

Visual C++

管理信息系统 完整项目 实例剖析

■ 明日科技 编著

本书特色:

- ◆ 详尽剖析一个完整的项目实例的开发过程
- ◆ 重点讲解系统各部分的设计思路、实现目标和设计流程
- ◆ 综合应用数据库系统开发的思路、技术和方法
- ◆ 随书附带的光盘中包含书中案例的源代码



附光盘



人民邮电出版社
POSTS & TELECOM PRESS

Visual C++

管理信息系统

完整项目

实例剖析

■ 明日科技 编著

人民邮电出版社



图书在版编目 (CIP) 数据

Visual C++管理信息系统完整项目实例剖析/明日科技编著. —北京: 人民邮电出版社, 2005.7

ISBN 7-115-13702-1

I. V... II. 明... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 089353 号

内 容 提 要

本书围绕一个完整的医药企业信息系统, 按照商业信息管理系统设计的实现过程, 详细介绍使用 Visual C++开发商业信息系统的思路、方法和技术。本书不仅给出系统设计步骤和源代码, 而且突出介绍了各个部分的设计思路、实现目标和设计流程, 力求使读者抓住本质, 迅速掌握使用 Visual C++开发管理系统的相关知识, 学到设计管理信息系统的设计思路和方法。

本书内容详实, 案例完整, 步骤详尽, 具有很强的实用性。适合于各级软件开发人员学习使用, 也可供大中专院校师生学习参考。

在本书配套光盘中提供了书中案例的全部源代码, 源代码都经过精心调试, 在 Windows 2000 下全部通过, 确保能够正常运行。

Visual C++管理信息系统完整项目实例剖析

- ◆ 编 著 明日科技
责任编辑 马 嘉
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 28.25
字数: 691 千字 2005 年 7 月第 1 版
印数: 1—6 000 册 2005 年 7 月北京第 1 次印刷

ISBN 7-115-13702-1/TP · 4822

定价: 48.00 元 (附光盘)

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223

前 言

随着全球信息技术的飞速发展,尤其是因特网的出现和普及,企业管理和决策都发生了根本的改变。采用信息技术促进、改造企业的生产、经营、管理,有效采集和处理信息,增强企业决策的有效性,适应全球经济发展的要求,将成为企业发展的必由之路。信息管理系统已成为现代企业的一个重要标志和衡量企业综合实力的重要组成部分。但是,信息系统的开发却非常困难。开发一个成功的企业信息管理系统,不仅要有方方面面的专业知识,还要充分分析企业内部、外部环境的变化和关系,以及在信息系统开发过程中所涉及的设备、技术上的复杂性。这就需要系统开发者不断总结经验,不断探索更好的方法和技术。如果把开发一个商业信息系统的设计思路、开发过程按实际开发过程完整介绍出来,对提高程序员的实际编程经验无疑是很有帮助的。但现在图书市场上介绍这方面的书籍很少,客观上制约了程序开发人员能力的提高。

本书通过对一个完整、实用的企业信息系统的分析和设计,详细介绍了使用 Visual C++ 开发商业信息系统的思路、技术和方法。本书虽是一本案例书,但不是仅给出系统设计步骤和源代码,而是突出介绍了各个部分的设计思路、实现目标和设计流程,让读者明白设计程序的思路与过程。力求使读者抓住本质,迅速掌握使用 VC++ 开发管理系统的相关知识,学到设计管理信息系统的设计思路和方法。

全书虽然并没有 100% 追求面向对象,但大多数模块都是以面向对象的设计思路开发完成的,初学者也许会陷入复杂的类层次关系中,当认真读完此书,读者会发现,它所带来的益处是许多介绍“面向对象程序设计”的书籍无法比拟的。

技术方面,作者遵循重点问题重点解决的方针,将多年开发经验和精华融入其中,尤其是打印及打印预览一章,更是对文档/视图结构进行了精华提炼。

全书共分 11 章,各章内容如下。

第 1 章“Visual C++ 6.0 编程基础”:主要介绍 Visual C++ 6.0 编程基础,包括:C++ 语法基础、Windows 的编程模式和 VC++ 可视化编程环境概述。

第 2 章“医药管理系统设计”:主要介绍医药管理系统需求分析,系统分析、系统设计、数据库设计等过程。

第 3 章“系统登录、主窗体程序设计”:主要介绍主窗体、系统登录窗体的设计思路、设计步骤及开发关键。另外,还通过专题讲座介绍了如何在 Visual C++ 6.0 使用 ADO 访问数据库。

