

21世纪高等学校规划教材 | 计算机应用



Visual C++ Windows 编程技术

曹 旻 编著

清华大学出版社

21世纪高等学校规划教材 | 计算机应用

Visual C++ Windows 编程技术

曹旻 编著

清华大学出版社
北京

内 容 简 介

本书介绍使用 Visual C++ 进行 Windows 应用程序设计的方法。全书分 10 章。内容包括 Windows 编程基础、Visual C++ 集成开发环境简介、Visual C++ API 编程初步、Windows API 应用的输入和输出、资源、Visual C++ MFC 编程初步、Windows MFC 应用程序的输入和输出、MFC 应用程序中的资源使用、文档的序列化、数据库访问技术。

本书对于从事 C++ 教学和开发的人员,有较高的参考价值。

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

Visual C++ Windows 编程技术/曹旻编著. —北京:清华大学出版社,2011.11

(21 世纪高等学校规划教材·计算机应用)

ISBN 978-7-302-26318-0

I. ①V… II. ①曹… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 149982 号

责任编辑:闫红梅 顾 冰

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:25.25 字 数:632 千字

版 次:2011 年 11 月第 1 版 印 次:2011 年 11 月第 1 次印刷

印 数:1~3000

定 价:39.00 元

产品编号:041470-01

编审委员会成员

(按地区排序)

清华大学

周立柱 教授
章 征 教授
王建民 教授
冯建华 教授
刘 强 副教授

北京大学

杨冬青 教授
陈 钟 教授
陈立军 副教授

北京航空航天大学

马殿富 教授
吴超英 副教授
姚淑珍 教授

中国人民大学

王 珊 教授
孟小峰 教授
陈 红 教授

北京师范大学

周明全 教授

北京交通大学

阮秋琦 教授

北京信息工程学院

赵 宏 副教授

北京科技大学

孟庆昌 教授

石油大学

杨炳儒 教授

天津大学

陈 明 教授

复旦大学

艾德才 教授

吴立德 教授

吴百锋 教授

杨卫东 副教授

同济大学

苗夺谦 教授

华东理工大学

徐 安 教授

华东师范大学

邵志清 教授

杨宗源 教授

东华大学

应吉康 教授

乐嘉锦 教授

浙江大学

孙 莉 副教授

吴朝晖 教授

扬州大学
 南京大学

 南京航空航天大学

 南京理工大学
 南京邮电学院
 苏州大学

 江苏大学
 中国矿业大学
 武汉大学
 华中科技大学
 中南财经政法大学
 华中师范大学

 江汉大学
 国防科技大学
 中南大学
 湖南大学

 西安交通大学

 长安大学
 哈尔滨工业大学
 吉林大学

 山东大学

 中山大学
 厦门大学
 仰恩大学
 云南大学
 重庆大学
 电子科技大学

 成都理工大学

 西南交通大学

李善平 教授
 李 云 教授
 骆 斌 教授
 黄 强 副教授
 黄志球 教授
 秦小麟 教授
 张功萱 教授
 朱秀昌 教授
 王宜怀 教授
 陈建明 副教授
 鲍可进 教授
 张 艳 教授
 何炎祥 教授
 刘乐善 教授
 刘腾红 教授
 叶俊民 教授
 郑世珏 教授
 陈 利 教授
 颜 彬 教授
 赵克佳 教授
 刘卫国 教授
 林亚平 教授
 邹北骥 教授
 沈钧毅 教授
 齐 勇 教授
 巨永峰 教授
 郭茂祖 教授
 徐一平 教授
 毕 强 教授
 孟祥旭 教授
 郝兴伟 教授
 潘小轰 教授
 冯少荣 教授
 张思民 教授
 刘惟一 教授
 曾 一 教授
 刘乃琦 教授
 罗 蕾 教授
 蔡 淮 教授
 于 春 副教授
 曾华燊 教授

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21世纪高等学校规划教材·信息管理与信息系统。

(6) 21世纪高等学校规划教材·财经管理与应用。

(7) 21世纪高等学校规划教材·电子商务。

(8) 21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail: weijj@tup.tsinghua.edu.cn

前言

Windows 是目前计算机领域中占有率近 90% 的操作系统平台,是目前和将来的流行和发展主方向之一,在古老的 DOS 操作系统已基本被淘汰的时候,一方面,计算机的许多专业基本课程都多少涉及一些编程,需要了解和掌握一些基本的 Windows 技术和编程方法;另一方面,在计算机专业中需要有其他一些专业基础课程的知识,才能更好地学习和理解 Windows 技术和编程方法。因此 Windows 程序设计既是专业课程的基础课程,又是技术性很强的专业课程。

本教材是面向计算机专业学生的教学用书,必须有一定的 C/C++ 语言程序设计基础,如果有其他语言的 Windows 开发知识,如 Java、C#、Delphi 等应用的开发基础和经验,则更好。

本书是作为专业选修课而配套的,也就是说,不作为计算机专业的入门语言课程来设置,其内容着重于 Windows 应用的开发,而不是像入门语言那样侧重于 C++ 的语法和常规算法。所以,本书的内容编排和编写风格上都与经典的教材有所区别:第一,基本语法和常规算法不讲;第二,不像一般的教材那样严谨;第三,内容较广,但不深入,对于程序的基本结构和编程的基本原理比较注重,但很多编程技术并不深入展开,有“知其然而不知其所以然”的嫌疑;第四,以代码“反推”编程原理,不是在讲解了所有的理论知识之后才开始编程实践,而是根据代码反过来讲解程序的基本结构和编程原理;第五,所有的实例未必是“经典”的,希望能达到的目的是,既能体现其所在章节中着重讲解的编程原理,又具有实用性和实验指导性。

本书每一章都配有相关的习题,书中所有例题采用 Visual Studio. NET 2005 编制,并调试通过。因此,它既可以作为学生的教材,也可以作为开发人员的参考书。

在书的写作过程中,作者始终贯彻教材规划的原则:突出实用性,不能只讲原理,而要重视实践,既有一定的原理,又要有具体的实例,便于读者逐步地进行 Visual C++ 的 Windows 应用程序的开发。因此,本书既能作为本科高年级的选修课教材,也适合高职、高专院校作为培养学生多种技能的教科书。

本着探索、分享和进取的精神,本书记录了我们对 Visual C++ 技术的理解、心得和体会,并将大量原创性习题和例题献给读者。但也正因为如此,作者学识和经验的浅薄和局限暴露无遗,所以我们真诚地期待着读者直率批评和慷慨赐教。

书中有大量的程序,在编辑排版方面,我们将其中需要较多讲解的程序按语句顺序加入行号,对于较大的程序则按原貌给出,给读者一个完整的程序结构。程序中通常有空行以及仅有 { 或者 } 的行,我们在不影响程序结构清晰性的情况下,予以压缩以节约版面;显示程序运行结果采用直接复制来显示结果窗口的方法。

最后,向所有帮助和关心本书编写的同事致以衷心的感谢,向清华大学出版社的领导和编辑致以衷心的感谢。

作 者

2009-09-20 于上海



目 录

第 1 章 Windows 编程基础	1
1.1 Windows 应用的特点	1
1.2 面向对象程序设计基础	4
1.2.1 类和对象	4
1.2.2 抽象与封装	6
1.2.3 类与类之间的关系	6
1.2.4 多态	8
本章小结	9
习题和思考题	9
第 2 章 Visual C++ 集成开发环境简介	10
2.1 Visual Studio. NET 结构概貌	10
2.2 Visual Studio. NET 核心成分	12
2.3 Visual Studio. NET 中开发和调试 Windows 应用	13
2.3.1 Visual C++. NET 中开发 Windows 应用	13
2.3.2 Visual C++. NET 基本调试方法	16
本章小结	20
习题和思考题	20
第 3 章 Visual C++ API 编程初步	21
3.1 Windows 应用的基本类型	21
3.1.1 Win32 项目	21
3.1.2 MFC 项目	22
3.1.3 CLR 项目	23
3.1.4 Windows 应用程序组成	24
3.1.5 本书采用的开发方法	25
3.2 Windows 应用程序的基本构成	26
3.2.1 Win32 控制台应用和 Windows 应用的比较	26
3.2.2 Win32 应用的基本构成成分	29
3.3 多窗口的建立	47
本章小结	54
实验和习题	54

第 4 章 Windows API 应用的输入和输出	64
4.1 输出	64
4.1.1 与设备无关的图形输出与 GDI	65
4.1.2 GDI 绘图	71
4.2 输入消息的处理	83
4.2.1 键盘消息和字符消息	83
4.2.2 鼠标消息的处理	95
4.2.3 定时器消息的处理	102
4.2.4 其他输入方式	105
本章小结	114
实验和习题	114
第 5 章 资源	123
5.1 资源简介	123
5.1.1 资源的定义和本质	123
5.1.2 资源的使用方法	123
5.2 Icon、Cursor、Bitmap 和 String	125
5.2.1 Icon 资源及其使用	125
5.2.2 Cursor 的使用	128
5.2.3 BitMap 的显示和处理	132
5.2.4 String 资源及其应用	138
5.3 菜单和加速键	139
5.3.1 菜单的定义和加载	140
5.3.2 应用程序使用菜单的方法	143
5.3.3 应用程序对菜单项的响应	144
5.3.4 菜单项的操作	144
5.3.5 加速键的使用	145
5.3.6 应用实例	146
5.4 对话框	155
5.5 其他子控制	165
本章小结	167
实验和习题	168
第 6 章 Visual C++ MFC 编程初步	174
6.1 MFC 简介	174
6.2 SDI 应用、MDI 应用和基于对话框的应用	183
6.2.1 MFC 的文档/视图结构	183
6.2.2 Windows 应用类型	184

6.3 MFC 应用程序的基本成分	184
6.3.1 最简单的 MFC 应用的组成	185
6.3.2 基于对话框的 MFC 应用的组成	190
6.3.3 SDI 的 MFC 应用的组成	194
6.3.4 MDI 的 MFC 应用的组成	207
本章小结	214
实验和习题	214
第 7 章 Windows MFC 应用程序的输入和输出	216
7.1 输出	216
7.1.1 图形输出相关的类	216
7.1.2 输出消息映射及 ClassWizard 的使用	219
7.1.3 GDI 绘图	224
7.2 输入消息的处理	235
7.2.1 键盘消息和字符消息的处理	235
7.2.2 鼠标消息的处理	242
7.2.3 定时器消息的处理	249
本章小结	255
实验和习题	256
第 8 章 MFC 应用程序中的资源使用	260
8.1 简单资源的使用	260
8.2 菜单 Menu 的使用和处理	270
8.2.1 菜单资源的加载和使用	270
8.2.2 菜单项的响应处理	276
8.2.3 文档/视图结构处理菜单项的特殊性	279
8.2.4 简单资源的综合使用实例	282
8.3 对话框资源和对话框类	291
8.4 控件的使用	297
8.4.1 控件简介	297
8.4.2 DDX 和 DDV 机制	300
8.4.3 控件消息的处理	304
本章小结	306
实验和习题	307
第 9 章 文档的序列化	312
9.1 文档/视图结构	312
9.2 对象及文档的序列化	315
9.2.1 文档的序列化	315

9.2.2 文档中对象的序列化.....	321
9.2.3 文档类型的确定.....	329
9.3 多种文档模板指定	331
• 本章小结.....	336
实验和习题.....	337
第 10 章 数据库访问技术	344
10.1 基本概念	344
10.2 使用 ODBC 访问数据库	347
10.2.1 MFC ODBC 类.....	347
10.2.2 建立 ODBC 应用的步骤	350
10.2.3 数据操作和处理	358
10.3 使用 ADO 访问数据库	372
10.3.1 ADO 对象.....	372
10.3.2 用 ADO 技术访问数据库的步骤	380
10.3.3 实例	381
本章小结	390
实验与习题	390
参考文献.....	392

Windows 编程基础

本章简单介绍 Windows 应用的特点和面向对象程序设计的基本概念。

1.1 Windows 应用的特点

Windows 操作系统起源于 MS-DOS(Microsoft Disk Operating System, 微软磁盘操作系统), MS-DOS 与现在使用的 Windows 操作系统的最大区别在于, 它是命令行形式的, 靠输入命令来进行人机对话, 并通过命令的形式把指令传给计算机, 让计算机实现操作。从实质上来说, 包括 Windows 95 和 Windows 9x 的系列版本, 这些看起来完全图形化的操作系统也完全是建立在 MS-DOS 之上的, 在 Windows 系列的任意一个版本中都可以找到 DOS 的影子。例如, 在 Windows 2000/NT 的开始菜单中的运行程序中输入 cmd 命令, 可进入命令行界面。因此, 谈到 Windows 应用的特点, 通常是和原来的 MS-DOS 应用相比较的。而现在, 我们基本上已经用 Windows 取代了 MS-DOS, 归纳一下, Windows 应用具有下列的特点。

1. 面向对象的特点

在程序设计课程中谈到 OO(Object Oriented, 面向对象) 概念, 大家第一个反应就是面向对象程序设计, 甚至马上会想到 C++、Java、C# 等语言, 因为这些语言都强调面向对象的编程。在编程语言的发展过程中, 语言本身有面向过程的, 如传统的 Pascal、FORTRAN、C 等语言, 后来随着 OO 思想的发展, 出现了 OO 语言, 如目前流行的 C++、Java、C# 等。但是, 这里所说的面向对象的特点, 并不是指设计思想、编程方法或者语言, 而是针对 Windows 应用本身。这样说可能让大家不理解, 下面来看一个例子。如图 1-1 所示, 图 1-1(a) 是一个控制台应用的运行结果, 图 1-1(b) 是 Windows 操作系统附件中自带的“记事本”(即 Notepad)。控制台窗口提供了一种标准的基于文本的界面和命令提示环境, 它的界面即模拟了 MS-DOS 应用的运行界面, 其输入是命令行的形式, 而其输出则是字符形式, 图 1-1(a) 的黑色部分模拟了标准的输出设备, 即屏幕, 该应用在标准输出设备上输出了“Welcome!”, 人机交互的界面就是屏幕。图 1-1(b) 所显示的记事本应用的人机交互界面是一个窗口, 窗口本身就是一个对象, 窗口的上部有菜单, 如“文件”、“编辑”等, 菜单本身也可以看作是对象。这里, 我们并不关心记事本应用是采用什么语言开发的, 所以, Windows 应用本身具有面向对象的特点, 而不是说 Windows 应用是采用面向对象的语言开发的, 这就是所谓的

Windows 应用的面向对象性。图 1-1(a)所示的控制台应用采用 C++ 语言开发,在 Visual Studio. NET 2005 中编译运行,它是将字符输出在“屏幕”上,它的界面就是字符形式,应用本身并不具有面向对象的特点。

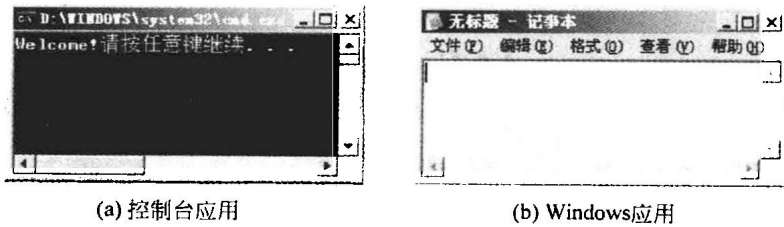


图 1-1 控制台应用和 Windows 应用的比较

2. 消息传递机制和事件驱动机制

传统的 MS-DOS 应用是面向过程的,因此程序的执行是按照程序代码的命令顺序进行的,最多是响应键盘输入等,这样的程序设计方法是面向程序而不是面向用户,交互性较差,用户界面不够友好,因为它强迫用户按照某种不可更改的模式进行工作;Windows 操作系统是以消息为基础、事件驱动的,事件驱动程序设计是一种全新的程序设计方法,它不是由事件的顺序来控制,而是由事件的发生来控制,而这种事件的发生是随机的、不确定的,并没有预定的顺序,这样就允许应用程序的用户用各种合理的顺序来安排应用的流程。对于需要用户交互的应用程序来说,事件驱动的程序设计有着过程驱动方法无法替代的优点,它是一种面向用户的程序设计方法,在程序设计过程中除了完成所需功能之外,更多地考虑了用户可能的各种输入,并针对性地设计相应的处理程序。Windows 应用对用户的输入的响应方式主要包括消息循环和事件响应两种机制。事件机制和消息机制从本质上来说是类似的,例如,用户单击鼠标左键,在消息机制中是产生了一条“单击鼠标左键”的消息,应用应该对此消息作处理;而在事件机制中,则是发生了一个“单击鼠标左键”的事件,应用需要处理该事件。当然,不同的机制,其处理过程和原理有所不同,不同的语言采用的机制也有所不同,Visual C++ 采用的是消息队列的方式,而 Visual Basic、Java 和 Visual C# 则采用事件处理的方式。

3. 多任务和多线程

和 MS-DOS 相比,Windows 是多任务操作系统,它能同时运行两个或多个程序,Windows 的每一个应用程序对屏幕的一部分进行处理,程序共享 CPU,从技术上来讲,它们并不同时运行。从 Windows 98 版本开始,Windows 操作系统支持两种类型的多任务:基于进程的多任务和基于线程的多任务。因此,传统的 MS-DOS 应用是单任务的,每次只能运行一个程序,而 Windows 应用是多任务的,每次既可以同时运行多个程序(进程),也可以运行同一个程序的多个实例,还可以同时运行同一个进程的多个线程。

图 1-2 是截取的一个屏幕界面,在该图中,Windows 操作系统下同时运行了两个应用:一个是 Visual Studio. NET 2005 的 IDE(Integrated Development Environment,集成开发环境),另一个是显示 Welcome 的控制台应用。其实,此时在 Windows 操作系统下也同时运行了 Welcome 的控制台应用的两个实例,如图 1-2 中部的两个黑色控制台界面所示,它

们的源代码完全相同,都在 Visual Studio. NET 2005 的 IDE 中编译、链接,然后运行,这就是同一个应用的两个实例。特别注意的是,控制台程序就是模拟传统的 MS-DOS 应用的,图 1-2 中部的两个黑色控制台界面是 Windows 操作系统下同一个应用的两个实例,即两个控制台,而该控制台应用本身仍然是单任务的,也就是说,在同一个控制台下,一次只能运行一个应用,而不能同时运行两个或两个以上的应用。这就是我们所说的控制台应用是单任务的,而 Windows 应用是多任务的。

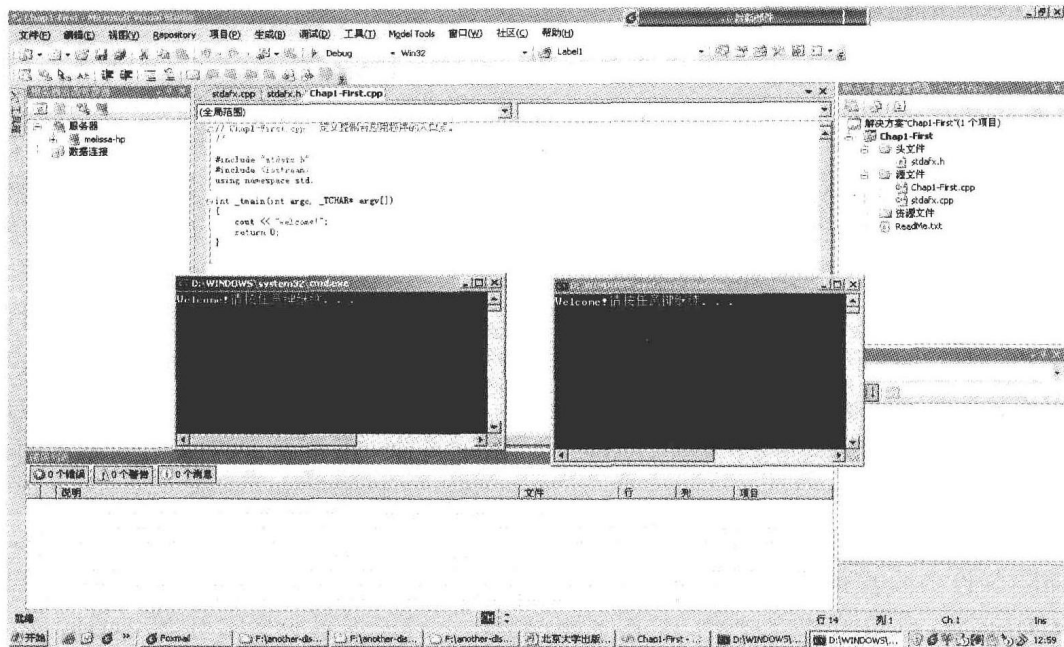


图 1-2 同一个控制台应用的两个运行实例

4. 多形式的输入输出手段

Windows 应用是基于 GUI(Graphical User Interface, 图形用户界面)的,用户使用 GUI 与程序进行可视化的交互,Windows 应用的输入输出手段与传统的 MS-DOS 应用程序和基于控制台的应用相比要丰富得多。

基于控制台的应用程序是文本输入,即用户通过键盘进行输入,Windows 应用的输入手段包括硬件方式和软件方式,硬件方式指用户通过鼠标、键盘等物理设备进行输入,软件手段则指用户无须通过任何硬件而仅仅编写一些代码就可以进行输入。在 Windows 应用中,软件输入的典型就是使用定时器(timer)的方式,每隔一个时间间隔就可以进行事先定义好的输入,这种方式适合自动输入,即无须用户直接交互和干预。

硬件输入主要是使用鼠标和键盘,但其输入接收方式和处理方式与传统的文本输入不同。基于控制台的应用通过调用函数(如 cin)得到用户的输入值,而 Windows 应用则是通过响应消息或处理事件来获取用户的输入。所以,在 Windows 应用中,用户通过鼠标进行输入的方式包括直接在用户区域单击/双击鼠标、单击菜单项或工具条中的菜单按钮、在各种控件(control)上单击/双击鼠标等;用户通过键盘进行输入的方式包括直接在用户区域(能获得输入焦点)进行键盘输入和通过各种控件(能获得输入焦点)进行键盘输入等。控

件在 Visual C++ 中习惯称为子控制窗口(child control window),简称子控制;在 Visual Basic 中通常称为控件;而在 Visual C# 中则称为 GUI 组件、控件或者窗口小部件(windows gadget)。我们将它们称为控件。控件是一组对象,例如按钮、滚动条等,用户可以使用键盘或者鼠标与之交互。

基于控制台的应用程序通过调用函数(如 cout)在屏幕或模拟屏幕的控制台中输出字符,而 Windows 应用的输出方式类似于其输入方式,也包括直接在用户区域进行图形输出和在各种控件中进行输出。

5. 设备无关的图形输出

Windows 应用的输出也和传统的 MS-DOS 应用以及控制台应用有所不同。首先,MS-DOS 应用独占整个显示屏幕,其他应用在后台等待;而 Windows 是一个多窗口的操作系统,由操作系统来统一管理屏幕输出,Windows 的每一个应用程序对屏幕的一部分进行处理,每个窗口要输出内容时,必须首先向操作系统发出请求,由操作系统完成实际的屏幕输出工作。其次,Windows 应用的所有输出都是图形,Windows 提供了丰富的图形函数用于图形输出,对输出图形是相当方便的,但由于字符也被作为图形来处理,输出时的定位要比 MS-DOS 复杂得多。最后,Windows 应用的输出是与设备无关的。传统的 MS-DOS 应用的标准输出设备是屏幕,如果要直接在打印机上输出,例如,输出报表,则需要针对不同的打印机在程序中插入不同的打印控制码,这样的程序编写起来烦琐,而且不容易移植;而 Windows 下的应用使用 GDI(Graphics Device Interface,图形设备接口)进行图形输出,GDI 屏蔽了不同设备的差异,提供了设备无关的图形输出能力,Windows 应用只要发出设备无关的 GDI 请求,由 GDI 去完成实际的图形输出操作。对于不同的打印机来说,GDI 只需要将数据传给驱动程序就可以了,然后由驱动程序产生命令绘制出相应的图形。

1.2 面向对象程序设计基础

OOP(Object-Oriented Programming,面向对象程序设计)始于 20 世纪 80 年代、90 年代间,此后,OOP 技术就成为软件设计的最重要的技术之一,是一种相当成熟的软件开发方法。面向对象思想的产生和成熟,建立在计算机硬件性能、软件理论和实践的巨大进步,以及软件资源不断积累的基础之上,面向对象程序设计语言使得程序员能够较直接地模拟现实世界问题域,表明计算机信息世界的模型向现实世界的模型靠拢了一大步。而从人类思维方法的角度来看面向对象程序的思想,其实十分朴实。

本书只介绍一些以后章节用到的概念和技术,希望在一开始能给读者一个简单但清晰的概貌。

1.2.1 类和对象

面向对象程序的基本元素是对象,类是对象的模板,设计类是面向对象程序设计的关键技术。

现实世界中任何实际存在独立的事物都可以称为对象(object),对象可以很大,例如国

家、地球,可以很小,例如分子、原子;对象可以是有形的,例如感冒药、听诊器,也可以是无形的,例如手术方案。在现实生活中,人们很少使用对象这个十分笼统的概念。

为了在程序里描述各种各样的对象,对象必须有共同的结构模型,对象有三个特征:标识、状态和行为。对象标识即对象名,是一个对象区别于其他对象的标识符;对象状态即对象的属性,是描述对象各种特征的信息,是对象的静态特征;对象行为即对象能够提供的服务,对象的行为也可以理解为与对象属性相关的所有合法的操作,是对象的动态特征。在程序中,对象状态表现为一组数据,在不同的文献中有不同的称呼,例如,属性、域、字段、数据成员等。对象行为表现为一组针对数据成员的操作的函数,在不同的文献里,它们也有不同的称呼,例如,函数、成员函数、方法等。

对象太多,在程序中直接地逐一构造各种对象是不堪设想的。而在现实世界里,每个对象都和其他某些对象有共性,可以同属于某一个分类。例如,白马、黑马、蒙古马、阿拉伯马都属于马。图 1-3 描述了从生物分类上概括“马”相关对象的分类。

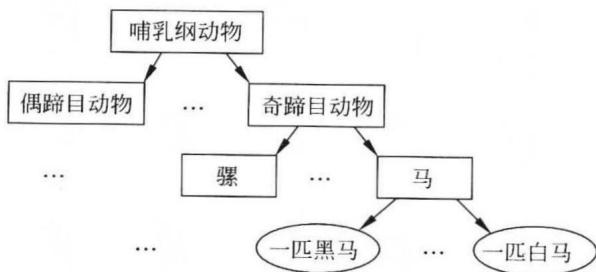


图 1-3 一匹白马和马的分类

在图 1-3 中,唯有“一匹白马”、“一匹黑马”是实实在在的事物,即对象。其他都是对一组对象的共性的描述。在面向对象程序中,像“马”、“奇蹄目动物”、“哺乳纲动物”等这样对事物概括性定义的部分,称为类。

类是抽象数据类型,类有名称,此外,类主要有三种成分:

- 完整描述事物特征的数据的类型声明,这是对对象的静态特征的抽象描述。
- 能够访问并处理数据的所有操作,这是对对象的动态特征的描述。
- 访问权限,对类以及类中定义的数据、方法访问的限制。

“白马非马”论,是中国两千多年前公孙龙《白马论》中的一个重要论题。白马不是马,那是诡辩,但不少学者认为,将一般概念的“马”,和特定的“白马”区分开来,从认识论发展史上看还是有一定进步意义的。我们无意参与讨论,这里只是借用这个故事来说明,在面向对象程序设计中,如果要表达图 1-3 所表达的关系,那么“马”是一个类,是一个概念层次上的描述,而一匹“白马”是该类的一个实例,是活生生存在的,“白马”确实非“马”。

类中只是定义了数据的类型和结构,并没有具体的值,根据类的定义产生一个对象的过程中,必须填入具体的值,哪怕是部分的值,或者是默认值(如 0、空、false 等)也行。所以类是概念性的,对象是具体的。

类的定义要占据一定的存储空间,类中定义方法和类名是存放在一起的。当一个具体的对象产生的时候,面向对象语言系统为这个对象分配存储空间,这个存储空间主要存放该对象的名和数据成员。一个类可以产生许多对象,每个对象拥有自己个性化的数据,也就是