

新世纪高等学校计算机系列教材

Visual C++ 面向对象程序设计

戴光明 李 向 主编



高等教育出版社
中山大学出版社

策划:湖北省计算机学会·武汉高联教科文中心

新世纪高等学校计算机系列教材

Visual C++ 面向对象程序设计

主 编	戴光明	李 向
编 者	邵玉祥	武 云
	彭 雷	胡霍真

高等教育出版社·北京

中山大学出版社·广州

图书在版编目(CIP)数据

Visual C++——面向对象程序设计/戴光明,李向主编.——北京:高等教育出版社,——广州:中山大学出版社,2005.11

(新世纪高等学校计算机系列教材/湖北省计算机学会·武汉高联教科文中心策划)

ISBN 7-309-02601-1

I. V… II. ①戴… ②李… III. C语言—面向对象—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2005)第111600号

内 容 简 介

本书以C++语言为基础,从三个层次系统地介绍了面向对象程序设计的语言、方法,首先较全面地介绍了C++语言的数据类型、表达式、流程控制语句、函数、数组、结构体、指针、引用和I/O流库,其次着重介绍了面向对象程序设计中的类与对象、继承与派生和多态性等概念,最后较详细地介绍了Windows API编程的基本原理、MFC的类层次结构以及MFC编程实践。并且,为便于实践教学,还在最后一章中给出了10个基础实验和2个综合实验供选用。

本书是作者总结多年教学和科研编程的实践经验编写而成的一书,中以浅显的语言讲明晦涩难懂的语法规则,并配以大量的图表和例题,内容叙述深入浅出、详略得当。在每一章的末尾对其重点和难点进行了小结。书中所有示例程序均在Visual C++ 6.0中经过调试和成功运行。

本书适合理工院校计算机类专业作为程序设计课程的教材和教学参考书,也可作为理工院校非计算机专业面向对象程序设计课程的教材和教学参考书。

注:凡需要本书或电子原稿备课者,可与执行编委唐元瑜老师联系(027-87551581/12967148295)。

版权所有 盗印必究

Visual C++——面向对象程序设计

(C) 戴光明 李 向 主编

责任编辑:崔 引 唐 源

封面设计:袁 作

责任校对:高 联

责任技编:潘 隆

出版发行:高等教育出版社

(地址:北京市西城区德外大街4号)

邮编:100011

中山大学出版社

(地址:广州市新港西路135号)

邮编:510275

经 销:广东新华发行集团

武汉高联科教信息有限公司(电话:027-87551581/87550151(带传真))

邮编:130034

印 刷:安陆市鼎鑫印务有限公司

开 本:787mm×1092mm

1/16

印 张:19.25

字 数:430千字

版 次:2005年11月第1版

印 次:2005年11月第1次印刷

印 数:5 000册

定 价:27.00元

《新世纪高等学校计算机系列教材》 编审指导委员会

主 任:卢正鼎(华中科技大学教授、博士生导师)

副 主 任:何炎祥(武汉大学教授、博士生导师,武汉大学东湖分校教授)

编 委:(以姓氏笔画排序)

王 虹(湖北经济学院教授)

王元珍(华中科技大学教授、博士生导师)

毛法尧(华中科技大学教授,华中科大文华学院教授)

尹朝庆(武汉理工大学教授,武汉科大城市学院教授)

叶俊民(华中师范大学副教授、博士)

李 刚(江汉大学教授,江汉大学文理学院教授)

李鸣山(武汉大学教授)

陈 珉(武汉大学教授、博士)

陈传波(华中科技大学教授、博士生导师)

陈建勋(武汉科技大学教授、博士)

陆际光(中南民族大学教授)

周双娥(湖北大学副教授、博士)

汪厚祥(海军工程大学教授、博士)

张彦铎(武汉工程大学教授、博士后)

金先级(华中科技大学教授,华中科大武昌分校教授)

胡金柱(华中师范大学教授、博士生导师,华中师大汉口分校教授)

袁蒲佳(华中科技大学教授,中南民大工商学院教授)

黄求根(武汉科技学院教授)

程元斌(江汉大学副教授)

程学先(湖北工业大学教授)

楚惟善(湖北工业大学教授,湖北工大工程技术学院教授)

谭连生(华中师范大学教授、博士)

