

21世纪高等院校计算机教材系列

Visual C++

面向对象程序设计教程

● 刘瑞新 主编
● 曹建春 沈淑娟 张连堂 等编著



购书可获得增值回报
提供教学用电子教案



机械工业出版社
CHINA MACHINE PRESS



21 世纪高等院校计算机教材系列

Visual C++面向对象程序设计教程

刘瑞新 主编

曹建春 沈淑娟 张连堂 等编著



机械工业出版社

本书是面向对象程序设计的基础教程。书中首先全面介绍了面向对象的相关概念和 C++程序设计的相关知识,包括类、对象、继承、重载、多态、虚函数以及流等;然后介绍了 Visual C++集成开发环境、Windows 程序设计的相关概念、资源的创建和使用以及 API 程序设计方法,包括绘图、文本输出、鼠标和键盘、常用控件等内容;最后介绍了 MFC 及应用程序框架以及 MFC 程序设计的方法和思路,包括对话框、基本控件、文档、视图和数据存储以及图形、图像和文本的输出等内容,还介绍了有关数据库编程的内容。

本书适合学习面向对象程序设计的高等院校学生使用,也适用于自学 Visual C++的学生和工作人员。

图书在版编目(CIP)数据

Visual C++面向对象程序设计教程 / 刘瑞新主编. —北京:机械工业出版社, 2004.7

(21 世纪高等院校计算机教材系列)

ISBN 7-111-14910-6

I. V... II. 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(2004)第 069507 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 丁 诚

责任印制: 施 红

煤炭工业出版社印刷厂印刷·新华书店北京发行所发行

2004 年 8 月第 1 版·第 1 次印刷

787mm×1092mm $\frac{1}{16}$ · 19.25 印张·476 千字

0001—5000 册

定价: 27.00 元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010) 68993821、88379646

封面无防伪标均为盗版

出版说明

计算机技术是一门迅速发展的现代科学技术，它在经济建设与社会发展中，发挥着非常重要的作用。近年来，我国高等院校十分注重人才的培养，大力提倡素质教育、优化知识结构，提倡大学生必须掌握计算机应用技术。为了满足教育的需求，机械工业出版社组织了这套“21世纪高等院校计算机教材系列”。

在本套系列教材的组织编写过程中，我社聘请了各高等院校相关课程的主讲老师进行了充分的调研和细致的研讨，并针对非计算机专业的课程特点，根据自身的教学经验，总结出知识点、重点和难点，一并纳入到教材中。

本套系列教材定位准确，注重理论教学和实践教学相结合，逻辑性强，层次分明，叙述准确而精炼，图文并茂，习题丰富，非常适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

参加编写本系列教材的院校包括：清华大学、西安交通大学、北方交通大学、北京邮电大学、北京化工大学、北京科技大学、山东大学、首都经贸大学等。

机械工业出版社

前 言

目前在高校中, 计算机类专业以及计算机类相关的专业(例如电子、通信、机电、自动化等专业)均开设有“面向对象程序设计”课程。因为 C++ 完全支持面向对象, 因此学习面向对象程序设计, C++ 是首选语言。而学习 C++, 最好的集成开发环境是 Visual C++。

现在市场上关于 C++ 和 Visual C++ 的教程很多。通常 C++ 的教程从 C 语言的语法开始讲解, 面向对象只占其中的一半左右, 大部分学校在授课时间内只能完成面向过程程序设计方法的讲授, 因而学生对面向对象程序设计的了解只是皮毛而已。而且在学习 C++ 时, 由于不了解一种具体的开发环境(Visual C++ 或 C++ Builder), 因此学生不能理解面向对象的精髓部分。一般情况下, 大部分的 Visual C++ 教程只讲解 MFC 编程方法, 而没有介绍 API 编程方法, 学习的人不了解 Visual C++ 内部运行方式, 因此学生要么只能写出简单的程序, 要么按照面向过程的编程方式编写程序。

本教材作者通过多年的“面向对象程序设计”的教学与 Visual C++ 程序开发经历, 积累了许多宝贵的经验。我们认为“面向对象程序设计”的教学需要将 C++ 和 Visual C++ 紧密结合在一起, 为此本教程便应运而生了。

本教程具有以下一些特点:

1. 本书将面向对象编程的思想与方法贯彻始终, 放弃了面向过程的编程方法。
2. 本教程将 C++ 和 Visual C++ 有机地结合在一起, 其中 C++ 和 Visual C++ 的讲述各占一半。
3. 实例丰富, 而且每一个实例都有代表性, 尤其是 MFC 程序设计中实例的讲解有利于面向对象相关概念的理解。
4. 本教程在讲解 Visual C++ 时分别介绍了 MFC 和 API 两种编程方法, 其中 API 方法有助于学习了解 Visual C++ 内部工作机制, MFC 编程方式有助于提高编程效率。

本教程共分 3 个部分 19 章, 较全面地介绍了 C++ 和 Visual C++:

第一部分共 7 章(第 1 章~第 7 章), 全面介绍了面向对象的相关概念和 C++ 程序设计的相关知识。其中首先讲解了 C++ 程序设计基础知识; 然后介绍了面向对象程序设计相关概念和机制(包括类、对象、继承、重载、多态、虚函数以及流等)在 C++ 中的实现。

第二部分共 7 章(第 8 章~第 14 章), 全面介绍了 Windows 程序设计的相关概念以及 API 程序设计方法。其中首先介绍了 Visual C++ 集成开发环境; 然后介绍了 Windows 程序设计的相关概念(包括窗口、对象与句柄、事件驱动、消息等)以及 Windows 应用程序的框架; 接着介绍了 Windows 下资源的创建和使用; 最后介绍了 API 方式下绘图、文本输出的方法, 鼠标和键盘的使用, 常用控件的创建和使用。

第三部分共 5 章(第 15 章~第 19 章), 全面介绍了 MFC 程序设计的方法和思路。其中首先介绍了 MFC 及应用程序框架; 然后介绍了对话框和基本控件的创建和使用, 文档、视图和数据存储以及图形、图像和文本的输出; 最后介绍了使用 MFC 进行数据库编程的方法。

本书由刘瑞新主编, 曹建春编写了第 1 章, 第 2 章, 张连堂编写了第 3 章~第 7 章, 李

晶、葛立峰编写了第 8 章，刘光民编写了第 9 章～第 11 章，翟丽娟、刘大明编写了第 12 章，李智、臧顺娟编写了第 13 章，董亚、马志伟编写了第 14 章，孟英、王峰编写了第 15 章，陈嘉编写了第 16 章，孙洪玲、马亚平编写了第 17 章，沈淑娟编写了第 18 章、第 19 章，全书由郝小会主审。鉴于作者水平有限，书中的错误在所难免，敬请广大读者指正。

本书适合学习面向对象程序设计的学生，也适合自学 Visual C++ 的学生和工作人员。

为了配合本书的教学，机械工业出版社为读者免费提供电子教案，可在机械工业出版社网站（www.cmpbook.com）上下载。

作 者

目 录

出版说明

前言

第 1 章 C++简单程序设计	1
1.1 C++语言概述	1
1.1.1 C++的发展历程及特点	1
1.1.2 简单的 C++程序结构	2
1.1.3 C++字符集	3
1.1.4 词法符号(简称符号)	3
1.1.5 基本数据类型	4
1.2 常量和变量	5
1.2.1 常量	5
1.2.2 变量	8
1.2.3 typedef	10
1.3 简单 I/O 操作	10
1.4 运算符、表达式与语句	14
1.4.1 运算符与表达式	14
1.4.2 算术运算符和算术表达式	15
1.4.3 关系运算符与关系表达式	16
1.4.4 逻辑运算符与逻辑表达式	17
1.4.5 位操作运算符	17
1.4.6 赋值运算符与赋值表达式	17
1.4.7 其他运算符	18
1.4.8 语句	18
1.5 习题	19
第 2 章 C++程序设计基础	21
2.1 基本结构与控制语句	21
2.1.1 基本结构	21
2.1.2 if 语句	22
2.1.3 switch 语句	23
2.1.4 while 语句	25
2.1.5 do~while 语句	25
2.1.6 for 语句	26
2.1.7 转向控制语句	27
2.2 函数	28
2.2.1 函数的定义	28

