

计 算 机 系 列 教 材



Visual C++面向对象程序设计

主编 彭玉华



WUHAN UNIVERSITY PRESS

武汉大学出版社

计 算 机 系 列 教 材

Visual C++面向对象程序设计

主 编 彭玉华

副主编 黄 薇 曾 辉

刘 艳 袁 琼

主 审 赵海廷



WUHAN UNIVERSITY PRESS

武汉大学出版社

图书在版编目(CIP)数据

Visual C++面向对象程序设计/彭玉华主编. —武汉:武汉大学出版社, 2011. 8

计算机系列教材

ISBN 978-7-307-09035-4

I. V… II. 彭… 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(2011)第153131号

责任编辑:林莉 责任校对:刘欣 版式设计:支笛

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:cbs22@whu.edu.cn 网址:www.wdp.com.cn)

印刷:通山金地印务有限公司

开本:787×1092 1/16 印张:25.75 字数:654千字

版次:2011年8月第1版 2011年8月第1次印刷

ISBN 978-7-307-09035-4/TP·406 定价:39.00元

版权所有,不得翻印;凡购买我社的图书,如有缺页、倒页、脱页等质量问题,请与当地图书销售部门联系调换。



前言

本教材作者通过多年 C++ 面向对象程序设计的教学和 Visual C++ 开发经历, 结合高等院校的教学模式, 先介绍 C 语言, 再介绍 C++ 和 Visual C++。由于专业总学时不断压缩, 专业课开设较多, 需对相近课程进行整合, 于是结合 C++ 和 Visual C++ 教学实践, 编写了这本 Visual C++ 教材。

本书继承了《Visual C++ 面向对象程序设计教程》一书的特点, 针对 C++ 中对 C 语言的扩展内容进行了详细介绍, 并介绍了 Visual C++ 6.0 的开发环境, 及对 C++ 语言进行调试的方法。在学习 C++ 语言基础上进行 Visual C++ 应用程序的开发。将 C++ 面向对象程序设计分为 C++ 语言基础、C++ 程序设计、多态性与虚函数 3 章进行系统介绍, 选例完整、突出重点。通过这个过程, 上机调试 C++ 程序, 不仅掌握 C++ 语言的知识, 同时熟悉 Visual C++ 6.0 的开发环境。从第 4 章到第 7 章介绍 Windows 图形界面资源(对话框、菜单、工具栏和状态栏、常用控件、图形和文本处理)的创建和使用的方法, 重点介绍 MFC 应用程序的编程。第 8 章介绍 Windows 文档与视图之间的关系, 更进一步加深对 Windows 应用程序的来龙去脉的理解。第 9 章介绍数据库编程及项目开发实例。一般在讲解内容后紧跟实例, 操作步骤清晰, 都能上机调试通过, 配合《Visual C++ 面向对象程序设计实验教程》的实验, 一步步引导读者上机操作和编程, 能更容易、更快捷地掌握 Visual C++ 编程技巧, 学会用 Visual C++ 开发信息管理系统和游戏。鉴于作者水平有限, 书中难免有些错误, 恳请广大读者批评、指正。

本书不仅适合于教学, 也适合于用 Visual C++ 6.0 编程和开发应用程序的开发人员学习和参考。

本书在编写、修改过程中, 作者参考了参考文献中列举的书籍及资料, 在此向这些书籍的作者表示诚挚的谢意!

在本书编写过程中得到了武汉大学出版社、武昌理工学院王化文教授以及有关领导和同仁的大力支持和帮助, 在此表示衷心的感谢!

作者

2011 年 6 月

目 录

第1章 C++基本知识	1
1.1 C++概述	1
1.1.1 C++的起源	1
1.1.2 C++程序的特点	1
1.1.3 C++程序与C程序的比较	2
1.1.4 C++程序的基本结构	3
1.2 C++的输入与输出	5
1.2.1 C语言中的printf和scanf的缺陷	5
1.2.2 标准输入/输出(I/O)流类	6
1.2.3 I/O格式控制符	8
1.2.4 输入输出操作的成员函数	10
1.2.5 文件流	13
1.2.6 顺序文件	13
1.2.7 随机文件	16
1.3 C++的函数	17
1.3.1 函数的定义	17
1.3.2 函数的调用	18
1.3.3 内联函数	22
1.3.4 带默认参数的函数	23
1.3.5 重载函数	26
1.4 指针和引用	27
1.4.1 指针变量	27
1.4.2 指针和数组	28
1.4.3 引用	29
1.5 C++新增运算符	31
1.5.1 C++语言中的动态内存管理运算符	31
1.5.2 作用域限定运算符	33
1.6 Visual C++6.0 开发环境	34
1.6.1 Visual C++6.0 集成开发环境简介	34
1.6.2 工具栏的显示或隐藏	35
1.6.3 创建一个控制台应用程序	37
习题	40

第2章 面向对象的C++程序设计	46
2.1 类和对象	47
2.1.1 类的定义	47
2.1.2 对象	50
2.1.3 构造函数	54
2.1.4 析构函数	58
2.1.5 对象数组	59
2.1.6 对象指针	62
2.2 类的共享	65
2.2.1 静态成员变量	66
2.2.2 静态成员函数	69
2.2.3 友元函数	70
2.2.4 友元类	72
2.3 类的继承和派生	73
2.3.1 单继承	74
2.3.2 多继承	85
2.3.3 派生类的构造函数与析构函数	87
习题	92
第3章 多态性与虚函数	100
3.1 类的多重继承	100
3.1.1 二义性问题	100
3.1.2 虚基类	102
3.2 类的多态性	104
3.2.1 虚函数	104
3.2.2 纯虚函数与抽象类	107
3.3 运算符的重载	108
3.3.1 运算符重载的形式	108
3.3.2 赋值运算符的重载	111
3.3.3 插入符和提取符运算符的重载	113
3.3.4 增1或减1运算符的重载	115
3.4 模板	117
3.4.1 函数模板	117
3.4.2 类模板	119
习题	121
第4章 对话框	129
4.1 从C++到Windows编程	129
4.1.1 简单的Windows应用程序	129
4.1.2 Windows编程特点	136



4.1.3 Windows 基本数据类型	137
4.2 MFC 应用程序	137
4.2.1 MFC 概述	137
4.2.2 简单的 MFC 应用程序	138
4.2.3 MFC 应用程序的分析	139
4.3 MFC 应用程序类型	140
4.3.1 MFC 应用程序框架类型	140
4.3.2 创建一个单文档应用程序	141
4.3.3 项目管理	143
4.4 消息和消息映射	146
4.4.1 消息分类	147
4.4.2 ClassWizard 映射消息	147
4.5 Visual C++ 常用操作	151
4.5.1 类的添加和删除	151
4.5.2 类成员函数和变量的添加	152
4.5.3 文件打开和成员定位	153
4.5.4 使用向导工具栏	153
4.6 对话框的使用	154
4.6.1 对话框的概述	155
4.6.2 创建基于对话框的应用程序	156
4.6.3 对话框编辑器的使用	157
4.6.4 添加并使用对话框	159
4.6.5 模式对话框和非模式对话框	168
4.6.6 消息对话框	170
4.6.7 通用对话框	172
习题	176
第 5 章 菜单、工具栏和状态栏设计	178
5.1 菜单设计	178
5.1.1 菜单资源编辑器	178
5.1.2 设置菜单项属性	178
5.1.3 菜单的命令消息	180
5.1.4 设置菜单项加速键	181
5.1.5 在程序中改变菜单的属性	182
5.1.6 动态添加菜单	184
5.1.7 动态删除菜单	187
5.2 工具栏	189
5.2.1 工具栏编辑器	189
5.2.2 工具栏和菜单项的关联	190
5.2.3 用户界面更新消息的处理	193

5.3 状态栏.....	193
5.3.1 状态栏的定义.....	193
5.3.2 状态栏的常用操作.....	194
5.4 菜单和工具栏综合实例.....	197
5.5 设计学生管理信息系统用户主界面.....	200
5.5.1 设计主界面中主菜单.....	200
5.5.2 设计主界面中背景.....	201
习题.....	202
第6章 常用控件.....	204
6.1 控件概述.....	204
6.1.1 控件.....	204
6.1.2 在对话框模板上创建控件.....	205
6.1.3 控件的属性.....	206
6.1.4 控件的数据交换和数据校验.....	206
6.2 静态控件和按钮.....	208
6.2.1 静态控件.....	208
6.2.2 按钮.....	208
6.3 编辑框和旋转按钮.....	213
6.3.1 编辑框.....	213
6.3.2 旋转按钮.....	217
6.4 列表框.....	225
6.5 组合框.....	234
6.6 滚动条.....	239
6.7 滑动条.....	245
6.8 进度条.....	249
6.9 日期时间控件、图像列表和标签控件.....	256
6.9.1 日期时间控件.....	256
6.9.2 图像列表.....	257
6.9.3 标签控件.....	258
6.10 文件系统控件.....	266
习题.....	269
第7章 图形和文本处理.....	271
7.1 图形设备接口 (GDI) 概述.....	271
7.1.1 图形设备接口.....	271
7.1.2 设备环境类.....	271
7.1.3 坐标映射.....	272
7.1.4 GDI 图形对象.....	274
7.2 简单图形绘制.....	280



7.2.1 画点、线	280
7.2.2 画矩形和多边形	281
7.2.3 画曲线	281
7.3 文本处理	284
7.3.1 创建字体	284
7.3.2 文本输出	288
7.3.3 文本对齐	291
7.3.4 文本输出颜色的设置	291
习题	295
第 8 章 文档视图	297
8.1 文档视图体系结构	297
8.1.1 文档类	297
8.1.2 视图类	303
8.1.3 文档视图之间的相互作用	305
8.2 文档和视图应用示例	306
习题	313
第 9 章 数据库应用及项目开发实例	314
9.1 数据库应用开发概述	314
9.1.1 ODBC 技术	314
9.1.2 DAO 技术	314
9.1.3 OLE DB 技术	315
9.1.4 ADO 技术	315
9.2 使用 ODBC 编程	315
9.2.1 创建数据库	315
9.2.2 创建 ODBC 数据源	317
9.2.3 编写一个简单的支持数据库的应用程序	318
9.2.4 实现数据库的显示功能	320
9.2.5 实现数据库的查询功能	322
9.2.6 实现数据库的编辑功能	323
9.3 数据库相关的 ActiveX 控件	325
9.3.1 ADO Data 控件和 DataGrid 控件	326
9.3.2 DataList 控件和 DataCombo 控件	328
9.4 使用 ADO 访问数据库	329
9.4.1 ADO 的内置对象	329
9.4.2 ADO 的对象指针	330
9.4.3 ADO 编程	330
9.5 学生信息管理系统项目开发实例	334
9.5.1 用户登录模块设计	335

9.5.2 使用 ADO 连接数据库	339
9.5.3 学生基本信息模块设计	341
9.5.4 履历信息模块设计	349
9.5.5 成绩管理模块设计	356
9.5.6 院系设置模块设计	366
9.5.7 课程设置模块设计	380
9.5.8 用户管理模块设计	390
习题	398
附录 A C/C++主要关键字及其用途	399
附录 B C/C++运算符的优先级别	401
参考文献	402



第1章 C++基本知识



C++是一种面向对象的程序设计语言，也是目前最受欢迎的程序设计语言之一。面向对象程序设计思想来源于客观世界，符合人们的思维习惯，是未来最重要的编程思想。目前 C++ 已经成为学习面向对象程序设计的基础语言，学习 C++，可以快速掌握面向对象程序设计的思想。本章首先简要概述 C++ 语言的起源、C++ 程序的特点、C++ 程序与 C 程序的比较及 C++ 程序的基本结构等基础知识，然后介绍 C++ 程序中的输入与输出流、C++ 程序中的函数、C++ 新增的运算符等基础知识。

1.1 C++概述

C++语言是一种新型的、面向对象的计算机程序设计语言，既适合用来编写系统程序，也适合用来编写应用程序软件。自 20 世纪 80 年代由 AT&T 贝尔实验室的 Bjarne Stroustrup 博士创建 C++ 以来，C++ 日益受到重视并得到广泛应用，已经成为程序设计最主要的语言之一。

1.1.1 C++的起源

C++语言是从 C 语言发展而来的。C 语言是一种面向过程的程序设计语言，实现了结构化和模块化，在处理小规模程序时，显得比较方便，但是在处理大规模、复杂的程序时，就显得明显不足。因为 C 语言对程序人员的要求比较高，要求程序设计人员必须全面、细致地设计程序的每一个步骤和细节，整体规划程序的各个环节，显得很烦琐，不适合设计大型软件。针对这个问题，20 世纪 80 年代提出了面向对象的程序设计方法，客观模仿事物被组合在一起的方式，将人们的习惯思维和表达方式运用在程序设计中，程序设计人员可以按照人们通常习惯的思维方式进行程序设计，从而设计出更可靠、更容易理解、可重用性更强的程序，C++就是在这种情况下产生的。自 1983 年 C++ 诞生以来，就受到了人们普遍重视，特别是在 1998 年推出了 C++ 国际标准版本以后，C++ 更是得到了飞速发展，成为当代程序设计的主流语言。

1.1.2 C++程序的特点

C++是 C 语言的扩充和延伸，C++既保留了 C 语言原有的许多优点，又扩充了 C 语言的许多功能，增加了面向对象的机制。C++拥有丰富的运算符和数据结构，采用结构化程序设计方法，具有良好的移植性和高效的机器代码，程序设计更加简单灵活、方便快捷。C++既支持传统的面向过程程序设计，又支持当前的面向对象程序设计（Object Oriented Programming, OOP），是一种新型的程序设计语言，完全兼容 C 语言，用 C 语言编写的程序可以不加修改地在 C++中使用。C++提出了将数据和数据操作方法封装在一起，作为一个整

体，称为“对象”，每个对象都属于一个类型，称为“类”，类中的数据用本类的方法进行处理，并通过继承、派生、重载和多态性等特征，实现了程序代码的重复应用和程序自动生成。

1.1.3 C++程序与 C 程序的比较

C++程序和 C 程序有许多相同之处，可以通过下面三个实例来进行比较。

【例 1.1】用 C 语言编程求一个边长为 6 的正方形的面积。

```
#include<stdio.h>
void main()
{
    int L,S;
    scanf("%d",&L);           //输入边长 L 的值
    S=L*L;
    printf("%d\n",S);         //输出面积 S 的值
}
```

运行程序时输入 6，结果为：

36

【例 1.2】用 C++语言编程求一个边长为 6 的正方形的面积。

```
#include <iostream.h>
void main()
{
    int L,S;
    cin>>L;                   //输入边长 L 的值
    S=L*L;
    cout<<S<<endl;           //输出面积 S
}
```

运行程序时输入 6，结果为：

36

这两个程序分别是用 C 和 C++来编写的，可以看出，程序在结构上基本是相同的：都是以 main 函数作为入口函数；以一对大括号“{}”把函数中的语句括起来；以分号作为语句结束标志。因为二者都是采用面向过程的方法。但是也有一定的区别，如表 1-1 所示。

表 1-1 C 语言与 C++语言的区别

语言	输入输出头文件	输出流 cout	输入流 cin
C++	iostream.h	“<<” 插入符	“>>” 提取符
C	stdio.h	printf()	scanf()

C++语言除了支持面向过程的程序设计方法外，还支持面向对象的程序设计方法。下面的实例采用的是面向对象的程序设计方法来实现这一功能。

【例 1.3】用 C++语言编程求一个边长为 6 的正方形面积的程序。



```

#include <iostream.h>
class Area                                //声明一个名为 Area 的类
{
public:                                   //声明公有访问权限的成员
    Area (int newLength);                //声明 Area 类的构造函数
    ~Area( );                            //声明 Area 类的析构函数
private:                                 //声明私有访问权限的成员
    int Length,Square;
};
Area::Area(int newLength)                //Area 类构造函数的具体实现
{
    Length =newLength;
}
Area::~~Area ( )                         //Area 类析构函数的具体实现
{
    Square = Length * Length;
    cout<< Square <<endl;
}
void main( )                             //主函数
{
    Area myArea(6);                      //声明一个 Area 类的对象 myArea,自动调用构造函数
}                                          //退出 main()函数时自动调用析构函数释放对象 myArea

```

运行结果为:

36

例 1.3 采用面向对象的程序设计方法,程序的功能与例 1.1、例 1.2 完全一样。首先声明了一个类名为 Area 的类,在类中又声明了公有的构造函数 Area()与析构函数~Area(),以及私有变量 Length, Square,在主函数中直接声明了 Area 类的对象 myArea 并赋予参数值 6。该对象会调用类中的公有构造函数,通过参数传递将值 6 传递给私有变量 Length,在主函数运行结束时释放对象 myArea,并调用类中的公有析构函数,在析构函数中计算私有变量 Square 的值并输出。通过这样的方法最后计算出正方形的面积。有关该程序中的公有构造函数与析构函数、私有变量、类和对象的具体概念及声明方式将在第 2 章具体讲述。读者可能觉得这种方法比前面的两种方法要麻烦,但在实际开发软件时,会逐渐感受到采用面向对象的方法给程序开发带来的好处。

通过上面的三个实例可以看出:用 C++语言编写的程序和 C 语言编写的程序在程序结构上基本上是相同的,例如程序由函数构成,并且都是从 main()函数开始执行,语句分为说明语句和执行语句,用分号“;”作为语句结束的标志,但是二者之间又不完全相同,分别具有各自的特点。

1.1.4 C++程序的基本结构

完整的 C++程序一般包含类、普通函数和主函数。其中,普通函数和主函数与 C 语言中

的类似，类是 C++ 新增加的一个概念，也正是因为有了它，才使得 C++ 可以进行面向对象的程序设计。在 C++ 类中的成员一般有函数成员和数据成员，将数据和数据操作方法封装在一起，采用信息屏蔽的原则，减少了成员与外界的联系，提高了数据的安全性，使程序设计更加方便、灵活。类中成员的访问属性通常包括公有成员、私有成员和保护成员，使类中的成员具有不同的访问权限，从而实现类的封装性、继承性和多态性。

通过下面一个实例来了解 C++ 程序的基本结构。

【例 1.4】 计算长方形的面积。

```
#include <iostream.h>

class RectArea                                //声明一个名为 RectArea 的类
{
public:                                        //公有类型函数成员
    void input(int a,int b)
    {
        Length=a;
        Width=b;
    }
    void output( )
    {
        int Square=Length*Width;
        cout<<Square<<endl;                //输出 Square 值并换行
    }
private:                                     //私有类型数据成员
    int Length,Width;
};                                           // “;” 为类声明的结束标志
void show( )                                //普通函数
{
    cout<<"The area is:"<<endl;            //输出一行提示文本
}
void main( )                                //主函数
{
    RectArea myRectArea;                    //声明 RectArea 类的对象 myRectArea
    int a,b;
    cin>>a>>b;                              //输入 a,b 的值
    show( );                                //调用普通函数 show( )
                                           //通过对象 myRectArea 来访问类中的成员函数

    myRectArea.input(a,b);
    myRectArea.output( );

}
```

例 1.4 程序运行时需要输入两个整型数，分别表示长方形的长与宽，以空格分开。假如输入 6 和 2，则运行结果为：



The area is:

12

事实上，C++程序通常是由编译预处理命令、函数、语句、对象、变量、类、输入、输出及注释等几个部分组成的。

(1) 编译预处理命令

在C++程序的开头的行，以“#”作为标记的命令项（该命令不以“;”作语句的结束标志），称为编译预处理命令。编译预处理命令是在程序被正常编译之前执行的，故称为预处理命令。C++提供的编译预处理命令主要有：宏定义命令，文件包含命令和条件编译命令三种。

(2) 函数

函数是构成C++程序的基本单位，一个C++程序由若干个函数构成。函数包含函数说明部分和函数体，函数体一般由说明语句和执行语句构成。C++的函数分为两类：由C++系统提供的标准库函数和用户根据需要自己定义的函数。另外，一个C++程序中必须有一个而且只能有一个主函数。主函数名是main()，它是程序的入口，主函数可以调用其他函数，其他函数可以相互调用，但是其他函数不能调用主函数。

(3) 语句

语句是构成C++程序的基本单元，用来描述程序的执行过程，以分号“;”作为语句结束标志。一条语句可以占用多行，多条语句可以安排在一行中。

(4) 对象和变量

大多数C++程序都有对象和变量，变量的类型很多，基本类型有整型、实型、字符型等；对象是一种“类”类型的变量。

(5) 类

类是C++中特殊的数据类型，类包含数据成员和成员函数，类典型的特征是将数据和数据操作方法封装在一起，作为一个整体使用。

(6) 输入与输出

C++不仅保留了C的输入、输出系统，还增加了标准输入、输出流。使用标准输入、输出流时，用#include将其头文件包含进来。

(7) 注释

C++允许在程序中使用注释来提高程序的可读性。注释方法有两种：

①以“//”开头的文字一直到行尾，称为行注释。

②兼容C语言的注释，即把注释内容放在一对符号“/*”与“*/”之间，可以占用多行，称为段落注释。

1.2 C++的输入与输出

1.2.1 C语言中的printf和scanf的缺陷

C语言程序一般采用scanf函数和printf函数来进行数据的输入和输出操作。例如在下面这段程序中，采用scanf函数实现从键盘输入两个十进制整数，采用printf函数将输入的两个十进制数输出到终端（显示器）。

【例 1.5】输入两个十进制整数并输出。

```
#include <stdio.h>
void main( )
{
    int a,b;
    scanf("%d%d",&a,&b);
    printf("%d,%d\n",a,b);
}
```

在 C 语言中要执行输入和输出操作必须指定输入或输出数据的格式。在例 1.5 的程序中, 分别输入和输出两个十进制整数, 都使用了“%d”格式控制。如果要输入、输出不同类型的数据必须指定不同的数据格式控制。这给编写程序带来了一定的难度。在 C++中, 这个问题得到了很好的解决。

1.2.2 标准输入/输出(I/O)流类

C++保留了 C 语言的输入、输出系统, 可以直接使用 C 语言输入、输出方式进行数据的输入、输出。另外, 在 C++的编译系统中, 有一个 I/O 流类库, 数据的输入与输出可以通过 I/O 流来处理。“流”是 C++的一个重要概念, 数据从一个对象到另一个对象的流动抽象为“流”。它负责在数据的生产者与数据的消费者之间建立联系并管理数据的流动。当数据从外部对象流向程序时称为“输入流”, 当数据从程序流向外部对象时称为“输出流”。使用 C++的 I/O 标准流进行数据的输入、输出操作时, 系统能自动完成数据类型的转换。表 1-2 列出了 C++的 I/O 标准输入、输出流类。表 1-2 中 I/O 标准输入、输出流类之间的关系如图 1-1 所示。

表 1-2 I/O 标准流

	类 名	说 明	包含文件
抽象流基类	ios	流基类	iostream.h
输入流类	istream	通用输入流类和其他输入流的基类	iostream.h
	ifstream	输入文件流类	fstream.h
	istream_withassign	cin 的输入流类	iostream.h
	istrstream	输入字符串流类	strstream.h
输出流类	ostream	通用输出流类和其他输出流的基类	iostream.h
	ofstream	输出文件流类	fstream.h
	ostream_withassign	cout、cerr 和 clog 的输出流类	iostream.h
	ostrstream	输出字符串流类	strstream.h
输入/输出流类	iostream	通用输入/输出流类和其他输入/输出流的基类	iostream.h
	fstream	输入/输出文件流类	fstream.h
	strstream	输入/输出字符串流类	strstream.h
	stdiostream	标准 I/O 文件的输入/输出类	swtdiostr.h
流缓冲区类	streambuf	抽象流缓冲区基类	iostream.h
	filebuf	磁盘文件的流缓冲区类	fstream.h
	strstreambuf	字符串的流缓冲区类	strstream.h
	stdiobuf	标准 I/O 文件的流缓冲区类	stdiostr.h
预先定义的流初始化类	istream_init	预先定义的流初始化类	iostream.h

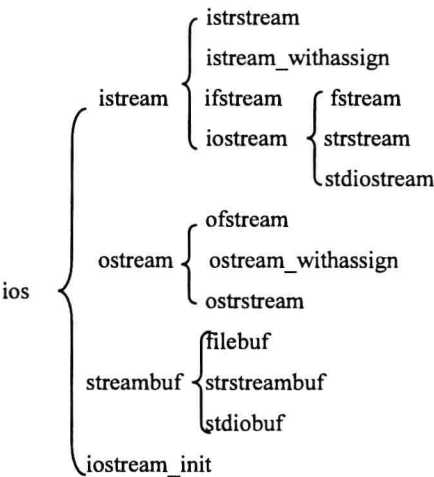


图 1-1 I/O 流关系图

为了方便程序设计，C++系统提供了 4 个预定义的标准 I/O 流对象：cin、cout、ceer 和 clog，如表 1-3 所示，使用这些对象时，必须在程序的开头用#include 将其头文件 iostream.h 包含进来。

表 1-3 C++系统预定义的标准流对象

流对象名	含 义	缺省设备
cin	标准输入	键 盘
cout	标准输出	显示器
ceer	标准错误输出	显示器
clog	ceer 的缓冲形式	显示器

1. 数据的输入

数据的输入可以使用输入流对象 cin，cin 代表标准输入设备(键盘)，使用提取符“>>”，表示从设备(键盘)获取数据送到输入流对象 cin 中，然后送到内存中的变量。使用 cin 可以获得多个从键盘的输入值，其具体使用格式如下：

```
cin>>表达式 1>>表达式 2...>>表达式 n;
```

其中，提取符后面的表达式通常是获取输入数据的变量或对象。例如：

```
int x,y;
double m,n
cin>>x>>y>>m>>n;
```

此时，要求从键盘输入两个整数和两个实数，输入时，数值之间用空格（空格数量不限）或回车隔开。执行上述语句，用户可以输入：

```
6 8 7.5 8.9↵
或
```