熊家军(空军雷达学院教授、博士)

戴光明(中国地质大学教授、博士)

执行编委:唐元瑜(华中科技大学副编审)

前 言

本书是根据教育部《关于进一步加强高校计算机基础教学的意见》的教学基本要求为普通高等学校各专业大学生编写的程序设计教材。

随着计算机程序设计语言、程序设计方法的进步,人们使用计算机解决问题的方法和层次在进一步加深,社会对学生使用计算机能力的要求也在逐步提高。学生不仅要能够编写面向过程的结构化程序,而且要掌握利用面向对象程序设计技术编程,如在像 Visual C++ 这样的可视化开发环境中开发 Windows 应用程序的能力。目前大部分学校在计算机程序设计方面的教学是首先讲授 C 语言程序设计,实验环境是 Turbo C;然后讲授 Windows 编程,实验环境是 Visual C++。按照这种模式进行教学的难点主要来自教授 Windows 编程上的困难。这种困难出自两个方面:一是要讲解的内容太多而课时太少。因为要讲清楚 Windows 编程,除了要讲清楚 Windows 编程思路与 DOS 编程思路上的差异外,还要求学生有面向对象相关的概念。而要在有限的学时(一般是 30~40 学时)内把这些问题讲清楚绝非易事。另一个困难来自教材的选择。相关的书籍要么是不适合作教材,要么是做教材又嫌内容太多。我们编写的这本教材其内容深度及分量是作者们根据多年的教学实践并经过多次反复讨论确定的,应该说是比较适合教与学的。

全书共分 12 章,其主要内容分为三个层次:第 1~5 章较全面地介绍了 C++ 语言的数据类型、表达式、流程控制语句、函数、数组、结构体、指针、引用和 I/O 流库;第 6~8 章着重介绍了面向对象程序设计中的类与对象、继承与派生和多态性等概念;第 9~11 章详细介绍了 Windows API 编程的基本原理、MFC 的类层次结构以及 MFC 编程实践,并在最后给出了两个较综合的程序设计的例子,希望给学生以一个整体的认识。此外,为便于实践教学,在第 12 章中给出了 10 个基础实验和 2 个综合实验,供任课教师选用。本书的授课学时数建议为 30~40 学时。

本书由戴光明、李向主编,其余编者为邵玉祥、武云、彭雷和胡霍真。

本书在编写过程中,得到了湖北省计算机学会及其教育与培训专业委员会、中国地质大学计算机学院,以及《新世纪高等学校计算机系列教材》编审指导委员会等有关领导与专家的大力支持与帮助,在此一并致谢。

由于编者水平有限,加之时间紧迫,书中缺点和差错在所难免,恳请读者批评指正。

编 者

2005 年 8 月

《新世纪高等学校计算机系列教材》

总 序

21 世纪人类已跨入了信息时代,以计算机为核心的信息技术正在迅猛发展,并不断改变着人类社会的工作方式、生产方式、生活方式和学习方式。当今,各行各业的现代化都离不开计算机,各行各业的人们都在学习和使用计算机,而计算机科学技术及其教育本身也在日新月异地发展变化。为了顺应时代的潮流,满足新世纪高等学校计算机教育事业、教学改革和人才培养对高质量特色教材的需求,湖北省计算机学会及其教育与培训专业委员会和武汉高联教科文中心等共同策划、组织并约请华中科技大学、武汉大学、华中师范大学、中国地质大学、中南民族大学、武汉科技大学、海军工程大学、空军雷达学院、湖北大学、湖北工业大学、武汉工程大学、武汉科技学院、江汉大学等高校长期奋斗在教学科研第一线,且具有丰富教学实践经验的部分优秀骨干教师共同编写了这套计算机系列教材。

这套系列教材共 40 余种,主要是根据中国计算机学会教育委员会、全国高等学校计算机教育研究会等联合推出的《中国计算机科学与技术学科教程 2002》(简称“CCC2002 教程”)中的课程体系与课程大纲的要求,进行规划和组织编写的,并主要供高等学校计算机及其相关专业本科或研究生教学使用。此外,本系列教材中也还包含了一部分适用于各类普通高校培养应用型计算机专业人员和适用于计算机基础教育的教材。