2.2.2	函数的调用	29
2.2.3	函数的值	30
2.2.4	函数的嵌套调用	30
2.3	数组	31
2.3.1	一维数组	31
2.3.2	二维数组	33
2.3.3	字符数组	35
2.4	指针	36
2.4.1	指针变量	36
2.4.2	指针数组和多级指针	38
2.5	引用	39
2.6	编译预处理	42
2.6.1	宏定义	42
2.6.2	文件包含	42
2.6.3	条件编译	43
2.7	习题	44
第3章	类和对象	46
3.1	面向对象程序设计	46
3.1.1	面向对象的由来	46
3.1.2	面向对象的基本概念	47
3.2	类的定义	48
3.2.1	结构类型的概念	48
3.2.2	类	50
3.3	对象	53
3.4	静态成员	62
3.5	友元	67
3.6	内联函数	70
3.7	习题	71
第4章	继承	74
4.1	派生类	74
4.1.1	继承的概念	74
4.1.2	派生类的定义	75
4.2	多重继承	80
4.3	习题	85
第5章	方法重载	87
5.1	函数重载	87
5.2	运算符重载	88
5.2.1	运算符重载的几个问题	89
5.2.2	运算符重载的形式	90

5.2.3 其他运算符的重载举例	94
5.3 习题	97
第6章 多态性和虚函数	98
6.1 多态性	98
6.2 虚函数	100
6.2.1 虚函数的说明方法	100
6.2.2 纯虚函数和抽象类	103
6.2.3 虚析构函数	105
6.3 习题	107
第7章 流	108
7.1 I/O 标准流类	108
7.1.1 输入输出流类层次	108
7.1.2 标准输入和输出	110
7.1.3 重载插入运算符	114
7.2 文件流类	115
7.3 字节流类	117
7.3.1 输入流	117
7.3.2 输出流	117
7.4 习题	118
第8章 Visual C++集成开发环境	120
8.1 Visual C++集成开发环境简介	120
8.2 Visual C++ 6.0 的菜单栏	120
8.2.1 File 菜单	121
8.2.2 Edit 菜单	122
8.2.3 View 菜单	123
8.2.4 Insert 菜单	125
8.2.5 Project 菜单	125
8.2.6 Builder 菜单	126
8.2.7 Tools 菜单	127
8.3 Visual C++ 6.0 的工具栏	128
8.3.1 工具栏的构成	128
8.3.2 工具栏的定制与修改	128
8.4 联机帮助	129
8.5 习题	129
第9章 Windows 编程基础	131
9.1 Windows 程序设计基本概念	131
9.1.1 Visual C++应用程序设计方法	131
9.1.2 Windows 程序设计基本概念	132
9.2 Windows 中的事件驱动	134

9.3	Windows 应用程序中常见消息	135
9.4	Windows 应用程序的基本框架	136
9.4.1	Windows 应用程序的组成	136
9.4.2	Windows 应用程序实例	137
9.4.3	Windows 应用程序结构分析	140
9.4.4	Windows 程序开发流程	144
9.5	习题	145
第 10 章	Windows 资源及资源编辑器的使用	146
10.1	资源及资源编辑器	146
10.1.1	资源概述	146
10.1.2	资源编辑器	147
10.1.3	资源的公共编辑方法	147
10.2	资源标识符	148
10.2.1	资源标识符基本概念及取值规则	148
10.2.2	资源标识符的编辑	149
10.3	菜单编辑器的使用	151
10.3.1	菜单及菜单项的创建	151
10.3.2	菜单及菜单项的移动和复制	152
10.3.3	定义菜单项加速键	152
10.3.4	创建右键菜单	153
10.4	对话框编辑器的使用	153
10.4.1	对话框的创建	153
10.4.2	对话框内控件的添加	154
10.4.3	对话框内控件的布局及尺寸控制	155
10.5	图形编辑器的使用	156
10.5.1	图形编辑器的整体概况和基本操作	156
10.5.2	图标或光标的创建	158
10.5.3	编辑工具栏	159
10.6	习题	160
第 11 章	使用基本资源	161
11.1	图标	161
11.1.1	图标资源的创建和加载	161
11.1.2	图标资源应用实例	162
11.2	菜单资源及其应用	164
11.2.1	菜单的创建	164
11.2.2	菜单的动态创建和显示	166
11.2.3	菜单的消息处理	168
11.2.4	创建加速键	168
11.2.5	综合举例	169

11.3	对话框	173
11.3.1	对话框概述	174
11.3.2	模态对话框的编程方法	174
11.3.3	非模态对话框的编程方法	175
11.3.4	通用对话框资源及其应用	176
11.3.5	对话框综合实例	177
11.4	习题	181
第 12 章	Windows 绘图及文本输出	182
12.1	Windows 绘图相关概念	182
12.1.1	设备描述表	182
12.1.2	图形刷新	184
12.1.3	映像模式	186
12.2	绘图工具和颜色的设置与应用	187
12.2.1	画笔的应用	187
12.2.2	画刷的应用	188
12.2.3	常用绘图函数	189
12.3	绘图综合实例	190
12.4	文本的输出方法与字体的设置	192
12.4.1	设置文本的设备环境	192
12.4.2	文本的输出	194
12.5	习题	195
第 13 章	键盘和鼠标	197
13.1	键盘	197
13.1.1	基本概念	197
13.1.2	键盘消息	198
13.1.3	键盘输入示例	199
13.2	鼠标	204
13.2.1	鼠标消息	204
13.2.2	鼠标消息的处理	204
13.2.3	捕捉鼠标	205
13.2.4	鼠标应用示例	205
13.3	习题	208
第 14 章	常用控件	209
14.1	控件概述	209
14.1.1	控件	209
14.1.2	控件的创建	210
14.1.3	控件的消息	211
14.2	常用控件操作函数	212
14.3	常用控件	213

14.3.1	按钮控件	213
14.3.2	滚动条控件	215
14.3.3	静态控件	218
14.3.4	列表框控件	219
14.3.5	编辑框控件	221
14.3.6	组合框控件	222
14.4	习题	224
第 15 章	MFC 及应用程序框架	225
15.1	MFC 概述	225
15.2	MFC 类库的组成和主要类的介绍	226
15.2.1	MFC 中的类的组织结构	226
15.2.2	宏和全局函数	227
15.3	应用程序框架	227
15.3.1	使用应用程序向导	228
15.3.2	查看类、资源和文件	230
15.3.3	类向导	232
15.3.4	工程文件简介	234
15.4	习题	235
第 16 章	对话框和基本控件	236
16.1	对话框	236
16.1.1	创建基于对话框的工程	236
16.1.2	在工程中调用对话框	244
16.2	常用控件	246
16.2.1	常用控件的控件通知消息	246
16.2.2	静态控件	246
16.2.3	按钮控件	247
16.2.4	编辑框控件	248
16.2.5	滚动条控件	250
16.2.6	列表框控件	251
16.2.7	组合框控件	252
16.3	习题	252
第 17 章	文档、视图和数据存储	253
17.1	文档/视图结构简介	253
17.1.1	文档	253
17.1.2	视图	254
17.1.3	文档模板	255
17.2	文档、视图实例	255
17.2.1	建立应用程序	255
17.2.2	程序结构剖析	256

17.2.3 添加代码	260
17.3 文件处理	263
17.3.1 对象序列化简述	263
17.3.2 对象序列化	264
17.4 习题	265
第 18 章 图形、图像和文本输出	266
18.1 三种图形输出类型	266
18.2 MFC 中与 GDI 有关的类	266
18.2.1 设备描述表类	267
18.2.2 图形对象类	267
18.3 屏幕映射方式	270
18.4 常见的绘图任务	272
18.4.1 输出文本	272
18.4.2 输出图形	274
18.4.3 输出图像	276
18.5 习题	278
第 19 章 数据库编程	279
19.1 数据库的相关概念	279
19.2 ODBC 数据库程序设计	280
19.2.1 ODBC 简介	280
19.2.2 MFC 的 ODBC 类简介	280
19.2.3 CDatabase 类	281
19.2.4 CRecordset 类	282
19.2.5 CRecordView 类	285
19.2.6 程序举例	285
19.3 DAO 数据库程序设计	290
19.3.1 什么是 DAO	290
19.3.2 程序举例	291
19.4 习题	294

第1章 C++简单程序设计

C++是目前最受欢迎的程序设计语言之一，它可以用来编写系统软件也可以用来编写应用软件，因为其完全支持面向对象程序设计。面向对象程序设计思想来源于人们的实践，它更符合人们的思维习惯，必将成为未来最重要的编程思想。目前 C++已经成为人们学习面向对象程序设计的教学语言，通过 C++的学习，可以快速掌握面向对象程序设计的思路。本章是 C++的基础，首先介绍 C++的发展和特点，然后举例说明简单 C++程序结构和 C++的词法和语法基础知识（包括词法符号、常量和变量的概念、基本数据类型、运算符、表达式和语句等），本章还将讲解 C++中的输入和输出，为用户和程序之间的交互奠定一定的基础。

1.1 C++语言概述

C++语言从 C 语言发展而来，产生于 1983 年，经过 20 年的发展已经逐步走向成熟，C++语言是目前最受欢迎的面向对象程序设计语言之一。

1.1.1 C++的发展历程及特点

C++是从 C 语言发展而来的，C 语言是上个世纪最受欢迎的程序设计语言。1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改，并为它起了一个有趣的名字“B 语言”。意思是将 CPL 语言煮干，提炼出它的精华。而且他用 B 语言写了第一个 UNIX 操作系统。而在 1973 年，美国贝尔实验室的 D.M.Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二字母作为这种语言的名字，这就是 C 语言。为了使 UNIX 操作系统推广，1977 年 D.M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年 Brian W.Kernighian 和 D.M.Ritchie 出版了名著《The C Programming Language》，从而使 C 语言成为当时世界上流行最广泛的高级程序设计语言。

C 语言具有许多优点，但 C 语言是一种面向过程的编程语言，因此与其他面向过程的编程语言一样，它已经不能满足运用面向对象方法开发软件的需要。1980 年由 AT&T 贝尔实验室的 Bjarne Stroustrup 博士在 C 语言基础上开发出 C++语言，C++支持面向对象的程序设计。

研制 C++语言的一个首要目标是“根除”C 语言中存在的问题，使 C++语言成为一个更好的 C 语言。C++语言的另一个重要目标就是支持面向对象的程序设计，因此在 C++语言中引入了类的机制。最初的 C++语言被称为“带类的 C 语言”，1983 年正式取名为 C++语言。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++语言。

C++语言的主要特点表现在两个方面：一是全面兼容 C 语言；二是支持面向对象的方法。

首先，C++语言的确是一个更好的 C 语言。它保持了 C 语言的简洁、高效和接近汇编语言等特点，对 C 语言的类型系统进行了改革和扩充，因此 C++语言比 C 语言更安全，C++语

言的编译系统能检查出更多的类型错误。

然而,也正是由于对 C 语言的兼容,使得 C++语言不是一个纯正的面向对象的语言。C++语言既支持面向过程的程序设计,又支持面向对象的程序设计。

C++语言最有意义的方面是支持面向对象的特征。虽然与 C 语言的兼容使得 C++语言具有双重特点,但它在概念上是和 C 语言完全不同的语言,我们应该注意按照面向对象的思维方式去用 C++语言编写程序。

如果已经有其他面向过程高级语言的编程经验,那么学习 C++语言时应该着重学习它的面向对象的特征,对于与 C 语言兼容的部分只要了解一下就可以了;如果是初学编程者,直接学习面向对象的程序设计方法是可以的,但是对于数据类型、算法的控制结构、函数等还是应该学习的。它们不仅是面向过程程序设计的基础,也是面向对象编程的基础。因为在面向对象编程中,对象是程序的基本单位,但对象的静态属性往往需要用某种类型的数据来表示,对象的动态属性要由成员函数来实现,而函数的实现归根到底还是算法的设计。

1.1.2 简单的 C++程序结构

C++的程序由编译预处理、程序主体和注释组成。下面是一个简单的 C++程序。

【例 1-1】 一个简单的 C++程序。

```
#include <iostream.h>
void main()
{
    for (int i=0; i<3; i++){           // 循环语句
        cout << "Hello! my friend!\n"; // 向屏幕输出 Hello! my friend!
    }
}
```

程序运行的结果是在屏幕上显示三行提示信息“Hello! my friend!”。

从上面的例子可以看出一个简单的 C++源程序包括三个部分:

1. 编译预处理

以符号“#”开头的行,称为编译预处理行。编译命令是在程序被正常编译之前执行的,故称为预处理命令。

常用的预处理命令有:

- ① 文件包含命令。
- ② 条件编译。
- ③ 宏定义。

关于编译预处理的内容请参阅 2.6 节。

2. 程序主体

一个简单的 C++程序主体是由多个函数组成的,其中程序中必须有一个而且只能有一个主函数。主函数的函数名是 main(),它是程序的入口,主函数可以调用其他函数,其他函数可以互相调用,但是其他函数不能调用主函数。

函数由一些语句组成,在 C++中语句如果没有错误,将被编译和运行。和其他语言不同,一条语句可以占用多行,多条语句可以安排在一行中,一条语句结束的标志是分号“;”,如:

```

for (int i=0; i<3; i++)
{
    cout << "Hello! my friend!\n";
}

```

就是一条循环语句，详细内容请参阅 2.1 节。

3. 注释

注释是为了提高程序的可读性。一般分为两种：序言注释和注解性注释。前者用于程序开头，说明程序或文件的名称、用途、编写时间、编写人等；注解性注释一般出现在程序中，用于注释难懂的地方或做一些标记以便阅读。

注释方法有两种：一种是“//”开头的一行文字，直到换行；另一种是兼容 C 语言的注释，即把注释放在一对符号“/*”与“*/”之间，可以占多行。

1.1.3 C++字符集

每种语言都有自己的一套符号，符号是组成程序的基本单位，它是由若干字符组成的具有一定意义的最小词法单元，如标识符、关键字、运算符、分隔符、常量、注释符等。这里组成符号的字符必须是这种语言字符集中的合法字符，在 C++中规定了一个自己的字符集。C++的字符由下列三种字符组成。

① 大小写英文字母 a~z A~Z

② 数字字符 0~9

③ 特殊符号，包括：

空格 ! # % ^ & * _ - + = ~ < > / \ |
 . , ; ? ' " () [] { }

1.1.4 词法符号（简称符号）

C++共有 6 种词法符号，分别为：

1. 关键字

关键字是系统预留的词法符号，具有专门的用途和含义。下面列举出 ANSI 标准的关键字及一些 Visual C++扩展的一些关键字。

auto	bool	break	case	catch	char	class
const	const_case	continue	default	delete	do	double
dynamic_cast	else	enum	explicit	extern	false	float
for	friend	goto	if	inline	int	long
mutable	namespace	new	operator	private	protected	public
register	reinterpret_cast	return	short	signed	sizeof	static
static_case	struct	switch	template	this	throw	true
try	typedef	typeid	typename	union	unsigned	using
virtual	void	volatile	while			

2. 标识符

标识符是程序员定义的词法符号，用它来命名程序中的一些实体。常见的有函数名、类

名、变量名、常量名、对象名、标号名、类型名等。C++语言规定标识符由大小写字母、数字符号和下划线组成，第一个字符必须是字母或下划线。

定义标识符有一定的要求：

- ① 标识符长度没有限制，但不同的编译系统有不同的要求，一般不超过 31 个字符。
- ② 第一个字符必须是字母或下划线。
- ③ 标识符中大小写是有区别的。XY, xy, xY, Xy 都是不同的标识符。
- ④ 标识符定义时应尽可能使用有意义的单词。
- ⑤ 标识符不能与关键字相同。
- ⑥ 标识符中间不能有空格。

3. 运算符

运算符实际上是系统预定义的函数名字，这些函数作用于被操作的对象，将获得一个结果值。详细内容请参阅 1.4 节。

4. 分隔符

分隔符又称标点符号，用来分隔单词和程序正文。C++语言常用分隔符有：

- ① 空格符：用来做单词之间的分隔。
- ② 逗号：变量说明时分隔多个变量。
- ③ 分号：作为语句结束时的标记。在 for 语句括号中三个表达式的分隔也用到分号。
- ④ 冒号：用作语句标号，在 switch 语句中也会使用到。
- ⑤ 大括号：用来构造程序块。

5. 常量

C++语言中，常量有数字常量、字符常量、字符串常量等。详细内容请参阅 1.2 节。

6. 注释符

前面已经讲过，有两种方法，一种是 C++语言新增的注释方法，即以“//”开头，占一行。另外就是原来 C 语言的注释方法，以一对“/*”和“*/”括起的注释信息。

1.1.5 基本数据类型

类型是对系统中的实体的一种抽象，它描述了某种实体的基础特性，包括值的表示、存储空间的大小以及对该值的操作。C++的数据类型包括基本数据类型和构造数据类型两类。构造数据类型又称复合数据类型，它是一种更高级的抽象。当变量被定义为某种类型时会受到系统对该类型的特别保护，确保其值不受非法操作。

C++语言的基本数据类型有如下四种：

- ① 整型，说明符为 int。
- ② 字符型，说明符为 char。
- ③ 浮点型（又称实型），说明符为 float（单精度），double（双精度）。
- ④ 空值型，说明符为 void，用于函数和指针。

为了满足各种情况的需要，除了 void 型外，其他三种类型前面还可以加上修饰符改变原来的含义。

- ① signed，表示有符号。
- ② unsigned，表示无符号。