第 4 章“基础信息管理”:主要介绍医药管理系统中基础信息模块的设计思路和方法。同时重点介绍了自定义基类、自定义派生类、动态创建控件等技术。

第 5 章“销售模块”:主要介绍医药管理系统中销售登记模块、销售退货模块、销售查询模块及销售退货查询模块的设计思路、方法和步骤。同时还介绍了列表控件类 ClistCtrl 的使用技术。

第 6 章“入库模块”:主要介绍医药管理系统中入库登记模块、入库退货模块、入库查询模块及入库退货查询模块的设计思路、方法和步骤。

第 7 章“调货模块”:主要介绍医药管理系统中调货管理模块、调货查询模块、入库查

询模块的设计思路、方法和步骤。

第 8 章“库存模块”：主要介绍医药管理系统中库存盘点模块、库存查询模块、仓库管理模块的设计思路、方法和步骤。

第 9 章“账务模块”：主要介绍医药管理系统中客户、供应商往来账目管理模块、结账管理模块、日结算、月结算管理模块的设计思路、方法和步骤。

第 10 章“打印及打印预览”：主要介绍了利用文档/视图结构实现打印的理论及方法，不仅限于此，还提供了类似于 Word 的多页预览功能。

第 11 章“系统设置模块”：主要介绍医药管理系统中密码设置模块、权限设置模块、用户设置模块及数据备份与恢复模块的设计思路、方法和步骤。

本书配套光盘中提供了书中案例的全部源代码，所有源代码都经过精心调试，在 Windows 2000 下全部通过，确保能够正常运行。

本书内容详实，案例完整，步骤详尽，具有很强的实用性。适合于各级软件开发人员学习使用，也可供大中专院校师生学习参考，是编程者不可多得的一本好书。

本书由明日科技组织编著，参加编写人员有梁小勇、王春才、张雨、李赫、邹天思、顾彦冷、刘欣、高春艳、王国辉、李浩然、张世辉、王晶莹、高茹、郝洪斌、郭锐、高月、郭铁、王晶洁、高飞、高润岭、邹淑芳等。作者的水平有限，书中难免存在疏漏之处，我们诚挚地希望读者对本书的不足之处予以批评指正。

技术支持网站：<http://www.mingrisoft.com>

技术服务信箱：mingrisoft@mingrisoft.com

明日科技

光盘使用说明

本书以 Visual C++ 实际开发“现代流通企业综合管理信息系统”为背景，按照商业信息管理系统设计的实现过程，循序渐进的介绍商业信息管理系统开发的思路、方法和技巧。为使读者更好地学习和使用本书，本书光盘附带了书中范例的源程序和可执行文件。为保证正常使用，用户在使用本书光盘时，应按如下说明设置。

一、运行环境及系统配置

(1) 本实例是在 Windows 2000 下开发的，程序测试环境为 Windows 2000 和 Windows XP。用户在 Windows 98、Windows 2000 和 Windows XP 下都可使用。

(2) 系统时间设定，为保证程序查询功能的正常使用，请将系统时间格式改为长日期格式。步骤如下（以 Windows 98 为例）：

- ① 打开控制面板，选择“区域与语言选项”，选择“区域选项”对话框。
- ② 在“区域选项”选项卡选择“自定义”按钮，将弹出“自定义区域”对话框。
- ③ 在“自定义区域”对话框中选择“日期”选项卡，设置日期格式为长日期格式。

二、数据库环境配置

(1) 本书实例的后台采用的是 Access 数据库，数据库文件“ypgl.mdb”保存在光盘目录下 Mingrisoft 中。

(2) 程序正常运行需要配置 ODBC，作者开发了“AutoConfig.exe”用于自动完成此工作。“AutoConfig.exe”的源程序在光盘目录下“自动配置 ODBC”目录中（通常情况下，读者无需考虑数据库的配置问题，管理系统会自动完成此项工作）。

(3) 如果由于某些特殊原因管理系统无法自动配置 ODBC，则需要读者手工完成，配置的方法如下（以 Windows 2000 为例）：

- ① 在 Windows 2000 下打开“控制面板”，双击“管理工具”，在“管理工具”窗口中双击“数据源（ODBC）”，打开“ODBC 数据源管理器”。
- ② 在弹出的“ODBC 数据源管理器”中选择“用户 DNS”选项页，如图 1 所示。
- ③ 单击【添加】按钮，打开“创建新数据源”对话框。在此对话框中双击“Microsoft Access Driver(*.mdb)”列表项，打开“ODBC Microsoft Access 安装”对话框。
- ④ 在“ODBC Microsoft Access 安装”对话框中“数据源名”文本框内输入“yaopin”。单击【选择】按钮，找到光盘目录下的“ypgl.mdb”文件。
- ⑤ 单击【确定】按钮，回到“ODBC Microsoft Access 安装”对话框，如图 2 所示。
- ⑥ 单击【确定】按钮，完成 ODBC 的创建工作。