当今,计算机科学技术突飞猛进地向前发展,计算机新技术和新产品不断涌现,高等教育事业和教学改革不断深化,国内教育逐步与国际教育接轨,社会对计算机专业人才的要求越来越高,等等。面对这些新形势,这套系列教材以培养学生具有较扎实的专业基础理论知识、实践能力、创新能力和较高的综合素质能力为目的,既注重知识的更新与合理的结构,又注意学习和汲取国内外优秀教材的优点与精华,并尽力反映国内外最新的教学科研成果及作者们宝贵的实践经验。

我相信,通过作者们的共同努力,定能将这套系列教材打造成为一套既具有时代特色,又非常适用的、高质量的系列教材,为我国高等教育事业的发展和高素质专业人才的培养作出应有的贡献。

湖北省计算机学会理事长

《新世纪高等学校计算机系列教材》

编审指导委员会主任

卢正鼎

2003 年 7 月

目 录

第1章 C++语言基础	(1)
1.1 C++基础知识	(1)
1.1.1 数据类型	(1)
1.1.2 C++的关键字	(2)
1.1.3 标识符、标点符号	(2)
1.1.4 常量、变量	(2)
1.2 运算符和表达式	(6)
1.2.1 算术运算符和算术表达式	(6)
1.2.2 关系运算符和关系表达式	(7)
1.2.3 逻辑运算符和逻辑表达式	(8)
1.2.4 位运算符	(9)
1.2.5 赋值运算符和赋值表达式	(11)
1.2.6 其他运算符和表达式	(11)
1.2.7 类型转换	(13)
1.3 基本输入/输出	(14)
1.3.1 输入/输出流	(14)
1.3.2 格式化输入/输出	(18)
本章小结	(19)
习题一	(20)
第2章 结构化程序设计	(21)
2.1 C++的语句及程序结构	(21)
2.1.1 C++语言的语句	(21)
2.1.2 程序的三种基本结构	(22)
2.2 分支结构语句	(23)
2.2.1 条件语句	(24)
2.2.2 条件运算符	(27)
2.2.3 switch 语句	(28)
2.3 循环结构语句	(30)
2.3.1 while()语句	(30)
2.3.2 do...while()语句	(31)
2.3.3 for()语句	(32)
2.4 程序控制语句	(34)
2.4.1 break 语句	(34)

2.4.2	continue 语句	(35)
2.4.3	goto 语句	(36)
本章小结		(36)
习题二		(37)
第 3 章	数组、指针与字符串	(39)
3.1	数组	(39)
3.2	指针	(41)
3.3	动态内存分配	(46)
3.3.1	new 运算和 delete 运算	(46)
3.3.2	动态内存分配与释放函数	(49)
3.4	字符串	(49)
3.4.1	用字符数组存储和处理字符串	(49)
3.4.2	字符串指针	(50)
3.5	结构体	(51)
3.5.1	结构体变量定义	(51)
3.5.2	结构体变量的使用	(52)
本章小结		(53)
习题三		(53)
第 4 章	函数	(57)
4.1	函数的定义、调用和说明	(57)
4.1.1	函数定义	(57)
4.1.2	函数调用	(58)
4.1.3	函数说明	(59)
4.2	函数间参数传递	(60)
4.2.1	传值参数	(60)
4.2.2	传址参数	(61)
4.3	函数重载	(64)
4.4	函数模板	(66)
4.5	作用域与存储类型	(67)
4.5.1	作用域	(67)
4.5.2	存储类型	(70)
4.6	程序的多文件组织	(75)
4.6.1	多源文件程序结构	(75)
4.6.2	源文件间的相互引用	(75)
4.6.3	函数和变量的连接性	(76)
本章小结		(77)
习题四		(78)

