M]. 2 版. 北京: 中国水利水电出版社, 2004.
- [12] 周鸣扬. Visual C++界面编程技术[M]. 北京: 北京希望电子出版社, 2003.
- [13] 原奕, 等. Visual C++实践与提高——数据库开发与工程应用篇[M]. 北京: 中国铁道出版社, 2005.
- [14] 求是科技. Visual C++ 6.0 数据库开发技术与工程实践[M]. 北京: 人民邮电出版社, 2004.
- [15] 李鲲鹏. Visual C++打印编程技术与工程实践[M]. 北京: 人民邮电出版社, 2003.
- [16] 李闽溟, 等. Visual C++ 6.0 数据库系统开发实例导航[M]. 北京: 人民邮电出版社, 2004.
- [17] 张越, 等. Visual C++网络程序设计实例详解[M]. 北京: 人民邮电出版社, 2006.
- [18] 陈坚, 等. Visual C++网络高级编程[M]. 北京: 人民邮电出版社, 2003.
- [19] 郎锐, 等. Visual C++网络通信程序开发指南[M]. 北京: 机械工业出版社, 2004.
- [20] 周奇. SQL Server 2005 数据库基础及应用技术教程与实训[M]. 北京: 北京大学出版社, 2008.
- [21] 罗伟坚. Visual C++经典游戏程序设计[M]. 北京: 人民邮电出版社, 2006.
- [22] 向世明. Visual C++数字图像与图形处理[M]. 北京: 电子工业出版社, 2005.
- [23] 汪晓华, 等. Visual C++网络通信协议分析与应用实现[M]. 北京: 人民邮电出版社, 2004.
- [24] [美]B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction, 3rd Ed[M]. 张国印, 李健利译. 北京: 电子工业出版社, 2004.
- [25] 朱娜敏, 魏宗寿, 李红. 精通 Windows 程序设计——基于 Visual C++实现[M]. 北京: 人民邮电出版社, 2009.
- [26] 刘腾红, 屈振新. Windows 程序设计技术[M]. 北京: 清华大学出版社, 2004.

目 录

第 1 章 面向对象程序设计 1

1.1 概述	2
1.2 类与对象概念	3
1.2.1 定义类	3
1.2.2 声明对象	5
1.2.3 构造函数和析构函数	6
1.2.4 类的组合	8
1.3 继承与派生	9
1.3.1 派生类声明	10
1.3.2 派生类构造函数和析构 函数	10
1.3.3 多继承	11
1.4 多态性	13
1.4.1 运算符重载	13
1.4.2 虚函数	16
1.4.3 纯虚函数与抽象类	16
1.4.4 类模板	17
本章总结	19
习题	20

第 2 章 Windows 程序原理 22

2.1 数据类型	23
2.2 事件驱动的程序设计模式	24
2.3 Windows 消息	25
2.3.1 消息与消息结构	25
2.3.2 键盘消息	27
2.3.3 鼠标消息	29
2.4 消息队列	29
2.4.1 进程与线程	29
2.4.2 消息队列	30
2.4.3 将消息发送到队列中	31
2.4.4 队列状态标志	32
2.4.5 从队列中提取消息	32

2.5 Windows 程序结构	34
2.5.1 程序组成	35
2.5.2 WinMain 函数	35
2.5.3 WinProc 函数	40
2.6 案例：使用 API 设计简易计算器	41
本章总结	49
习题	50

第 3 章 MFC 基础 53

3.1 概述	54
3.1.1 AFX 技术	54
3.1.2 MFC 类库	56
3.2 MFC 特性	56
3.3 MFC 类结构	58
3.3.1 CObject 根类	58
3.3.2 应用程序体系结构类	67
3.3.3 可视化控件类	70
3.4 基于 MFC 创建 Win32 程序	71
3.5 使用 MFC 向导创建 Windows 程序	73
3.5.1 Visual C++ 应用程序向导	73
3.5.2 创建 MFC 应用程序的一般 步骤	74
3.6 MFC 应用程序结构	77
3.6.1 程序结构	77
3.6.2 项目管理	78
3.7 MFC 程序内部机制	79
3.8 MFC 消息映射	85
3.8.1 映射与消息映射	85
3.8.2 消息宏	86
3.8.3 消息传动路由	90
3.8.4 消息映射实例分析	90
本章总结	91
习题	92

第4章 基于对话框的程序设计 94

4.1 基于对话框的程序设计步骤 95	
4.1.1 创建对话框工程 96	
4.1.2 添加对话框模板 96	
4.1.3 设置对话框模板属性 97	
4.1.4 改变对话框模板图标 98	
4.1.5 对话框设计内容 99	
4.2 登录对话框设计 99	
4.2.1 对话框模板设计 99	
4.2.2 设计登录对话框类 103	
4.2.3 在启动主对话框前运行 “用户登录”对话框 109	
4.3 设计添加信息对话框 110	
4.3.1 设计 IDD_Add 对话框模板 110	
4.3.2 设计 IDD_Add 对话框类 112	
4.4 设计修改信息对话框 117	
4.4.1 设计 IDD_Edit 对话框模板 117	
4.4.2 设计修改信息对话框类 117	
4.5 设计主对话框 118	
4.5.1 设计 IDD_MainDlg 对话框 模板 118	
4.5.2 完善 IDD_MainDlg 对话框类 122	
本章总结 126	
习题 126	

第5章 基于文档/视图的程序设计 128

5.1 应用 MFC 向导创建文档/视图 框架 129	
5.2 类结构与框架分析 131	
5.2.1 应用程序类 131	
5.2.2 文档模板 134	
5.2.3 文档类 CDocument 139	
5.2.4 视图类 CView 140	
5.2.5 框架类 142	
5.2.6 文档、文档模板、视图类和 框架类的关系 146	
5.3 视图分割 147	

5.3.1 动态分割 148	
5.3.2 静态分割 149	
5.4 将对话框与文档/视图关联 150	
5.4.1 对话框模板复用 150	
5.4.2 创建主对话框 152	
5.4.3 将对话框与视图关联 155	
5.4.4 文本显示 155	
5.5 菜单设计 156	
5.5.1 菜单类 CMenu 157	
5.5.2 菜单资源设计 158	
5.5.3 建立菜单消息映射 160	
5.5.4 更新命令用户接口消息 161	
5.6 定制序列化 163	
5.6.1 创建学生类 CStudent 164	
5.6.2 定制类的串行序列化 166	
5.6.3 文档 CEx5_1Doc 序列化 167	
5.6.4 修改菜单功能 168	
本章总结 170	
习题 170	

第6章 数据库应用程序设计 172

6.1 数据库基础 173	
6.1.1 基本概念 174	
6.1.2 数据库系统体系结构 176	
6.1.3 管理信息系统 178	
6.2 创建数据库 179	
6.2.1 创建空数据库 179	
6.2.2 创建数据表 179	
6.3 注册 ODBC 数据源名 182	
6.3.1 ODBC 简介 182	
6.3.2 手动注册 DSN 183	
6.3.3 自动注册 DSN 184	
6.4 基于 MFC 封装 ODBC 186	
6.4.1 CDatabase 类 187	
6.4.2 CRecordset 类 188	
6.5 基于 ODBC 设计数据库应用程序 191	
6.5.1 创建应用程序框架 191	
6.5.2 设计 CMyDB2Set 类 192	

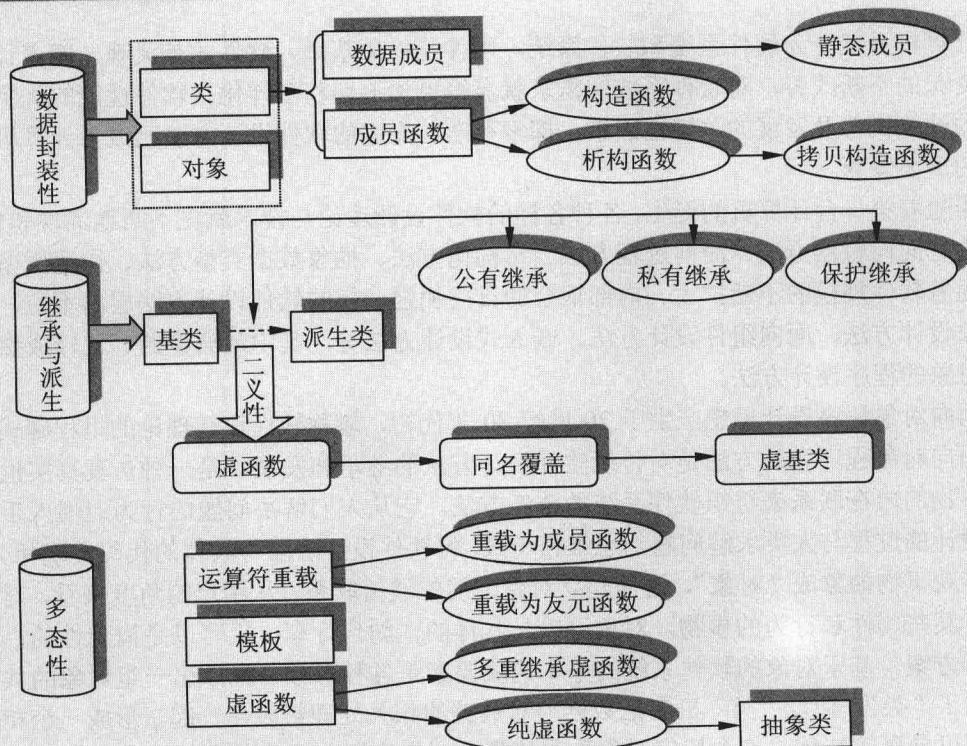
6.5.3 设计 CMyDB2View 类	195	7.5.5 设计 CServFileView 类	258
6.5.4 设计事件响应代码	200	本章总结	268
6.6 DAO 编程基础	204	习题	269
6.6.1 DAO 简介	204	第 8 章 多媒体应用程序设计	271
6.6.2 DAO 对象模型	205	8.1 概述	272
6.6.3 封装 DAO	205	8.1.1 多媒体概念	272
6.6.4 基于 DAO 设计数据库应用 程序	207	8.1.2 图形设备接口	275
6.7 ADO 概述	218	8.1.3 映像模式	277
本章总结	224	8.1.4 三基色与调色板	280
习题	224	8.1.5 MFC 对 GDI 的封装	281
第 7 章 网络应用程序设计	226	8.2 文本处理	285
7.1 概述	227	8.2.1 选择文本字体	285
7.1.1 网络通信	227	8.2.2 设置文本属性	287
7.1.2 网络编程模式	229	8.2.3 文本输出	288
7.2 Windows Sockets 编程	231	8.3 案例一：控制字体显示	289
7.2.1 概述	231	8.4 图形绘制	289
7.2.2 CAsyncSocket 类	232	8.4.1 绘图函数	290
7.2.3 CSocket 类	234	8.4.2 绘图模式	293
7.3 案例一：一个点对点的聊天程序	235	8.5 案例二：绘制橡皮筋直线	294
7.3.1 创建客户端应用程序	235	8.6 位图操作	296
7.3.2 创建服务器端程序 SocketServer	240	8.6.1 与设备有关的位图 DDB	296
7.4 WinInet 编程	245	8.6.2 与设备无关的位图 DIB	298
7.4.1 WinInet API 简介	246	8.7 案例三：显示 DIB 位图	307
7.4.2 认识 MFC WinInet 类	247	8.8 音频处理	309
7.4.3 使用 WinInet 类编程的基本 操作	249	8.8.1 RIFF 格式	309
7.5 案例二：FTP 客户端程序设计	250	8.8.2 WAVE 文件格式	310
7.5.1 创建应用程序的 MFC 框架	250	8.8.3 声音播放	311
7.5.2 设计 CFtpInfoView 类	251	8.8.4 MCI 基础	315
7.5.3 完善 CFtpClientApp 类	253	8.9 案例四：简易音频播放器	320
7.5.4 完善 CMainFrame 类	254	本章总结	325
		习题	326
		参考文献	327

面向对象程序设计

教学目标

本章主要介绍面向对象程序设计的基本理论与基本技术,包括类与对象的定义、构造派生类、二义性、虚基类、运算符重载、虚函数与抽象类、类模板等。通过学习,要求掌握类的封装性、继承性与多态性的基本概念及应用;熟悉二义性问题及其解决办法;了解什么是虚基类、虚函数与抽象类;能运用面向对象的程序设计方法编写简单的应用程序。

知识结构





引言

C++语言是一种面向对象程序设计(Object Oriented Programming, OOP)语言,它汲取了传统结构化程序设计(Structured Programming, SP)的精华,尝试对程序结构进行“统一化、模块化”处理,其实质是一种更高层次上的结构化。

在面向对象程序设计理论体系中,“对象”是构建软件系统的基本元素,而从一组对象中抽象出来的共性就形成了“类”,一个具体的对象称为类的一个“实例”。类的多个实例虽然具有不同的名称,但具有相同的属性和行为,只是具体的取值范围可能不同而已。例如,某个“大学生”是一个对象,而“所有大学生”则构成“大学生类”。每个大学生具有相同的信息结构(姓名、性别、专业等),但实际取值却可能是不同的。

从数据存储角度看,类可看做是用户自定义的数据类型,一个对象就是该类型的一个变量;而从程序设计角度看,类把数据与处理封装在一起,具有继承性、多态性等特点。

本章将以 Visual C++ 6.0 英文版为系统平台,简要介绍面向对象程序设计的基本理论与基本技术。

1.1 概 述

计算机系统分为硬件系统和软件系统,硬件是物质基础,软件才是灵魂。随着计算机硬件资源的不断改善,对软件系统的需求就显得更加迫切和多样化。软件性能的优劣已成为制约计算机普及发展的关键性因素,探索科学、先进的软件设计方法已成为计算机基础研究的重要课题。

伴随着第一台计算机的诞生,各种各样的软件设计理论与技术就在不断地涌现和完善,传统的有面向机器方法、面向过程方法、结构化方法、抽象数据类型方法、快速原型方法等。随着软件规模的不断扩大、内部耦合的日益加强,新的软件设计方法层出不穷,如面向对象设计方法、面向组件设计方法、嵌入式设计方法等,其中最典型的、比较成熟的是面向对象的程序设计方法。

面向对象程序设计思想产生于 20 世纪 80 年代初,堪称软件设计理论的里程碑式的革命。面向对象程序设计方法是对传统软件设计方法的继承和发展,是一种仿真真实世界中各种事物的内在联系来组织软件系统的全新方法,它从人们解决问题的行为习惯入手,使软件设计的过程与人类求解问题的思维相一致,将软件设计阶段与具体的代码实现相分离。它将客观事物抽象成“对象”,进而建立对象之间的通信联系。从用户的角度来看,对象就是具有某些属性和行为的事物,对象可以是具体的,如自行车;也可以是概念性的,如车辆通行方案。基于对象的软件系统的基本构成要素是各种对象,抽象出一组对象的共性,就形成了“类”,通过“类”机制把数据与对数据的相关处理封装在一起,形成一个相互依存、不可分离的整体,一个特定对象称为该类的一个实例。面向对象的程序设计方法正是以类为设计核心的,实现了模块内信息的有效封装,隐藏了对象的具体细节,使模块间关系更简单、数据更安全,同时提高了工作效率和程序的可靠性,方便了大型复杂软件系统的前期调试和后期维护。

描述面向对象方法所涉及的类、对象、继承、多态等基本概念的程序设计语言,称为面向对象程序设计语言(Object-Oriented Programming Language, OOPL)。OOPL 属于高级语言,可以更直接、更系统地描述各类对象及其相互关系。

按功能侧重程度的不同,可把面向对象程序设计语言分为纯面向对象程序设计语言和混合型面向对象程序设计语言两大类。在纯面向对象程序设计语言中,几乎所有的语言成分都是对象,人们不能使用这种语言去进行面向过程程序设计方法和结构化设计方法的软件设计,这方面的代表是 Java 语言。而混合型面向对象程序设计语言既支持面向对象设计方法,也支持面向过程设计方法和结构化设计方法,大部分面向对象程序设计语言都属于这种类型,最典型的是 C++语言。

C++作为一种应用最广泛的面向对象程序设计语言,具有类、对象、消息等概念,全面支持面向对象技术的抽象性、封装性、继承性和多态性特征。

- (1) 抽象能力强,函数模板和类模板提供了更高级别的抽象。
- (2) 提供强大而特有的多继承机制,表达能力强。
- (3) 支持运算符重载,对象的运算更易表达且表达更加自然。
- (4) 支持高效的内存管理,提供自动和人工回收两种方式。
- (5) 是一种强类型语言,编译阶段就能发现程序潜在错误,不会将错误带到运行阶段。
- (6) 加强了异常处理,支持对象类型的异常处理。
- (7) 支持命名空间,更加有利于大型软件工程项目。
- (8) C++代码质量高、运行速度快、可移植性好。

由于 C++继承并扩展了 C 语言,增加了支持面向对象技术的语言成分。凭借 C 语言广泛的用户基础,C++语言一经推出后,迅速获得了商业上的成功。迄今为止,C++语言已拥有很多实用版本,如 Turbo C++、Borland C++、MS C++、Visual C++等。其中,Visual C++是目前最流行的 C++语言开发环境,目前主要版本是 6.0 版,本书所有例子都是在英文版 Visual C++ 6.0 环境下编译和调试的。

1.2 类与对象概念

类与对象是面向对象程序设计思想的理论基石。类是一组对象共有特性的抽象,而某个特定对象就称为所属类的一个实例。类机制实现了数据与操作的有效封装,保障了程序模块之间的功能独立性与数据安全性。

1.2.1 定义类

类定义的一般形式如下:

```
class <类名>
{
private:
    私有数据成员和成员函数
protected :
    受保护数据成员和成员函数
public:
```

公有数据成员和成员函数

};

其中，class 是关键字；<类名>表示所要定义类名称；花括号中为类的定义实体，包括数据成员和成员函数。private、protected 和 public 关键字用来说明类成员的访问属性，具体涵义见表 1-1。

表 1-1 类成员访问属性

访问属性	涵义
public	公有成员，表示可被本类成员或外部程序中声明的该类对象所访问
private	私有成员，表示只可被本类成员访问，外部程序无法直接访问。这也是默认属性
protected	受保护成员，与私有成员类似，但与派生类稍有区别

【例 1.1】 定义一个 Person 类，用于管理学生基本信息。

首先启动 Visual C++ 6.0 系统，选择 File/New 菜单项。

选择 Files 选项卡，选择 C/C++ Header File 选项，表示新建头文件(.h 文件)。在右侧的 File 文本框中输入文件名，例如此处输入 MyClassObject，并在 Location 文本框中设置文件存储位置，如图 1.1 所示。

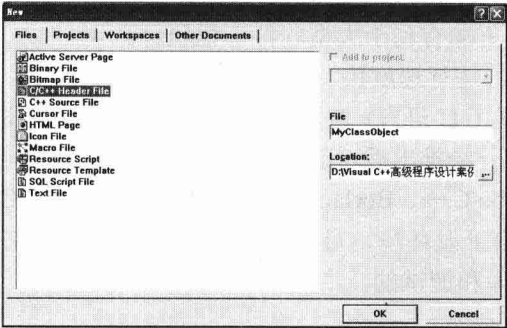


图 1.1 新建.h 文件

最后单击 OK 按钮，进入代码编辑界面，可输入下面的代码：

```
struct Per
{
    int Per_Number;           //编号
    char Per_Name[9];         //姓名
    bool Per_Sex;             //性别
};

class Person
{
private:
    Per p[10];                //10 条学生信息
    int Size;                 //实际学生信息数
public:
    void Insert(Per Pertemp);  //插入 Pertemp 学生信息
    void Delete(int tempNumber); //删除指定编号为 temp 的学生信息
```

```

int GetSum();           //返回实际学生信息数
Per FindNode(int tempNumber); //按编号查找
void Display();         //显示信息
};

```

这里, 首先定义了一个结构类型 **Per**, 再定义类 **Person**, 其中声明了一个 **Per** 类型的私有数据成员 **p[10]**, 一个 **int** 型变量 **Size**, 以及 **Insert(Per Pertemp)** 等 5 个成员函数。

选择 **File/Save** 菜单项, 将上述代码保存到 **MyClassObject.h** 文件中。

上面的 5 个成员函数提供了 **Person** 类的外部接口。注意, 此处类体内仅说明了成员函数的原型, 并没有定义具体代码, 所以还不能实现任何功能。为了体现类数据封装性, 一般把成员函数的具体实现放在类定义体之外的一个同名实现文件(.cpp)中。

与上述步骤类似, 在图 1.1 中选择 **C++ Source File** 选项, 创建 **MyClassObject.cpp** 文件。进入代码编辑界面后, 即可定义成员函数的具体实现代码, 格式如下:

```

返回类型 类名:: 成员函数名(形参表)
{
    函数体;
}

```

其中, “**::**” 称为类作用域运算符, 用于指定成员函数属于某个类, 例如:

```

void Person::Display()
{
    for(int i=0;i<Size;i++)
    {
        cout<<p[i].Per_Number;
        cout<<p[i].Per_Name;
        cout<<(p[i].Per_Sex==1?"男":"女")<<endl;
    }
}

```

注意以下几点。

(1) 数据成员一般为 **private**, 不能嵌套定义, 且不能直接初始化。

(2) 成员函数一般为 **public**, 如果代码段较短小, 可直接放在类体内, 类似于 **inline** 函数。

1.2.2 声明对象

定义了一个类只是创建了一种新的数据类型, 只有声明了对象才真正拥有了存储空间。可以在定义类的同时声明对象。例如下面的代码:

```

class Person
{
private:
    Per p[10];
    int Size;
public:
    void Insert(Per Pertemp);
    void Delete(int tempNumber);
}

```

```
int GetSum();
Per FindNode(int tempNumber);
void Display();
}p1; //同时声明了对象 p1
```

也可以在定义类后再声明对象，例如下面声明了两个 **Person** 对象：

```
Person p2,p3;
```

一旦声明了一个类的对象，就可以用“.”运算符来访问类的 **public** 成员，例如：

```
p2.Display();
```

C++语言允许同类对象之间相互赋值，例如：

```
p1= p2;
```

若一组对象同属于某一个类，则构成一个对象数组。声明对象数组的格式如下，例如：

```
Person p[10];
```

声明了对象数组之后，就可以引用其数组元素了，例如：

```
cout<<p[1].Display();
```

1.2.3 构造函数和析构函数

任何一个类都拥有两个特殊的成员函数：构造函数和析构函数。这两个函数既可以显式定义，也可以隐含调用。

1. 构造函数

构造函数用于初始化对象，否则对象的初始值将是不确定的。构造函数的定义格式如下：

```
<类名>::<类名>(<参数表>)
{
    <函数体>
}
```

构造函数与类同名，一般由系统在创建对象时自动调用。

例如，下面是 **Person** 类的构造函数：

```
class Person
{
private:
    Per p[10];
    int Size;
public:
    Person(int s){Size=s;} //构造函数
}p1;
```

定义了构造函数后，在定义对象时可以将参数传递给构造函数来初始化对象。例如：

```
Person p(10);
```