三、源程序使用方法

(1) 如用户使用源程序，应在系统上安装 Visual C++ 6.0 企业版。

(2) “医药进销存管理系统”的源程序在附赠光盘中“Mingrisoft”文件夹下。要使用它，请将该文件夹拷贝到计算机硬盘上，并去掉所有文件的只读属性。

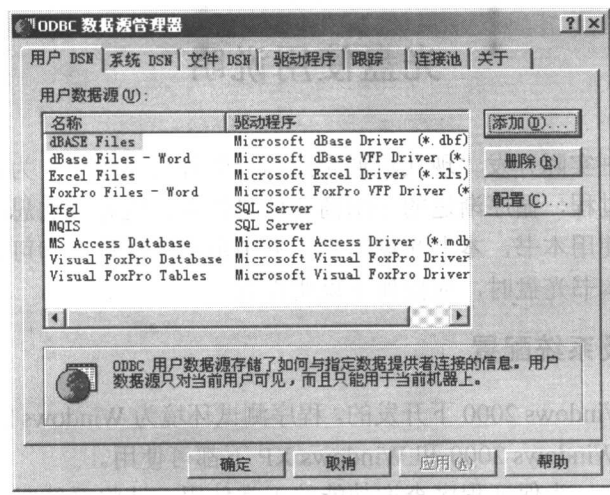


图 1 ODBC 数据源管理器

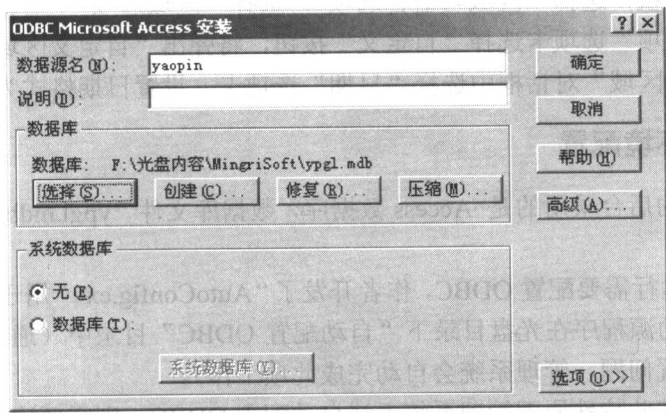


图 2 ODBC Microsoft Access 安装

(3) 运行“医药进销存管理系统.exe”，待系统进入登录界面后，可通过下面的用户名和密码登录系统。

用户名: mrkj

密码: 111

四、多媒体演示内容

- (1) 系统环境配置及程序使用演示。
- (2) 程序中关键模块的开发过程演示。

目 录

第 1 章 Visual C++ 6.0 编程基础	1	2.4.5 输出设计	54
1.1 C++语法基础	1	2.4.6 网络设计	55
1.1.1 程序设计语言介绍	1	2.4.7 系统安全和可靠性设计	56
1.1.2 C++语法基础知识	3	2.5 数据库设计	56
1.1.3 C++高级应用 ——虚函数	11	2.5.1 数据库概念结构设计	56
1.2 Windows 的编程模式	12	2.5.2 数据库物理结构设计	60
1.2.1 主程序	12	第 3 章 主窗体、系统登录程序设计	72
1.2.2 消息处理	13	3.1 概述	72
1.2.3 创建窗口	13	3.1.1 功能介绍	72
1.3 VC++可视化编程环境概述	15	3.1.2 业务流程	72
1.3.1 介绍 Visual C++的组成	15	3.2 关键技术	73
1.3.2 菜单介绍	15	3.3 主窗体程序设计	76
1.3.3 调试技术	23	3.3.1 实现目标	76
第 2 章 医药管理系统设计	33	3.3.2 开发流程	76
2.1 医药管理系统概述	33	3.3.3 设计思路	77
2.1.1 系统介绍	33	3.3.4 创建主窗体	77
2.1.2 主要功能	33	3.3.5 创建程序菜单	77
2.1.3 主要模块介绍	34	3.3.6 为对话框创建工具栏	79
2.2 需求分析	37	3.3.7 为菜单和工具栏添 加消息处理函数	83
2.2.1 开发背景	37	3.3.8 为对话框创建状态栏	83
2.2.2 业务调查	38	3.3.9 设置窗体背景	86
2.2.3 手工操作业务流程	39	3.3.10 根据权限设置菜单和 工具栏是否可用	86
2.2.4 用户要求	40	3.3.11 开发小结	93
2.2.5 数据字典	41	3.4 数据库连接设计	93
2.3 系统分析	43	3.4.1 实现目标	93
2.3.1 问题分析与解决	43	3.4.2 设计思路	94
2.3.2 系统功能分析	44	3.4.3 编程代码	94
2.3.3 子系统分析	49	3.5 系统登录程序设计	96
2.3.4 系统结构	49	3.5.1 实现目标	97
2.4 系统设计	51	3.5.2 开发流程	97
2.4.1 设计目标	51	3.5.3 设计思路	97
2.4.2 开发及运行环境	52	3.5.4 程序开发步骤	98
2.4.3 编码设计	52	3.5.5 开发小结	101
2.4.4 输入设计	53		