第5章 C++的I/O 流库	(81)
5.1 流	(81)
5.2 标准输入/输出流	(82)
5.3 输入/输出中的格式化	(83)
5.3.1 用ios 类成员函数格式化	(83)
5.3.2 用操作符函数格式化	(85)
5.3.3 自定义操作符函数	(87)
5.4 磁盘文件	(88)
5.4.1 文件的打开和关闭	(88)
5.4.2 文本文件的读/写	(89)
5.4.3 二进制文件的读/写	(90)
本章小结	(92)
习题五	(92)
第6章 面向对象程序设计	(95)
6.1 面向对象的基本概念	(95)
6.1.1 结构化程序设计	(96)
6.1.2 面向对象程序设计	(97)
6.1.3 可视化编程技术	(98)
6.2 面向对象程序设计的基本特点	(98)
6.2.1 抽象	(98)
6.2.2 封装	(99)
6.2.3 继承	(100)
6.2.4 多态	(101)
本章小结	(101)
习题六	(101)
第7章 类和对象	(103)
7.1 类	(103)
7.1.1 类的定义	(103)
7.1.2 类成员的访问控制	(104)
7.1.3 类的成员函数	(105)
7.1.4 对象	(108)
7.2 构造函数和析构函数	(109)
7.2.1 构造函数	(110)
7.2.2 析构函数	(112)
7.2.3 拷贝构造函数	(114)
7.3 静态成员	(120)
7.3.1 静态数据成员	(120)

7.3.2 静态成员函数	(122)
7.4 友元	(123)
本章小结	(126)
习题七	(127)
第8章 继承与多态	(129)
8.1 继承与派生	(129)
8.1.1 类的继承与派生	(129)
8.1.2 继承与派生的目的	(129)
8.1.3 派生类的声明	(129)
8.1.4 派生类的生成过程	(130)
8.1.5 继承方式	(132)
8.1.6 派生类的构造函数和析构函数	(137)
8.1.7 同名覆盖原则与二义性问题	(140)
8.1.8 虚基类及其派生类构造函数	(142)
8.1.9 赋值兼容规则	(146)
8.2 多态性	(147)
8.2.1 多态的类型	(147)
8.2.2 静态联编与动态联编	(147)
8.2.3 虚函数	(148)
8.2.4 纯虚函数	(149)
8.2.5 抽象类	(150)
本章小结	(151)
习题八	(152)
第9章 基于API的Windows编程	(154)
9.1 Windows编程的特点	(154)
9.1.1 事件驱动	(154)
9.1.2 应用程序和操作系统密不可分	(155)
9.1.3 图形界面	(156)
9.1.4 用户界面	(157)
9.2 Windows的管理机制	(157)
9.2.1 消息	(157)
9.2.2 消息的产生	(158)
9.2.3 消息的传递	(158)
9.2.4 消息的处理	(158)
9.2.5 Windows中的消息管理	(159)
9.3 Windows应用程序的主程序	(159)
9.3.1 主程序的格式	(159)

9.3.2	Windows 中常用的数据类型	(160)
9.3.3	窗口类的注册	(162)
9.3.4	创建窗口	(165)
9.3.5	窗口的显示	(166)
9.3.6	建立消息循环	(167)
9.3.7	终止应用程序	(168)
9.3.8	完整的主程序	(169)
9.4	Windows 应用程序的窗口函数	(171)
9.4.1	窗口函数	(171)
9.4.2	窗口函数的声明	(172)
9.4.3	窗口函数对消息的响应	(172)
9.4.4	消息映射表	(173)
	本章小结	(175)
	习题九	(175)
第 10 章	基于 MFC 的 Windows 编程	(176)
10.1	MFC 类库简介	(176)
10.2	MFC 类库的层次结构	(176)
10.2.1	窗口类	(177)
10.2.2	图形类	(177)
10.2.3	程序结构类	(178)
10.2.4	部分非 CObject 派生类	(178)
10.2.5	应用程序框架	(178)
10.3	文档/视图结构	(179)
10.3.1	文档/视图概念	(179)
10.3.2	视图类	(180)
10.3.3	文档类	(180)
10.3.4	文档/视图结构中的应用程序类	(181)
10.3.5	文档/视图结构中的框架窗口类	(182)
10.3.6	应用程序的运行机制	(183)
	本章小结	(184)
	习题十	(185)
第 11 章	Windows 编程实践	(186)
11.1	单文档应用程序	(186)
11.1.1	创建单文档程序	(186)
11.1.2	类和文件	(187)
11.1.3	消息处理机制	(188)
11.2	输入/输出	(188)

