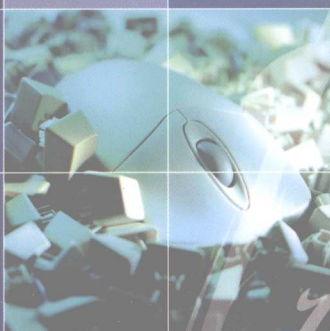
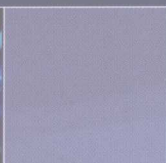




21st CENTURY
实用规划教材

21世纪全国应用型本科计算机案例型规划教材

C++面向对象与 Visual C++程序设计案例教程



rogramming

主 编 黄贤英 刘恒洋



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

C++面向对象与 Visual C++ 程序设计 案例教程

主 编 黄贤英 刘恒洋

副主编 谢丽明 吴志强

参 编 谭 智 李 冰



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书综合了 C++ 基础知识、Windows 程序设计和 MFC 编程基础，精选各个常用的知识点。贯穿全书的应用案例是一个完整的以 Visual Studio 2005 为开发平台的点对点通信程序。

本书适用于高等院校学生，既可以作为单独的教材使用，也可以作为参考教材，供学生在“C++面向对象程序设计”实验和课程设计中使用，也可供工程技术人员参考使用。阅读本书需要有一定的 C 语言基础。

图书在版编目(CIP)数据

C++面向对象与 Visual C++程序设计案例教程/黄贤英, 刘恒洋主编. —北京: 北京大学出版社, 2009.3
(21 世纪全国应用型本科计算机案例型规划教材)

ISBN 978-7-301-14504-3

I. C… II. ①黄…②刘… 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(2009)第 017303 号

书 名: C++面向对象与 Visual C++程序设计案例教程

著作责任者: 黄贤英 刘恒洋 主编

策划编辑: 李 虎 孙哲伟

责任编辑: 孙哲伟

标准书号: ISBN 978-7-301-14504-3/TP·0985

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: 北京飞达印刷有限责任公司

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 22.75 印张 525 千字

2009 年 3 月第 1 版 2009 年 3 月第 1 次印刷

定 价: 35.00 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岸边的贝壳，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,强化实际操作的训练,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

C++是一种应用广泛的面向对象的程序设计语言，MFC 是微软开发 Windows 应用程序的主流框架。全书以 C++语法、Windows 程序设计、MFC 的理论讲述为基础，精选各种常见语法结构，以 Visual Studio 2005 为开发平台，以提高学生应用开发能力为出发点，以应用案例为主线。读者通过本书的学习，能掌握 Windows 应用程序的特点、数据和函数的组织方法和结构；深刻理解面向对象程序设计的思想、方法和技巧，从而把握程序设计方法的发展方向，提高程序设计能力和应用系统的开发能力。

本书的主要特色如下：

- (1) 注重应用，按基本概念、基本原理、使用方法的思路阐述，条理清晰，通俗易懂；
- (2) 以理论讲述为基础，以应用案例为主线，通过对应用案例的不断完善，促进学生积极思考，激发其学习兴趣。对比较孤立的知识点，设计了单独完整的实例。书中还配备大量习题，用于巩固所学的知识；
- (3) 将 C++程序设计语言、Windows 程序设计和 MFC 程序设计紧密结合，由浅入深，形成一个完整体系。对未深入介绍的知识点，提供了相关参考资源。

本书内容分为 4 部分 14 章，具体内容如下：

第 1 部分（1~5 章）介绍 C++的关键语法和面向对象程序设计的思想。第 1 章介绍面向对象的基本概念和本书结构；第 2 章介绍 C++区别于 C 语言的语法知识，函数及其重载、函数模板等；第 3 章介绍 C++中类和对象的定义和使用，类模板；第 4 章介绍继承和多态的概念及其使用；第 5 章介绍输入/输出流及其使用。

第 2 部分（6~8 章）介绍 Windows 程序设计基本知识。第 6 章介绍 Windows 应用程序的结构和消息响应机制；第 7 章介绍键盘消息与鼠标消息的响应和处理方式；第 8 章介绍 Windows 应用程序中的资源及其使用。

第 3 部分（9~11 章）介绍 MFC 的核心机制以及 MFC 应用程序开发。第 9 章介绍 MFC 类层次结构；第 10 章介绍 MFC 中各种 Windows 标准控件的使用；第 11 章介绍 MFC 中两类主要的应用程序（文档/视图结构和基于对话框的结构）及其开发过程。