3.6 专题技术	101	5.4 销售退货模块设计	193
3.7 本章总结	107	5.4.1 实现目标	193
第4章 基础信息管理	108	5.4.2 开发流程	194
4.1 基础信息管理概述	108	5.4.3 设计思路	194
4.1.1 功能分析	108	5.4.4 程序开发步骤	195
4.1.2 业务流程	108	5.5 销售查询模块设计	218
4.2 关键技术	109	5.5.1 实现目标	218
4.2.1 在查询结果中查询	109	5.5.2 设计思路	219
4.2.2 自动生成编号	109	5.5.3 程序开发步骤	219
4.3 抽象出查询的基类	111	5.6 销售退货查询模块设计	220
4.3.1 实现目标	111	5.6.1 实现目标	220
4.3.2 开发流程	111	5.6.2 设计思路	220
4.3.3 设计思路	112	5.6.3 程序开发步骤	221
4.3.4 程序开发步骤	112	5.7 专题技术	221
4.3.5 开发小结	123	第6章 入库管理	233
4.4 基础信息管理类设计	123	6.1 入库管理概述	233
4.4.1 实现目标	123	6.1.1 功能分析	233
4.4.2 开发流程	123	6.1.2 业务流程	233
4.4.3 设计思路	125	6.2 入库登记模块设计	233
4.4.4 程序开发步骤	125	6.2.1 实现目标	233
4.4.5 数据访问类与界面类 如何协调工作	134	6.2.2 开发流程	234
4.5 本章总结	139	6.2.3 设计思路	235
第5章 销售管理	140	6.2.4 程序开发步骤	235
5.1 销售管理概述	140	6.3 入库退货登记设计	245
5.1.1 功能分析	140	6.3.1 实现目标	245
5.1.2 业务流程	140	6.3.2 开发流程	245
5.2 关键技术	140	6.3.3 设计思路	245
5.2.1 使 CListCtrl 控件可 编辑	140	6.3.4 程序开发步骤	247
5.2.2 显示自动提示窗口 (CListCtrlPop)	149	6.4 入库查询模块设计	255
5.2.3 处理局部白色背景	156	6.4.1 实现目标	255
5.2.4 给编辑框加一个下划线	157	6.4.2 设计思路	255
5.2.5 修改控件字体	158	6.4.3 程序开发步骤	256
5.3 销售登记模块设计	158	6.5 入库退货查询模块设计	256
5.3.1 实现目标	158	6.5.1 实现目标	256
5.3.2 开发流程	158	6.5.2 设计思路	256
5.3.3 设计思路	159	6.5.3 程序开发步骤	257
5.3.4 程序开发步骤	160	第7章 调货管理	258
		7.1 调货管理概述	258
		7.1.1 功能分析	258
		7.1.2 业务流程	258

7.2 调货登记模块设计·····	258	9.4.3 程序开发步骤·····	329
7.2.1 实现目标·····	258	第 10 章 打印及打印预览 ·····	338
7.2.2 开发流程·····	259	10.1 打印及打印预览概述·····	338
7.2.3 设计思路·····	259	10.2 关键技术·····	339
7.2.4 程序开发步骤·····	260	10.2.1 文档/视图的内部结构·····	339
7.3 调货查询模块设计·····	279	10.2.2 加入 Document/View 框架·····	352
7.3.1 实现目标·····	279	10.3 模拟 Word 2000 的打印预览·····	359
7.3.2 设计思路·····	280	10.3.1 实现目标·····	359
7.3.3 程序开发步骤·····	280	10.3.2 开发流程·····	359
第 8 章 库存管理 ·····	281	10.3.3 设计思路·····	360
8.1 库存管理概述·····	281	10.3.4 程序开发步骤·····	362
8.1.1 功能分析·····	281	10.4 打印绘图·····	394
8.1.2 业务流程·····	281	10.4.1 系统分析·····	394
8.2 库存盘点模块设计·····	281	10.4.2 开发流程·····	395
8.2.1 实现目标·····	281	10.4.3 设计思路·····	395
8.2.2 开发流程·····	282	10.4.4 程序开发步骤·····	398
8.2.3 设计思路·····	282	10.5 专题技术·····	411
8.2.4 程序开发步骤·····	283	10.5.1 打印及打印预览·····	411
8.3 库存查询模块设计·····	301	10.5.2 Windows 图形设备接口 GDI (Graphics Device Interface)·····	413
8.3.1 实现目标·····	301	第 11 章 系统设置管理 ·····	421
8.3.2 设计思路·····	302	11.1 系统设置管理概述·····	421
8.3.3 程序开发步骤·····	302	11.1.1 功能分析·····	421
8.4 仓库管理模块设计·····	303	11.1.2 业务流程·····	421
8.4.1 实现目标·····	303	11.2 密码设置模块设计·····	422
8.4.2 设计思路·····	303	11.2.1 实现目标·····	422
8.4.3 程序开发步骤·····	303	11.2.2 设计思路·····	422
第 9 章 账务管理 ·····	310	11.2.3 程序开发步骤·····	422
9.1 账务管理概述·····	310	11.3 权限设置模块设计·····	424
9.2 客户、供应商往来账目管理 模块设计·····	310	11.3.1 实现目标·····	424
9.2.1 实现目标·····	310	11.3.2 设计思路·····	425
9.2.2 设计思路·····	311	11.3.3 程序开发步骤·····	425
9.2.3 程序开发步骤·····	312	11.4 用户设置模块设计·····	429
9.3 结账管理模块设计·····	319	11.4.1 实现目标·····	429
9.3.1 实现目标·····	320	11.4.2 设计思路·····	429
9.3.2 设计思路·····	322	11.4.3 程序开发步骤·····	430
9.3.3 程序开发步骤·····	322	11.5 数据备份模块设计·····	432
9.4 日结算、月结算管理模块设计·····	328	11.5.1 实现目标·····	432
9.4.1 实现目标·····	328		
9.4.2 设计思路·····	329		

11.5.2 设计思路.....	432	11.6.1 实现目标	435
11.5.3 程序开发步骤.....	433	11.6.2 设计思路	436
11.6 数据恢复模块设计.....	435	11.6.3 程序开发步骤	436

第 1 章 Visual C++ 6.0 编程基础

1.1 C++语法基础

1.1.1 程序设计语言介绍

1. 软件

计算机内部所有能够存储的各种数据和能够执行的各种程序都称为软件。而程序一词经常有两种理解：(1) 由程序员编写的源代码；(2) 可执行的软件。

程序通常可以分为以下几类：

(1) 操作系统 (Operating System)

为用户管理计算机软硬件的程序。例如 DOS、Windows 98、Windows NT、Windows 2000、Windows CE、Windows XP、UNIX、Mac OS、Solaris 等。

实际上，操作系统包含很多可执行程序，这些程序组合在一起，完成一个或几个特定的任务。这些程序的根本目的在于有效地组织计算机的硬件资源，并为用户提供一个访问硬件的友好界面。通常也可以将其称为系统软件。

(2) 应用软件 (Applications)

在操作系统下执行的，具有特殊用处的程序，如字处理软件 Microsoft Word、游戏软件星际争霸、财务软件等。

这部分程序，也就是程序员主要的服务方向，也是软件设计中商业利润比较大的一部分内容。

(3) 应用程序开发环境 (Application Development Environment)

协助程序员开发应用程序的特殊程序，如 Microsoft Visual Studio、Delphi、PowerBuilder 等。

2. 程序语言

程序语言是人与计算机交流的工具。通过程序语言我们可以编写程序，控制计算机执行相应任务。到目前为止，先后出现了四代程序语言。

□ 第一代语言：机器语言 (machine language)

计算机执行的每一个操作都是由一组特定的二元指令所指定，称为操作码。通常将这些操作码称为机器语言。

该语言不仅难于读写，撰写复杂，而且非常容易出错。

□ 第二代语言：汇编语言 (assembly language)

以接近英文和数学式的方式写作程序，它的语意与机器语言有一对一的对应关系。

该语言能发挥特定机器的硬件功能，编译后的程序运行速度快、效率高，但因仍与机器硬件相关，编写仍然较困难。

□ 第三代语言：高级程序语言（high-level programming language）

包括 Fortran、Pascal、C、C++、Java、Basic、C#等，用这些语言编写应用程序的时候，不需要知道 CPU 执行的细节。这些语言都有自己的特色，能胜任某一方面的程序设计。例如 Fortran 用于科学及工程的方程式运算。而程序语言 C 具备汇编语言的优点，可以直接进行位运算，而且有高度的结构性，代码相对容易维护，可理解性又提高了。C++ 不仅继承了 C 的各种优点，而且引入了面向对象程序设计的思想。

□ 第四代语言：特殊用途语言

使用在特殊的环境中，便于非专业程序设计人员使用的语言，例如 Perl、SQL、MATLAB 等，它们通常都不需要声明变量，而且有很多现成的功能可以套用。

3. 编程思想

编程思想由结构化编程发展到面向对象编程。

结构化编程的主要思想是“把问题细化”，即把现实中的目标分解成一系列的任务，对每一个任务再进行分解，直到每个任务都能被解决为止。这是处理复杂问题的一种非常成功的方法。即使是现在，在处理很多问题上仍会经常用到这个解决问题的思想。当然，这种编程思想存在着问题。首先，它将数据结构从函数中分离出来，这使得程序员在设计程序的时候不得不把一个事物的属性与方法分离开考虑，于是就很难真实地表现现实中的模型。其次，代码的可重用性不高。虽然，现在用的很多函数都是用 C 语言编写的，但是，现在需要的可重用的代码已经不再是这个层次上的代码，而是集成了很多功能和数据的完整的组件，并且要保证其可维护。

面向对象的编程思想满足以上的种种需求，它提供一种方法，可实现软件组件的可重用性，并将数据与操纵数据的任务结合起来。

面向对象编程的实质就是将“对象”作为模型，而不是“数据”。用作模型的对象可以是屏幕上的界面，如按钮、列表框，也可以是现实世界中的对象，如自行车、人等。

面向对象的编程思想必须要涉及到封装、继承、多态性这 3 种技术，C++ 完全支持这 3 种技术。

□ 封装

实际的事物有很多组成部分，描述一个事物也可以从不同的角度出发，从程序设计的角度出发，需要两个元素——数据和函数，当从这两个元素出发去描述一个事物时，会看到：一个事物本身具有一些属性，相应地存在一些行为能改变其中的一些属性。要用程序中的对象来代表这些实际的事物，于是，用数据代表事物的属性，用函数反映事物的行为。这样，与该对象相关的数据和行为就被封装在这个对象里了。

例如：一个人有肤色、身高、心跳频率等属性，有自己的说话、跑、跳等行为。但有时，属性并不是实际存在的一个事物特征的反映，它可能是一种人为抽象出来的状态的反映，例如：一个人有静止、说话、跑步、睡觉 4 种状态，可以定义一个属性来表示当前这个人处于什么状态。假设这个人处于百米跑起跑前的静止状态，当发令枪响起，这个人开始跑（即发出跑的动作），同时状态属性发生变化，由静止状态变为跑步状态；还有其他的属性也随着这个跑的动作的发出而发生变化，如心跳频率加快、移动速度变快等。这个过程即反映了行

为改变属性这一特征。

程序并不一定要完全模拟实际的事物。在系统分析时，希望设计结果能真实反映实际问题，但是，也没有必要对涉及到的所有的事物都建立模型。下面针对系统分析过程中可能遇到的问题进行举例。一个顾客染发，染发的动作会改变头发的颜色属性。显然，染发的动作由理发师发出，但在程序中是否真的由理发师发出这个动作还需要根据具体问题再进行分析。

(1) 当具体问题中理发师的存在仅仅完成一个染发的作用，那么要考虑理发师是否有存在的必要，如果没必要，就果断地删除这个对象，把染发的动作交给顾客自己来完成。

(2) 当具体问题是描述理发师的行为状态变化，那么可能就把这个染发的动作交给理发师来完成。

该例子也说明，行为交给哪个对象来完成，是需要就具体问题进行分析而得到的，关于这个问题可以参看下列规则：

(1) 如果行为影响或修改了一个对象，最好把该行为交给目标对象（而不是动作的发起者）。

(2) 如果行为涉及到多个对象以及它们之间的关联，从中找出处于中心地位的对象，将行为交给该对象完成。

这两个规则只是经验的反映，经验不能反映所有的情况，也有可能把行为交给其他对象来完成也是合理的，这需要根据实际的情况进行判断。

□ 继承

这是代码重用的很有效的方法，新的类可以通过继承原有类，并选择性地增加或修改其中的属性或行为，以达到利用原有类的目的。现介绍一下代码重用的方式。

注意：类是与对象相关联的概念，将在稍后介绍。

① 源代码剪贴。最原始的形式，缺点多多。首先是复制或修改原有代码可能会出错，其次需要对源代码有一定程度的了解。另外，存在严重的配置管理问题，人们几乎无法跟踪源代码多次修改重用的过程。

② 源代码包含。许多程序设计语言都提供包含库中源代码的机制。使用这种重用形式时，配置管理问题有所缓解，因为修改了库中源代码之后，所有包含它的程序自然都必须重新编译。

③ 继承。利用继承机制重用类库中的类时，无需修改已有的代码，就可以扩充或具体化原有类，因此，基本上不存在配置管理的问题。

□ 多态

允许使用相同的接口，与各种不同的派生类定义出来的对象交互，能够产生正确的行为。

例如，中国人说汉语，美国人说英语。“说”的动作相同，却有着不同的内容——汉语和英语；对于不同类型的电视机，都可以使用相同的动作：单击【播放】按钮，开始播放电视频道，显然不同类型的电视机显示的方法会有所区别，但没必要关心这些，人们看电视的目的已经达到了。

1.1.2 C++语法基础知识

在读这本书之前，读者最好看看 C++这方面的专著。这里只讲一些常用的概念，并不对 C++作全面深入的讲解。

C++程序由对象、函数、变量及其他组件组成。

从最简单的程序讲起：

```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
    cout<<"Hello World\n";
```

```
    return 0;
```

```
}
```

这是一个 Console 程序,撰写 Console 程序时需要注意:主程序为 main,可以使用 Cruntime 函数和不涉及 GUI 的 Win32 API 函数。撰写 Console 程序是学习 C++的第一步。

进入 Console 模式进行编程,其操作步骤如下:

(1) 选择【File】菜单中的【New...】菜单项,在弹出的对话框中选择【Projects】标签。再选择“Win32 Console Application”程序,然后输入工程名称,单击【OK】按钮;在接下来的对话框中,为了脱离 VC 提供的代码支持,选择“an empty project”,然后单击【Finish】按钮,会得到一个不包含任何工程文件的工程。之后,要加入包含着主函数的头文件。

(2) 选择【File】菜单中的【New...】子菜单,在弹出的对话框中选择【Files】标签,再选择“C++ Source File”,选中“Add to Project”项,选中(1)中的工程名称。接下来,确定文件的名称,单击【OK】按钮,于是就得到了需要的文件,同时可以在这个空文件里键入上述代码。

(3) 单击  运行程序,它会弹出一个对话框提示没有可执行程序,询问是否创建可执行程序。单击【是】即可,然后 VC 开始编译,链接目标文件。如果没有意外的话,会得到一个 DOS 窗口,窗口内有如下的输出:

```
Hello World
```

```
Press any key to continue
```

最后的一句话并不是源代码反映的内容,这是由编译器提供的,方便查看输出内容的提示。如果读者想看程序真正的输出结果,就进入命令行下(即以前的 DOS 窗口下),运行该程序,读者就可以看到 Hello World,而没有其他的信息。

#include<iostream.h>该语句的意思是:将文件包含到当前的文件中,符号#是预处理标志。

注意:每次启动编译器时,先运行预处理器。预处理器浏览源代码,找到以#开头的行,处理这些行,预处理的作用是改变源代码的文本。结果生成一个新的源代码文件——一个通常看不到的临时文件,但读者可以指定编译器保存它,这样,读者可以在感兴趣的时候或者需要的时候检查它。编译器不读取最初的源代码文件,它读取预处理器输出的结果并编译该文件,最终生成可执行程序。

初步介绍预处理作用如下,以便读者初步认识预处理的概念:

① 包含另一个文件(通常是头文件),为了引入需要的代码。

② 定义符号,起到开关的作用,可以根据机器的情况、操作系统的情况及用户的需求来决定哪部分代码有效,例如,如果要在 Win32 的环境下编程,那么,就定义 Win32 这个符号。

③ 定义宏,简化常用的数据,或者简化复杂的函数声明、定义的过程。

预处理的功能不止这些，以上是本书程序中略有涉及的内容。

正式的程序从 `main` 函数开始，每一个 C++ 程序都有一个 `main` 函数。函数是指能实现一个或多个功能的代码块。通常函数是由其他函数调用或激活，而 `main` 属于特殊情况。程序在开始的时候自动调用 `main`。从本质上讲，也是被调用，不过，那些都不是现在需要关心的事情。

所有的函数都以左大括号开始，以右大括号结束。两个大括号之间是函数体。

使用对象 `cout` 将一个字符串打印到屏幕上。

最后返回 0。

这些就是一个 C++ 程序的基本轮廓。下面开始了解 C++ 的基本语法。

1. 基本数据类型

在任何一台计算机中，每种变量类型都占据一定量的内存单元。但并不是每种变量类型占的内存大小在每台计算机上都相同，一个整型变量在一台机器上可能是 2 字节，而在另一台机器上可能是 4 字节，但对任一台机器而言，这个值是确定的，并不随时间而变化。字符型变量 `char`（用于保存字符）通常只有 1 字节。

在大多数计算机上，短整型是 2 字节，长整型是 4 字节，而整型可能是 2 字节也可能是 4 字节。编程语言并没有对此做出精确的定义，它定义的是短整型必须小于或等于整型的大小，整型必须小于或等于长整型的大小。

整型的大小由读者所使用的处理器（16 位还是 32 位）和编译器决定的。在使用 Visual C++4.0 及以上版本）的 32 位（奔腾）计算机上，整型为 4 字节。

变量类型如表 1.1 所示。

表 1.1 变量类型

类 型	大 小	值
<code>bool</code>	1 字节	<code>true</code> 或 <code>false</code>
<code>unsigned short int</code>	2 字节	0~65 535
<code>short int</code>	2 字节	-32 768~32 767
<code>unsigned long int</code>	4 字节	0~4 294 967 295
<code>long int</code>	4 字节	-2 147 483 648~2 147 483 647
<code>int(16 位)</code>	2 字节	-32 768~32 767
<code>int(32 位)</code>	4 字节	-2 147 483 648 ~ 2 147 483 647
<code>unsigned int(16 位)</code>	2 字节	0 ~ 65535
<code>unsigned int(32 位)</code>	4 字节	0 ~ 4 294 967 295
<code>char</code>	1 字节	256 个字符
<code>float</code>	4 字节	1.2e-38 ~ 3.4e38
<code>double</code>	8 字节	2.2e-308 ~ 1.8e308

2. 表达式和语句

在 C++ 中，语句用于控制程序的执行顺序、计算表达式的值，或者什么都不做（空语句）。

所有的 C++ 语句都以分号结尾, 即使空语句也是如此。

典型的语句:

```
value=value2-value3;
```

C++ 中任何一个计算值的操作都可称为表达式, 表达式总能返回一个值。表达式可以简单到 1、2、3 这样的数, 也可以是 $a+b$ 等形式。重要的是它能返回一个值。

运算符是一种能使编译器进行某项操作的符号。运算符作用于操作数, 在 C++ 中所有操作数都是表达式。

(1) 赋值运算符 (=): 使赋值运算符右边的操作数的值改变为赋值运算符左边的值。例如: $value=24$ 。

(2) 数学运算符: 加 (+)、减 (-)、乘 (*)、除 (/)、取模 (%)。取模运算符 (%) 就是求整型除法的余数。

① 一些特殊的运算符: 如, +=, -=, *=, /=, %=。

例如:

$a+=2$; 等价于 $a=a+2$;

$a-=2$; 等价于 $a=a-2$;

$a*=2$; 等价于 $a=a*2$;

$a/=2$; 等价于 $a=a/2$;

$a\%=2$; 等价于 $a=a\%2$;

② 单目运算符: 如++, --。

$a++$; 和 $++a$; 等价于 $a=a+1$;

$a--$; 和 $--a$; 等价于 $a=a-1$;

$x=a++$; 等价于两步操作: 先是 $x=a$; 然后是 $a=a+1$;

$x=++a$; 等价于两步操作: 先是 $a=a+1$; 然后是 $x=a$;

在复杂的表达式之中, 多个运算符同时出现, 不同的符号有不同的执行优先级。例如: $a=5-4*6$, 对于该表达式, 由于 * 号的优先级大于 - 号, 所以先执行 $4*6$, 然后执行 $5-24$, 最后执行 $a=-19$ 。可以通过添加括号 () 来改变运算顺序。将原式改成 $a=(5-4)*6$, 这时, 先执行 $(5-4)$, 再执行 $1*6$, 最后执行 $a=6$ 。

(3) 关系运算符: 用来对两个量进行比较的 (等于、大于或小于) 符号。每条关系语句的值要么为真要么为假, 每个表达式都可以按真或假来求值。凡是数值运算结果为 0 的表达式返回假, 否则都返回真, 也就是返回 False 或 True。C++ 基本的关系运算符如表 1.2 所示。

表 1.2

关系运算符

名 称	运 算 符	例 子	值
等于	==	$1==1$	true
		$1==2$	false
不等于	!=	$1!=1$	false
		$1!=2$	true
大于	>	$1>2$	false
		$2>1$	true