11.2.1	文本输出	(189)
11.2.2	键盘	(192)
11.2.3	鼠标	(195)
11.3	菜单	(198)
11.3.1	菜单简介	(198)
11.3.2	菜单的建立和实现	(198)
11.4	综合实例一(模拟时钟应用程序设计)	(202)
11.4.1	编程要求	(202)
11.4.2	问题分析	(202)
11.4.3	实现步骤	(203)
11.4.4	程序清单	(204)
11.5	综合实例二(数据库应用程序设计)	(213)
11.5.1	使用 Visual C++ 6.0 开发数据库的优势	(213)
11.5.2	常用的数据库系统开发技术	(214)
11.5.3	基于 ODBC 的数据库编程	(215)
本章小结		(229)
习题十一		(230)
第 12 章	实 验	(231)
12.1	Visual C++ 实验环境介绍	(231)
12.2	基础实验	(236)
	实验一 熟悉实验环境	(236)
	实验二 简单程序设计	(241)
	实验三 函数与结构化程序设计	(244)
	实验四 复杂数据类型	(246)
	实验五 类与对象	(250)
	实验六 继承与派生	(254)
	实验七 多态性与虚函数	(257)
	实验八 函数模板与类模板	(260)
	实验九 基于对话框的应用程序	(262)
	实验十 文档视图结构应用程序	(264)
12.3	综合实验	(272)
	综合实验一:贪吃蛇程序	(272)
	综合实验二:俄罗斯方块	(279)
参考文献		(295)



C++语言基础

C++是从C语言发展演变而来的,是在C的基础上引入了类和对象的机制后形成的。最初的C++被称为“带类的C”,1983年正式取名为C++。C++语言的标准化工作从1989年开始,于1994年制定了ANSI C++标准草案。以后又经过不断完善,成为目前的C++。C++全面兼容C,同时支持面向对象的程序设计方法。

1.1 C++基础知识

1.1.1 数据类型

数据类型是对数据进行划分的基本方法,也是数据结构的一种体现。一种语言提供的数据类型的丰富程度在很大程度上决定了这种语言所提供的数据结构的丰富程度。C++提供了十分丰富的数据类型,除基本类型外,还提供了导出(构造)数据类型,如数组、结构体、共用体、枚举等以满足不同应用的需要,另外还提供了指针和引用类型等。

C++是强制类型语言,它的数据必须先声明其数据类型才可使用。当数据被定义了数据类型以后,它的存储空间的大小和数值范围即已经确定。表1.1给出了C++中所有的基本数据类型,它是根据ANSI标准给定的类型、字宽和范围,其中字宽和范围是针对32位计算机的。有关构造数据类型的描述及应用分别在以后各章节中详细讲述。

表 1.1 C++中的基本数据类型

类 型	名 称	占字节数	取值范围
char	字符型	1	-128~127
[signed] char	有符号字符型	1	-128~127
unsigned char	无符号字符型	1	0~255
short [int]	短整型	2	-32768~32767
signed short [int]	有符号短整型	2	-32768~32767
unsigned short [int]	无符号短整型	2	0~65535
int	整型	4	$-2^{31} \sim (2^{31}-1)$
signed [int]	有符号整型	4	$-2^{31} \sim (2^{31}-1)$
unsigned [int]	无符号整型	4	$0 \sim (2^{31}-1)$
long [int]	长整型	4	$-2^{31} \sim (2^{31}-1)$

续表

类 型	名 称	占字节数	取值范围
signed long [int]	有符号长整型	4	$-2^{31} \sim (2^{31}-1)$
unsigned long [int]	无符号长整型	4	$0 \sim (2^{31}-1)$
float	实型	4	约 $-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$
double	双精度型	8	约 $-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$
long double	长双精度型	8	约 $-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$
void	无值型	0	

1.1.2 C++ 的关键字

关键字是系统中已经预定义的英文单词,也称为保留字。关键字具有特定的含义,系统已为它们定义了专门的用途,在程序中不能另作他用。

在C++中定义了48个关键字,如表1.2所示。

表 1.2 C++ 中的关键字

asm*	auto	break	case	catch*	char	chass*	const
continue	default	delete*	do	double	else	enum	extern
float	for	friend*	goto	if	inline*	int	long
new	operator	private*	protected*	public*	register	return	short
signed	sizeof	static	struct	switch	template*	this*	throw*
try	typedef	union	unsigned	virtual*	void	volatile	while

注意:标有“*”的关键字为C语言中所没有的;另外有5个关键字只能在C++中使用,不能用于Visual C++,它们是:asm,catch,template,throw以及try。