第 4 部分（12~14 章）结合实例，介绍 C++及 MFC 的高级应用。第 12 章介绍利用 MFC 的数据库类开发数据库应用程序；第 13 章介绍 Windows 多线程应用程序的开发；第 14 章介绍利用 WinSock 库开发 Windows 网络通信程序。

本书面向普通本科、专科和高职院校的学生，理论学时 24~32 学时，实验 24 学时。既可以作为单独的教材使用，也可以作为参考教材，供学生在“C++面向对象程序设计”课程实验和课程设计中使用，也适合工程技术人员参考。阅读本书需要有一定的 C 语言基础。

为了更好地配合任课教师的教学，帮助学生解决学习中的困惑，本书还提供了配套的电子教案。如果需要，可从北大出版社第六事业部网站下载或者通过邮件 liuhengyang@cqit.edu.cn 索取。

本书的编写得到了重庆工学院教材出版基金资助，同时得到了平顶山学院、南昌工程学院的大力支持。本书由黄贤英、刘恒洋、谢丽明、吴志强、谭智、李冰执笔编写。

由于编者水平有限，疏漏和不足之处在所难免，敬请读者批评指正。

编 者
2008 年 11 月

目 录

第 1 章 概述	1	2.3.1 重载函数声明	30
1.1 C++语言程序设计的特点	2	2.3.2 重载函数的解析步骤	31
1.1.1 抽象性	2	2.4 函数模板	32
1.1.2 封装性	3	2.4.1 函数模板的定义	32
1.1.3 继承性	3	2.4.2 函数模板的实例化	35
1.1.4 多态性	4	2.5 本章小结	36
1.2 类与对象	4	2.6 思考与练习	36
1.2.1 类	4	第 3 章 类	39
1.2.2 对象	5	3.1 类和对象	40
1.2.3 类和对象的关系	5	3.1.1 类定义	40
1.3 本书的目标及结构	5	3.1.2 类对象	44
1.3.1 C++、MFC 和 Windows 程序设计	5	3.1.3 类成员函数	45
1.3.2 示例系统概述	5	3.1.4 隐含的 this 指针	48
1.3.3 本书的结构安排	7	3.1.5 静态类成员	49
1.4 示例系统开发前的准备	7	3.2 类的初始化、赋值和析构	53
1.4.1 数据类型和数据结构的定义	7	3.2.1 类的显式初始化	53
1.4.2 示例系统高层协议设计	7	3.2.2 类的构造函数	54
1.4.3 创建服务端项目	7	3.2.3 类的析构函数	57
1.5 本章小结	9	3.2.4 类对象数组	59
1.6 思考与练习	10	3.3 重载操作符和用户定义的转换	60
第 2 章 C++基础	11	3.3.1 操作符重载	60
2.1 C++语言基础	12	3.3.2 友元	63
2.1.1 字符串类型	12	3.3.3 操作符=	64
2.1.2 const 限定修饰符	14	3.3.4 操作符[]	65
2.1.3 引用类型	16	3.3.5 操作符++和--	65
2.1.4 sizeof 操作符	18	3.3.6 用户定义的转换	67
2.1.5 new 和 delete 表达式	19	3.4 类模板和标准模板库	69
2.1.6 类型转换	20	3.4.1 类模板的定义	69
2.2 函数	23	3.4.2 类模板的实例化	73
2.2.1 函数参数类型检查	23	3.4.3 类模板的成员函数	76
2.2.2 参数传递	24	3.4.4 标准模板库——容器	78
2.2.3 返回一个值	28	3.5 示例系统中类的设计	81
2.2.4 inline 函数	29	3.5.1 消息实体类	82
2.3 重载函数	29	3.5.2 在线用户信息类	84
		3.6 本章小结	84

3.7	思考与练习	85	5.7.2	ifstream 类	125
第 4 章	类的继承和多态性	88	5.7.3	使用 open()函数打开文件	125
4.1	继承和多态的概念	89	5.7.4	fstream 类	126
4.2	基类和派生类	90	5.7.5	文件的随机读/写	126
4.2.1	定义基类	91	5.8	格式控制	127
4.2.2	定义派生类	92	5.8.1	布尔量的输出控制操作符	127
4.3	基类成员访问	93	5.8.2	带进制整数的输出控制 操作符	128
4.4	基类和派生类的构造	96	5.8.3	浮点数输出控制操作符	128
4.4.1	基类构造函数	97	5.8.4	输入控制操作符	129
4.4.2	派生类构造函数	98	5.8.5	刷新缓冲区操作符	130
4.4.3	析构函数	98	5.8.6	填充字符设置操作符	130
4.5	基类和派生类中的虚拟函数	99	5.9	示例系统输入/输出类的设计	131
4.5.1	纯虚拟函数在基类中 的应用	100	5.9.1	输入/输出需求	131
4.5.2	虚拟函数的静态调用	100	5.9.2	输入/输出类的实现	132
4.5.3	虚拟析构函数	101	5.10	本章小结	133
4.6	多重继承	103	5.11	思考与练习	133
4.6.1	多重继承的定义	103	第 6 章	Windows 应用程序	135
4.6.2	多重继承的构造函数 和析构函数	103	6.1	Windows 编程基础	136
4.6.3	虚继承	106	6.1.1	窗口	136
4.7	本章小结	110	6.1.2	事件驱动	137
4.8	思考与练习	110	6.1.3	句柄和 Windows 消息	137
第 5 章	C++的输入/输出流	113	6.2	Windows 应用程序常用消息	139
5.1	概述	114	6.3	Windows 中的事件驱动程序设计	140
5.2	输出操作符<<	115	6.4	Windows 应用程序的基本结构	141
5.2.1	常见的输入/输出	115	6.4.1	Windows 应用程序的组成	141
5.2.2	变量地址输出	115	6.4.2	源程序的组成结构	142
5.2.3	输出操作符的优先级	116	6.5	应用程序举例	147
5.3	输入操作符>>	116	6.6	本章小结	149
5.4	其他输入/输出方式	119	6.7	思考与练习	149
5.4.1	get()和 put()函数	119	第 7 章	Windows 中的键盘与 鼠标消息	150
5.4.2	read()和 write()函数	120	7.1	键盘消息	151
5.4.3	其他函数	121	7.2	键盘消息应用举例	153
5.5	重载输出操作符<<	121	7.3	鼠标消息	156
5.6	重载输入操作符>>	122	7.4	鼠标消息应用程序实例	159
5.7	文件输入和输出	123	7.5	本章小结	160
5.7.1	ofstream 类	124			

7.6 思考与练习	160	第 10 章 MFC 中 Windows	
第 8 章 Windows 编程中		标准控件的应用	197
资源的应用	161	10.1 概述	198
8.1 菜单和加速键资源及其应用	162	10.2 按钮控件及其应用	199
8.1.1 菜单的创建过程	162	10.2.1 按钮控件的使用	200
8.1.2 菜单项的操作	165	10.2.2 按钮控件示例	200
8.1.3 菜单的动态创建	167	10.3 滚动条控件	204
8.1.4 加速键资源的创建过程	168	10.3.1 滚动条类的结构及其方法 ..	204
8.1.5 菜单加速键资源应用示例 ..	169	10.3.2 滚动条类编程实例	205
8.2 位图资源及其应用	171	10.4 列表框控件	208
8.2.1 位图的概念	171	10.4.1 列表框控件的类结构	208
8.2.2 位图的操作过程	172	10.4.2 列表框控件类的成员函数 ..	208
8.2.3 位图操作实例	174	10.4.3 列表框和应用程序	
8.3 对话框资源及其应用	175	之间的消息传递	210
8.3.1 模态对话框的编程方法	175	10.4.4 列表框应用举例	210
8.3.2 模态对话框应用实例	178	10.5 编辑框控件	211
8.3.3 非模态对话框的编程方法 ..	180	10.5.1 编辑框控件简介	211
8.4 图标资源的应用	181	10.5.2 编辑框与应用程序间	
8.4.1 图标资源的操作	181	的消息传递	212
8.4.2 图标资源应用举例	182	10.5.3 编辑框类编程实例	213
8.5 本章小结	182	10.6 组合框控件	214
8.6 思考与练习	183	10.6.1 组合框类的结构	
第 9 章 MFC 基础	184	及组合框的特点	215
9.1 MFC 概述	185	10.6.2 组合框与应用程序	
9.2 MFC 类的组织结构及		之间的消息传递	215
主要类的简介	185	10.6.3 组合框控件应用举例	215
9.2.1 MFC 类的组织结构	185	10.7 对话框通用控件	216
9.2.2 根类	186	10.7.1 Picture 控件的使用	216
9.2.3 应用程序体系结构类	187	10.7.2 Spin 控件的使用	217
9.2.4 可视对象类	190	10.7.3 Progress 控件的使用	218
9.2.5 通用类	192	10.7.4 Slider 控件的使用	220
9.2.6 OLE 类	193	10.7.5 Date Time Picker	
9.2.7 ODBC 数据库类	193	控件的使用	221
9.3 MFC 中的全局函数与全局变量	193	10.7.6 List Control 控件的使用	222
9.4 应用程序向导	194	10.7.7 Tree Control 控件的使用	225
9.5 本章小结	196	10.8 本章小结	228
9.6 思考与练习	196	10.9 思考与练习	228

第 11 章 简单 MFC 应用程序开发	230
11.1 文档/视图概述	231
11.1.1 单文档界面与多文档界面 ..	231
11.1.2 文档/视图结构	231
11.1.3 SDI 程序中文档、 视图对象的创建过程	232
11.1.4 文档/视图结构程序 的消息传递过程	232
11.2 文档/视图结构程序的 主要成员	233
11.2.1 CWinApp 类	233
11.2.2 CDocument 类	233
11.2.3 CView 类	235
11.2.4 CDocTemplate 类	236
11.2.5 CFrameWnd 类	237
11.2.6 串行化处理	238
11.3 SDI 编程实例	239
11.4 对话框应用程序设计	241
11.4.1 对话框和控件的基本概念 ..	241
11.4.2 对话框模板的设计	242
11.4.3 对话框类的设计	242
11.4.4 完善示例系统的客户端	246
11.5 本章小结	263
11.6 思考与练习	264
第 12 章 MFC 数据库应用程序开发	265
12.1 数据库系统的基本概念	266
12.2 ODBC 基本概念	267
12.3 MFC 的 ODBC 类简介	268
12.4 CDatabase 类	268
12.5 CRecordset 类	269
12.5.1 动态集、快照、光标 和光标库	269
12.5.2 SQL 查询	270
12.5.3 记录集的建立和关闭	271
12.5.4 滚动记录	272
12.5.5 修改、添加和删除记录	273
12.6 示例系统中数据库 操作类的设计	275
12.6.1 数据库设计	275
12.6.2 注册数据源	275
12.6.3 数据库操作类实现	277
12.7 本章小结	279
12.8 思考与练习	279
第 13 章 Windows 多线程应用 程序开发	280
13.1 多任务、进程和线程	281
13.1.1 Windows 95/NT 的抢先式多任务	281
13.1.2 进程与线程	281
13.1.3 线程的创建和终止	281
13.2 线程同步	283
13.2.1 为什么需要同步	283
13.2.2 事件对象	284
13.2.3 临界段	286
13.2.4 互斥体	287
13.2.5 信号计数器	288
13.3 线程通信	289
13.3.1 利用全局变量进行通信	290
13.3.2 利用自定义的消息 进行通信	290
13.4 示例系统多线程设计	292
13.4.1 服务端多线程的设计	292
13.4.2 客户端多线程的设计	295
13.5 本章小结	296
13.6 思考与练习	296
第 14 章 WinSock 编程基础	297
14.1 网络原理和协议	298
14.1.1 协议的特征	298
14.1.2 Win32 支持的协议	301
14.1.3 WinSock2 协议信息	302
14.1.4 Windows 套接字	303
14.1.5 选择合适的协议	304
14.2 网际协议	305
14.2.1 TCP 和 UDP	305
14.2.2 定址	305
14.2.3 创建套接字	308
14.3 WinSock 基础	308
14.3.1 WinSock 的初始化	308

14.3.2	错误检查和控制	308	14.4.4	完善消息处理线程	316
14.3.3	面向连接的协议	308	14.4.5	完善服务端主函数	318
14.3.4	无连接协议	313	14.4.6	完善客户端消息处理线程 ..	319
14.4	示例系统中通信协议的实现	314	14.5	本章小结	331
14.4.1	完善基础类	315	14.6	思考与练习	331
14.4.2	收发消息类的实现	315	附录	332
14.4.3	完善服务端收发消息线程 ..	316	参考文献	347

第 1 章

概 述

教学目标

- 理解 C++ 的特点
- 掌握类与对象的概念
- 了解本书的组织结构和示例系统
- 完成开发示例系统的前期准备工作

知识结构

知识要点	相关知识	能力要求
C++ 的特点	抽象性、封装性、继承性、多态性	结合现实世界举例说明 C++ 的特点
类与对象	类、对象、类和对象的关系	结合现实世界举例说明类与对象的概念
示例系统	C++、MFC、Windows 程序设计、开发环境、总体框架、其他约定	区别 C++、MFC、Windows 程序设计，描述示例系统总体框架

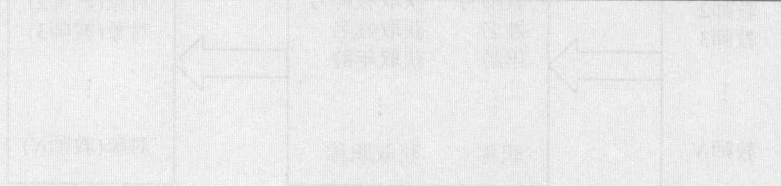


图 1-1-1 C++、MFC 和 Windows 程序设计的关系

C++是从 C 语言发展而来的, C++在 C 语言的基础上增加了面向对象的机制, 这主要体现在数据封装、重载、继承和多态等几个方面。本章主要介绍 C++的特点、类与对象的基本概念, 以及本书的组织结构, 最后介绍示例系统的基本功能、基本数据类型的定义以及高层协议的设计, 为后续章节示例系统的开发做准备。

1.1 C++语言程序设计的特点

C++语言是美国 Bell 实验室于 20 世纪 80 年代在 C 语言的基础上成功开发出来的, 它保留了 C 语言的全部优点, 增加了面向对象机制, 支持面向对象程序设计方法。C++既可以用于结构化程序设计, 又可以用于面向对象程序设计, 因此是一种混合型面向对象程序设计语言, 得到了广泛应用。

面向对象程序设计方法的基本特性包括抽象性、封装性、继承性和多态性。C++作为一种面向对象的程序设计语言, 对以上特性具有非常好的支持。

1.1.1 抽象性

将自然界中的各种事物进行归纳分类是人类认识客观世界时经常采用的思维方法。分类所依据的重要原则是抽象, 即分析和提取事物中与当前目标有关的本质特征, 忽略非本质特征, 找出事物的共同特性, 形成一个抽象的概念——将具有共性的事物分为一“类”。

下面以设计一个“教师类”为例来说明 C++中对事物的抽象过程。假设教师类用于高校人事管理系统, 那么该人事管理系统管理的主要对象就是教师, 重点对“教师”进行分析。根据人事管理系统的功能可以归纳出“教师”的主要属性和操作。不能把目光只停留在单个教师的个体信息身上, 而是要将整个教师作为一个整体来对待, 并抽象形成一个类别, 即面向对象中的类(教师类)。作为这个类的实例, 可以建立许多教师个体实体, 各个教师实体就是该类的对象, 将教师群体抽象为类与对象的过程如图 1.1 所示。

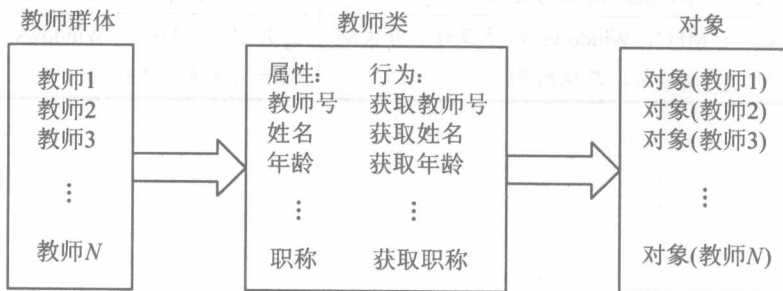


图 1.1 抽象过程示意图

因此, 抽象性是对事物本质特征的概括性描述。将客观事物抽象为类和对象是面向对象程序设计的第一步, 也是非常重要和关键的一步。

C++的抽象性是指从特定的实例中抽取共同性质形成一般化概念的过程, 它包括行为抽象和数据抽象两个方面。行为抽象是指任何一个具有明确功能的操作, 即使这个操作由一系列更简单的操作来支持, 使用者都可以将其视为单个操作实体。数据抽象是指定义数

据对象和对该对象的操作，并规定这些对象的值只能通过规定的操作进行调用和修改。在C++程序设计中，对具体问题的分析和抽象是通过类的定义和应用来实现的。

1.1.2 封装性

封装是面向对象方法的一个重要原则，即将对象的属性和行为代码封装在对象的内部，形成一个独立的单位，并尽可能隐蔽对象的内部细节。C++面向对象的封装特性包含两层含义：一是将对象的全部属性和行为封装在对象的内部，形成一个不可分割的独立单位，对象的属性值只能由这个对象的行为来读取和修改；二是“信息隐蔽”，即尽可能隐蔽对象的内部细节，对外形成一道屏障，只保留有限的对外接口与外部发生联系。

面向对象程序设计方法的信息隐蔽作用体现了自然界中事物的相对独立性，程序设计者与使用者只需关心其对外提供的接口，而不必过分注意其内部细节。封装性使对象以外的事物不能随意获取对象的内部属性，从而有效地避免了外部的错误访问对它产生的影响，大大减轻了软件开发过程中查错的工作量，有效地减小了排错的难度。

C++一般用类来实现封装性，并通过设置对数据的访问权限来控制对内部数据的访问，即将类的部分成员作为类的外部接口，将其他成员隐藏起来，实现对数据成员的合理控制，使程序不同部分之间的相互影响尽可能降到最低。

1.1.3 继承性

客观事物既有共性，也有个性，如果只考虑事物的共性，而不考虑事物的个性，就不能真实反映出客观世界中事物之间的层次关系。面向对象中的抽象性意味着舍弃对象的次要特性，提取事物的本质共性，形成类。如果在类的基础上，根据实际目标的合理考虑一部分对象的个性特征，则可形成一个新的类，新类具有前一个类的全部特征，是前一个类的派生类。这种由前一个类派生出新类的现象符合自然界事物之间的层次结构关系，又称为类的继承结构或类的层次结构，图1.2所示为汽车类继承关系的层次结构。

继承是连接类与类的一种层次模型，继承是面向对象程序设计能够提高软件开发效率的重要原因之一。继承意味着派生类中无需重新定义在上一级类中已经定义的属性和行为，而是自动地、隐含地拥有其上一级类的全部属性和行为。继承是可以传递的，这体现了大自然中特殊与一般的关系。

C++允许从一个或多个已经定义的类中派生出新的类并继承其数据和操作，被继承的类称为基类或父类，派生出的新类称为派生类或子类。在新的派生类中既可以重新定义，也可以增加新的数据与操作，这很好地体现了程序代码可重用的思想。

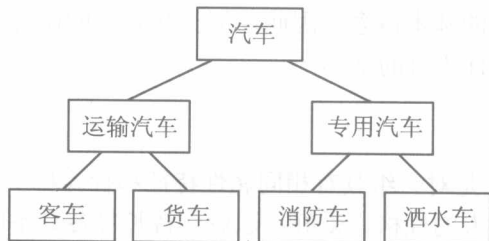


图 1.2 类继承的层次结构关系

1.1.4 多态性

客观世界具有多态性，如不同的对象个体在获得相同的信息(即消息)时可以产生各种不同的行为和结果。面向对象程序设计借鉴了客观世界的这种多态性。

面向对象程序设计的多态性是指父类中定义的属性或行为，派生类继承之后，可以具有不同的数据类型或表现出不同的行为特性。如类中的同名函数可以对应多个具有相似功能的不同函数，可以使用相同的调用方式来调用这些具有不同功能的同名函数。

多态性使得同一个属性或行为在父类及其各派生类中具有不同的语义。以图 1.3 为例，程序设计者首先可以通过定义一个一般的“几何图形”类，该类中具有“计算面积”等行为，但这个行为并不具有具体含义，也就是说并不确定执行时究竟是计算什么几何图形的面积。然后再定义一些派生类，如“三角形”、“矩形”、“圆”等，它们都继承父类“几何图形”的“计算面积”这个行为；接下来，程序设计者可以在这些子类中根据具体需要重新定义“计算面积”这个操作，使之分别实现对“三角形”、“矩形”、“圆”等图形的面积计算功能。这些派生类均继承于“几何图形”，但其功能却各不相同，这样，同一个消息发出后，“三角形”、“矩形”、“圆”等类的对象接收到这个消息后，将执行计算不同图形的面积的功能，这就是多态性的表现。

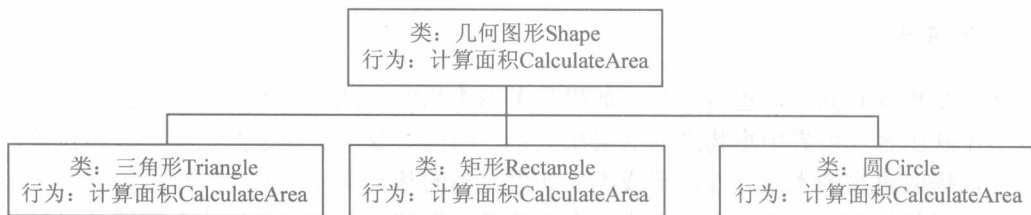


图 1.3 多态性示意图

C++中的多态性分为编译时多态和运行时多态。编译时多态是指在程序编译阶段由编译系统根据操作数的个数不同或操作数的类型不同，确定需要调用哪个同名的函数；运行时多态是指在程序运行阶段才根据程序运行中产生的消息确定需要调用哪个同名的函数。这些函数虽然名称相同但却具有不同的功能，因此将产生不同的操作。在 C++中，编译时多态是通过函数重载和运算符重载实现的，运行时多态是通过继承和虚函数来实现的。

1.2 类与对象

类和对象是面向对象的基本概念，因此理解类和对象的概念是学好本书的关键，也是学好其他面向对象程序设计语言的基础。

1.2.1 类

在客观世界中，“类”是对一组具有相同属性特征和行为特征的同类事物的抽象。如张三、李四等，虽然每个人的性格、爱好、职业、特长等各有不同，但是他们的基本特征是相似的，都具有相同的生理构造，都能吃饭、说话、走路等，于是把他们统称为一个类“人”。具体的每一个人“人”这个大类中的一个实例，即一个对象。

1.2.2 对象

在现实世界中,对象是认识世界的基本单元。如某一个人、某一辆车、某一个动物等,这些都是具体的对象。对象具有静态特征和动态特征,在程序设计语言中,把静态特征描述为属性,把动态特征描述为行为。一个对象由一组属性和一组行为构成,对象只有在具有属性和行为的情况下才具有意义。换句话说,对象是包含数据和处理这些数据的操作的程序单元,是构成面向对象程序的基本计算单位,由数据及其操作构成。

1.2.3 类和对象的关系

类和对象之间的关系是抽象和具体的关系。类是对多个对象进行抽象的结果,对象是类的个体,一个对象是类的一个实例。从类到对象的过程称为实例化。

指定一个类时,并没有指向某一个具体的事物,而是指明了某一类事物的集合。这在程序设计语言中表现为,定义一个类时,并不占用内存空间,只有利用这个类定义对象时,这个对象才占用内存空间,也就是说,这个时候已经指向了这个类中某一个具体的个体了。

1.3 本书的目标及结构

本书的目标是既让读者掌握一定的理论基础知识,又能开发一个基于 TCP/IP 的点对点网络通信系统(以后称为示例系统)。在讲述 C++基础理论、Windows 程序设计以及 MFC 程序设计的同时逐步完成示例系统的设计与开发。本节将介绍示例系统与本书中各知识点的关系以及本书的结构安排。

1.3.1 C++、MFC 和 Windows 程序设计

简单地说,C++是一种标准化的编程语言。MFC(Microsoft Foundation Class)是微软公司为了方便在 Windows 平台上进行应用程序开发,采用 C++语言对 Windows API(Application Programming Interface)进行封装,开发出来的一套应用程序开发类库。Windows 程序设计是指基于 Windows 平台的应用程序设计,它有其特有的消息机制和事件驱动机制。

1.3.2 示例系统概述

1. 开发环境

本书中的所有程序(包括示例系统)均在 Microsoft Visual Studio 2005 中测试通过。

2. 总体构架

示例系统的主要功能是实现基于 TCP/IP 的点对点网络通信,它分为两大部分:服务端示例系统(服务器)和客户端示例系统(客户端),服务器项目名称为 PeerToPeerServer,客户端项目名称为 PeerToPeerClient。该系统的主要功能示意图如图 1.4 所示。