1.1.3 标识符、标点符号

标识符是由大小写字母、数字字符和下划线组成,且以字母和下划线开始的字符序列。标识符可用作常量名、变量名、函数名、自定义的数据类型名等。

C++中的标点符号共有9个: #、(、)、,、...、:、;、{和}。每一个符号都有其特定的用途,但它们不表示任何操作,只有与其他语言成分结合使用时才能显示出它们的作用。标点符号的具体使用场合和使用方法将在以后的有关章节作介绍。

分隔符的作用是将程序中的词法符号分隔开来,在C++中的分隔符有:运算符、空格、标点符号、回车键和Tab键。

1.1.4 常量、变量

常量是指具有固定的值、在程序运行期间值不发生变化的量。常量分为两类:一类是数值常量;另一类是标识符常量。数值常量就是常数,而标识符常量就是具有固定不变值的名字。

1. 数值常量

根据数值常量类型的不同,可分为整型常量、实型常量、字符型常量和字符串型常量。

1) 整型常量

整型常量是用整数来表示的常量。它可以有符号,但没有小数,包括正整数、负整数和零。整型常量可以用十进制数、八进制数、十六进制数三种方法表示。

整型常量中的长整型用L(或l)作后缀表示。例如,1234L,4321L。123L或123从值的大小上来看没有区别,但在内存空间中的存储格式却不一样。

类似地,整型常量中的无符号型用U(或u)作后缀表示。例如,1234u,4321u。

如果一个常量的后缀为L(或l)U(或u)或者是U(或u)L(或l),则都表示为unsigned long型的常量。例如,1234UL,1234UL,1234Lu。

注意:当没有明确指定为长整型或无符号整型常量时,编译时可根据常数的大小,由编译系统自动进行识别。

2) 实型常量

实型常量也称为浮点数,是用实数来表示的常量。它由整数部分和小数部分组成,并且只能用十进制表示。

实型常量有两种表示形式:

(1) 小数形式。它由整数部分和小数部分组成。整数部分和小数部分可以不同时出现,但至少要有个部分,且小数点不能省。如0.23,45.64,.89,-73.25,23.,.47。

(2) 指数形式。也称科学表示法,它常用来表示绝对值很大或很小的数。其表示格式为:

$\langle \text{小数} \rangle E(\text{或} e) \langle \text{指数} \rangle$

它表示 $\langle \text{小数} \rangle \times 10^{\langle \text{指数} \rangle}$,其中“小数”部分不能省,“指数”部分可正可负,但必须是整数。例如,1.23E-3,5.0e3等为合法的实型常量。1.23E-3和1.23e-3都表示 1.23×10^{-3} ;而e10,.E8,1.5e2.34等都是不合法的实型常量。

实型常量的后缀F(或f)表示单精度型,如1.234E-3F,236.43f;而后缀L(或l)表示双精度型常量。例如:5.0e3L,343.24l。

3) 字符常量

字符常量是用一对单引号(')括起来的一个字符,如:'a','B','?',' #'等都是合法的字符常量。在这里,单引号只是字符常量的定界符,不是字符常量的一部分。

上述字符常量是用该字符的图形符号表示的。但有些字符(比如'\','"'等)以及一些控制字符是没办法直接用图形符号表示的。为此,C++提供了另一种表示字符型常量的方法,即转义序列法。其具体方法如下:

以反斜杠(\)开头,后面跟一个字母代表一个控制字符,如'\a'代表响铃等。

由于'\','"'、'"'这3个字符有特殊的用法,所以当它们作为字符型常量出现时要采用转义语义序列,即'\\'表示字符'\','\''表示单引号','\"'表示双引号'\"'。

一些常用的转义序列字符如表1.3所示。

表1.3 C++中常用转义序列字符

转义字符	名 称	功能或用途
\a	响铃	用于输出
\n	换行符	用于输出
\r	回车符	用于输出
\f	换页符	用于输出

续表

转义字符	名 称	功能或用途
\t	水平制表符(tab 键)	用于输出
\v	纵向制表符	用于输出
\b	退格符(backspace 键)	用于退回一个字符
\\	反斜杠	用于输出或文件的路径中
\'	单引号	用于需要单引号的地方
\"	双引号	用于需要双引号的地方
\0	空字符	用于输出

4) 字符串常量

字符串常量是用一对双引号(")括起来若干个字符的序列。例如:

"This is a string." "abcdefg" "1223" "<? # \$"

字符串以双引号为定界符,但双引号本身不属于字符串。如果要在字符串中插入双引号、单引号或"\",都应借助转义符"\",比如:"\Hello!" ", "This\'s book.", "c:\\my document\\data.txt"等。

字符常量与字符串常量有如下几点区别:

(1) 一个字符常量占1个字节的存储空间,而字符串常量则可占用多个字节的存储空间(字符的总数加1,即字符串常量最后都要有一个'\0'作为该字符串的结束符)。例如,'a'仅占1个字节,用来存放字符a的ASCII码值(0x61);而"a"却要占2个字节,即除了1个字节存放字符'a'本身的ASICC码值外,还得有1个字节存放字符串常量的结束符'\0',因此"a"的存放值为0x6100。

(2) 字符常量和字符串常量所具有的功能不一样,如字符常量有加法和减法运算,而字符串常量不具有这些运算。例如:

'a'+ 'b'+1 是合法的。

"a"+ "b"+1 是不合法的。

(3) 字符串常量可通过相关的函数进行连接、拷贝等操作,但字符常量却不能。

2. 标识符常量

标识符常量是用标识符来表示的常量。定义标识符常量后,可以像数值常量一样使用该标识符。C++中可用两种方法定义标识符常量:一种是使用编译预处理命令的方法;另一种是利用常量说明语句,即用常量说明符const定义常量。比如:

```
#define PI 3.14159           // 用编译预处理指令定义标识符常量PI
#define STR "Hello,World!"
#define CHR 'a'
const float pi=3.14159;      // 用常量说明符const定义标识符常量pi
const char chr='a';
```

注意,用常量说明符const定义常量时,必须对它进行初始化。

通常情况下,用标识符常量比用字符型常量具有更大的灵活性和可读性。

例 1-1 已知半径为 2,求圆面积和周长。

其程序可如下:

```
#include <iostream.h>
#define PI 3.1415926           // A
const double r=2;              // B
void main ()
{
    double area, circumference;
    area=PI * r * r;
    circumference=2 * PI * r;
    cout<<"area="<<area<<endl; // 输出
    cout<<"circumference="<<circumference<<endl;
}
```

如果要增加其精度,只要改变A语句中PI的值。同样,当要求半径为2.5的圆面积和周长时,只要将B语句中的 $r=2$ 改为 $r=2.5$ 即可。

在C++编程中,常量定义一般都用const,而不用#define。

变量不具有固定的值,在程序运行期间其值可以改变。

变量具有3个基本要素:变量名、变量类型和变量值。当创建一个变量时,编译程序要为其分配一定数量的存储单元(若干个连续的字节),该存储单元用于存放变量的值。存储单元的大小(字节的个数)与该变量的类型相关,即编译系统根据表1.1中的各种数据类型的字节数给变量分配相应的存储单元,比如整型变量分配4个字节的存储单元,等等。

1) 变量说明

变量说明的一般格式为:

<存储类型><变量类型><变量名1><'<变量名2>'...<变量名n>>;

其中:

“存储类型”为可选部分,它表明变量存储空间的开辟类型。

“变量类型”为该变量所属的类型,它决定了编译系统为其分配存储空间的字节数以及该变量可进行的操作。其中变量的类型可选择上面所述的数据类型。

“变量名”可为任一合法的标识符,其命名规则与标识符的规则相同。在同一个语句中可同时说明多个类型相同的变量,它们之间要用逗号(,)分隔开。

注意:在进行变量说明(声明)时,必须以分号(;)结束。

在C++中,变量的说明是作为一条语句来处理的,称为变量说明语句。变量说明语句可以出现在程序中放置语句的任意位置。

在变量名所标识的存储单元中存放的数据称为变量的值。改变变量的值(对变量赋值)就是用新值取代其存储单元中的旧值;而使用变量的值(对变量的引用),就是从该存储单元中取出数据。不管什么类型的变量,都必须先定义、后引用。

2) 变量的赋值与初始化

只有给变量的存储单元指定一个确定的值,才可以引用该变量。首次给一个变量赋值称为变量的初始化,或称给变量赋初值。对变量赋初值的方法有以下3种:

(1) 在变量说明的同时对变量赋初值。例如: