普通高等教育“十二五”规划教材

面向对象C++程序设计

李梅莲　鄢靖丰　主　编
王　爽　张一品　副主编

中国电力出版社
CHINA ELECTRIC POWER PRESS

普通高等教育“十二五”规划教材

面向对象C++程序设计

主　编　李梅莲　鄢靖丰
副主编　王　爽　张一品
编　写　高　静　石　琳
邹汪平　常玉华

中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书紧密结合当前教学改革和应用实践，立足于读者已学习 C 语言的基础之上，突出面向对象编程思想。首先介绍了从 C 语言快速过渡到 C++所涉及的基础知识，接着引导读者深入学习类与对象、运算符重载、继承与派生、多态性、C++输入/输出流。为了扩展读者面向对象的知识面，本书对异常、STL 泛型程序设计、MFC 编程基础也进行了阐述，最后通过综合实例阐明 C++程序设计的方法和技巧。

本书依据我国计算机程序设计教育的特点，重点放在让读者掌握分析问题和解决问题的方法上，力求将复杂的概念用简洁的语言描述出来，让读者学会用 C++语言编写实际应用程序。本书内容取舍得当、语言流畅、结构合理，融趣味性与科学性于一体。

本书可作为高等教院本、专科计算机相关专业面向对象程序设计语言课程的教材，也可供各类计算机编程人员学习参考。

图书在版编目（CIP）数据

面向对象 C++程序设计 / 李梅莲，鄢靖丰主编. —北京：中国电力出版社，2014.2

普通高等教育“十二五”规划教材

ISBN 978-7-5123-5429-6

Ⅰ. ①面… Ⅱ. ①李… ②鄢… 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（2013）第 317429 号

中国电力出版社出版、发行

（北京市东城区北京站西街 19 号 100005 http://www.cepp.sgcc.com.cn）

汇鑫印务有限公司印刷

各地新华书店经售

*

2014 年 2 月第一版 2014 年 2 月北京第一次印刷

787 毫米×1092 毫米 16 开本 18.25 印张 439 千字

定价 **32.50** 元

丛书编委会成员

丛书编写院校名单

北京农业职业学院
北京印刷学院
北京信息职业技术学院
北京工业职业技术学院
北京电子科技职业学院
江苏食品药品职业技术学院
江苏经贸职业技术学院
江苏农牧科技职业学院
常州机电职业技术学院
泰州师范高等专科学校
扬州市职业大学
徐州工程学院
南通市广播电视大学
苏州市职业大学
义乌工商职业技术学院
浙江警官职业学院
南昌师范学院
萍乡高等专科学校
重庆文理学院
四川职业技术学院
四川工商职业技术学院
四川交通职业技术学院
成都职业技术学院
内江师范学院
攀枝花学院
武汉软件工程职业学院
山东服装职业学院
山东信息职业技术学院
淄博职业学院
辽宁建筑职业学院
辽宁理工职业学院
营口职业技术学院
许昌学院
郑州升达经贸管理学院
河南化工职业学院
黄河科技学院
河北建材职业技术学院
河北软件职业技术学院
廊坊职业技术学院
黄山学院
太原师范学院
肇庆工商职业技术学院
广东工程职业技术学院
佛山职业技术学院
广西经贸职业技术学院
新疆工程学院
珠海城市职业技术学院

前　言

面向对象软件开发方法是吸收了软件工程领域有益概念和有效的方法而发展起来的一种软件开发方法。它集抽象性、封装性、继承性和多态性于一体，可以帮助人们开发出模块化、数据抽象程度高的，体现信息隐蔽、可复用、易修改、易扩充等特性的程序。目前，高校计算机各专业基本都开设了面向对象程序设计课程，大多数依托 C++这门语言进行入门介绍。

本书依据 ANSI C++标准进行编写，引导读者从一开始就按标准 C++的规定编程。为了使本书更好地满足教学需要，作者结合多年的教学经验和后续课程的需求，在参考大量书籍的基础上，从内容的取舍、结构的安排、例题和习题的选取等方面下足了功夫。本书思路清晰，通俗易懂，重在实用，强调增强学生的实际编程能力，让学生学得会、用得上。

在教材编写过程中，作者对 C++的知识体系和核心内容进行了深入地探讨，综合考虑 C++的整体结构和 C++初学者的接受能力，以实用为宗旨，做到多讲常用的内容。读者通过本书可以很好地掌握 C++面向对象中类、继承、多态等核心内容，同时，也可以了解 STL 编程、MFC 编程知识，最后，读者还能够通过书中综合案例及第 10 章的案例掌握面向对象的编程方法和技巧。

书中选取的案例贴近生活，注重提升学生的学习兴趣，选取适当的习题加深书中概念的理解。书中除 MFC 编程基础章节中的基于对话框例题采取了较新的 Visual Studio 2010 开发环境外，其他章节所有例题源程序在 Visual C+ + 6.0（简称 VC++）环境下编译运行通过，便于读者学习。

参与编写的作者均是来自教学一线的教师，其中第 1 章、第 3 章、第 4 章由李梅莲编写，第 10 章由鄢靖丰编写，第 2 章由王爽编写，第 5 章由张一品编写，第 6～第 9 章分别由高静、石琳、邹汪平、常玉华编写。全书由李梅莲、鄢靖丰主编和统稿。许昌学院信息工程学院的领导对本书的编写从人力物力上给予了大力支持，在此表示感谢！

由于作者水平有限，书中难免出现疏漏或不足，敬请广大读者不吝赐教。

李梅莲

2013 年 12 月

目　录

第1章 从C到C++

C语言是面向过程的结构化和模块化的语言，适宜处理较小规模的程序。C++保留了C语言的优点，增加了面向对象的机制，同时在原来面向过程机制的基础上，对C语言的功能做了不少扩充，是增强版的C语言。本章将对C++的产生、特点及对C++语言在非面向对象的扩充做简要的概述。

学习目标

1. 快速从C语言过渡到C++。
2. 了解C++的产生、特点。
3. 掌握C++的结构特性，会在C++编译环境下编写简单的控制台应用程序。
4. 掌握C++在非面向对象方面对C语言的扩充。

1.1 C++的产生及特点

C++是由贝尔实验室的Bjarne Stroustrup博士及其同事于20世纪80年代初在C语言的基础上开发成功的，与C语言兼容，用C语言编写的程序基本上可以不加修改地用于C++。C++保留了C语言原有的所有优点，既可用于面向过程的结构化程序设计，又可用于面向对象的程序设计，是一种功能强大的混合型的程序设计语言。

C++对C语言的“增强”，表现在以下两个方面：

（1）在面向过程机制的基础上，对C语言的功能进行了扩充。

（2）增加了面向对象的机制。

面向对象程序设计的主要优点：

（1）可提高程序的重用性（reusage）。

（2）可控制程序的复杂性（complex）。

（3）可改善程序的可维护性（maintance）。

（4）能够更好地支持大型程序设计。

（5）增强了计算机处理信息的范围。

（6）能很好地适应新的硬件环境。

面向对象程序设计是针对开发较大规模的程序而提出来的，目的是提高软件开发的效率。不要把面向对象和面向过程对立起来，面向对象和面向过程不是矛盾的，而是各有用途、互为补充，后续章节学习中将会体验到。

学习C++，既要会利用C++进行面向过程的结构化程序设计，也要会利用C++进行面向对象的程序设计。本书是在学习过C语言程序设计的基础上学习C++在面向对象程序设计中的应用。

C++语言发展大概可以分为以下三个阶段：

（1）第一阶段。从20世纪80年代到1995年。这一阶段C++语言基本上是传统类型上的面向对象语言，并且凭借着接近C语言的效率，在工业界使用的开发语言中占据了相当大的份额。

（2）第二阶段。从1995年到2000年，这一阶段由于标准模板库（STL）和后来的Boost等程序库的出现，泛型程序设计在C++中占据了越来越多的比重。当然，由于Java、C#等语言的出现和硬件价格的大幅度下降，C++受到了一定的冲击。

（3）第三阶段。从2000年至今，由于以Loki、MPL等程序库为代表的产生式编程和模板元编程的出现，C++出现了发展历史上又一个新的高峰，这些新技术的出现以及和原有技术的融合，使C++已经成为当今主流程序设计语言中最复杂的一员。

C++语言特点：

（1）C++保持与C语言兼容。

（2）用C++编写的程序可读性更好，代码结构更合理，可直接在程序中映射问题空间的结构。

（3）生成代码的质量高，运行效率仅比汇编语言代码段低10%～20%。

（4）可移植性强，从开发时间、费用到形成软件的可重用性、可扩充性、可维护性和可靠性等方面有了很大的提高，使得大中型程序开发项目变得容易很多。

（5）支持面向对象机制，可方便地构造出模拟现实问题的实体和操作。

出于保证语言简洁和运行高效等方面的考虑，C++的很多特性都是以库（如STL）或其他形式提供的，没有直接添加到语言本身里。关于此类话题，Bjarne Stroustrup的《C++语言的设计和演化》（1994）里做了详尽的介绍。

C++引入面向对象的概念，使得开发人机交互类型的应用程序变得更简单、快捷。很多优秀的程序框架包括MFC、QT、wxWidgets就是使用的C++。

C++语言也有一些缺点：复杂的C++程序的正确性难以保证。也有人提出不支持多线程的原语等缺陷。不过有如此多的知名人士提出了如此多的缺陷，也说明了C++被广泛使用和成功地应用。

C++语言由于语义本身复杂及与Unix相抵触，在Unix/Linux领域受到很多著名人士的强烈批评与抵制。

C++语言代码性能：人们通常认为，使用Java或C#的开发成本比C++低。但是，如果充分分析C++和这些语言的差别，会发现这句话的成立是有条件的。这个条件就是软件规模小，复杂度低。如果不超过3万行有效代码（不包括生成器产生的代码），这句话基本上还能成立。否则，随着代码量和复杂度的增加，C++的优势将会越来越明显。造成这种差别的就是C++的软件工程性。

1.2 简单C++程序分析

为了让读者较快地从C语言过渡到C++，下面先介绍几个简单的C++程序，并做简单的分析。

【例1-1】 输出一行字符："This is my first C++ program."。程序如下：

```
#include <iostream>                                    //包含头文件 iostream
using namespace std;                                   //使用命名空间 std
int main()
{
   cout<<"This is my first  C++ program."<<endl;       //输出字符串信息
   return 0;
}
```

在运行时会在屏幕上输出以下一行信息：

```
This is my first  C++ program.
```

如同 C 语言，C++用 main 函数代表主函数的名字。每个 C++程序都必须有且仅有一个 main 函数。main 函数前面的 int 的作用是声明函数的类型为整型。“return 0;”的作用是向操作系统返回一个零值。如果程序不能正常执行，则会自动向操作系统返回一个非零值，一般为-1，标准 C++要求 main 函数必须返回系统一个整数值。

函数体是由大括号{ }括起来的。本例中主函数内只有一个以 cout 开头的语句，是 C++用于输出的语句，cout 实际上是系统定义的标准输出流对象，采用插入运算符“<<”将数据输出到标准设备显示屏上并自动对输出的数据类型进行检查。“endl”是 C++输出时的控制符，作用是换行符，和“\n”作用相同。

与 C 语言一样，C++所有语句最后都应当有一个分号。

程序的第 1 行“#include <iostream>”，这是 C++的一个预处理命令，它以“#”开头以与 C++语句相区别，行的末尾没有分号。#include <iostream>的作用是将文件 iostream 的内容包含到该命令所在的程序文件中。文件 iostream 的作用是向程序提供输入或输出时所需要的一些信息，由于这类文件都放在程序单元的开头，所以称为“头文件”（head file）。在程序进行编译时，先对所有的预处理命令进行处理，将头文件的具体内容代替 #include 命令行，然后再对该程序单元进行整体编译。

程序的第 2 行“using namespace std;”的意思是“使用命名空间 std”。C++标准库中的类和函数是在命名空间 std 中声明的，因此程序中如果需要用到 C++标准库（此时就需要用 #include 命令行），就需要用“using namespace std;”作声明，表示要用到命名空间 std 中的内容。

在初学 C++时，对程序中的第 1，2 行只需理解：如果程序有输入或输出时，必须使用“#include <iostream>”命令以提供必要的信息，同时要用“using namespace std;”，使程序能够使用这些信息，否则程序编译时将出错，关于输入输出流及命名空间在第 6 章及第 7 章做详细介绍。

【例 1-2】 求圆的面积程序。可以写出以下程序：

```
// 求圆的面积程序(本行是注释行)
#include <iostream>                          //预处理命令
using namespace std;                         //使用命名空间 std
const double PI=3.14156;                     //定义常量 PI
int main()                                   //主函数首部
{                                            //函数体开始
   double r,s;                               //定义变量
   cout<<"输入圆的半径："<<endl;              //输入提示
```

```
    cin>>r;                                    //输入语句
    s=PI*r*r;                                  //赋值语句
    cout<<"s="<<s<<endl;                       //输出语句
    return 0;                                  //如程序正常结束,向操作系统返回一个零值
}                                              //函数结束
```

本程序的作用是求圆的面积。第1行“//求圆的面积程序”是一个注释行，C++规定在一行中如果出现“//”，则从它开始到本行末尾之间的全部内容都作为注释。程序体中的cin为系统定义的输入流对象，用于接收从键盘输入的数据，可自动做类型检查。

如果在运行时从键盘输入“5↙”，则输出为

```
"s=78.539"
```

【例1-3】 给出两个整数x和y，求两数中的大者。

```
#include <iostream>
using namespace std;
int max(int x,int y)                           //定义求两个数的最大值max函数
{  int z;
   if(x>y)
      z=x;
   else z=y;
   return(z);
}
int main()
{  int a,b,m;
   cout<<"输入两个整数: ";
   cin>>a>>b;
   m=max(a,b);                                 //调用max函数,将得到的值赋给m
   cout<<"max="<<m<<endl;
   return 0;
}
```

本程序包括两个函数：主函数main和被调用的函数max。

程序运行情况如下：

```
18  25 ↙(输入18和25给a和b)
max=25  (输出m的值)
```

注 意

输入的两个数据间用换行符或一个或多个空格间隔，不能以逗号或其他符号间隔。

在上面的程序中，max函数出现在main函数之前，因此在main函数中调用max函数时，编译系统能识别max是已定义的函数名。如果把两个函数的位置对换，即先写main函数，后写max函数，这时在编译main函数时遇到max，编译系统无法知道max代表何种含义，因而无法编译，按出错处理。

为了解决这个问题，在主函数中需要对被调用函数做声明。上面的程序可以改写为如下：

```
#include <iostream>
using namespace std;
int main()
{  int a,b,m;
   int max(int x,int y) ;                    //max 函数声明
   cout<<"输入两个整数: ";
   cin>>a>>b;
   m=max(a,b);                               //调用 max 函数,将得到的值赋给 m
   cout<<"max="<<m<<endl;
   return 0;
}
int max(int x,int y)                         //定义求两个数的最大值 max 函数
{  int z;
   if(x>y)
      z=x;
   else z=y;
   return(z);
}
```

只要在被调用函数首部的末尾加一个分号，就成为对该函数的函数声明。函数声明的位置应当在函数调用之前。

下面举一个包含类（class）和对象（object）的 C++程序，目的是使读者初步了解 C++是怎样体现面向对象程序设计方法的。

【例 1-4】　包含类的 C++程序。

```
#include <iostream>
using namespace std;
class Student                                //声明一个类,类名为 Student
{  private:                                  //以下为类中的私有部分
   int num;                                  //私有变量 num,表示学生学号
   int score;                                //私有变量 score,表示学生成绩
   public:                                   //以下为类中的公用部分
   void setData()                            //定义公用函数 setData
   {  cout<<"输入学号: ";
      cin>>num;                              //输入 num 的值
      cout<<"输入成绩: ";
      cin>>score;                            //输入 score 的值
   }
   void display()                            //定义公用函数 display
   {
   cout<<"学号: "<<num<<"  成绩: "<<score<<endl;  //输出学生信息
   }
};                                           //类的声明结束
Student stud1,stud2;                         //定义 stud1 和 stud2 为 Student 类的变量,称为对象
int main()                                   //主函数首部
{  stud1.setData();                          //调用对象 stud1 的公有 setData 函数
   stud2.setData();                          //调用对象 stud2 的公有 setData 函数
   stud1.display();                          //调用对象 stud1 的公有 display 函数
   stud2.display();                          //调用对象 stud2 的公有 display 函数
   return 0;
}
```

C++的类类似于 C 语言中的结构体，所不同的是类中可包含数据和函数两种成员，分别称为数据成员和成员函数。在 C++中把一组数据和有权调用这些数据的函数封装在一起，组成一种称为“类（class）”的数据结构。在例 1-4 程序中，数据成员 num、score 和成员函数 setData、display 组成了一个名为 Student 的类类型。成员函数是用来对数据成员进行操作的。也就是说，一个类是由一批数据以及对其操作的函数组成的。

类可以体现数据的封装性和信息隐蔽性。在例 1-4 程序中，在声明 Student 类时，把类中的数据和函数分为 private（私有的）和 public（公用的）两大类。把全部数据（num，score）指定为私有的，把全部函数（setData，display）指定为公用的。在大多数情况下，会把所有数据指定为私有，以实现信息隐蔽。

具有类类型特征的变量称为对象（object）。

程序运行情况如下：

```
输入学号：101
输入成绩：98
输入学号：102
输入成绩：90
学号：101  成绩：98
学号：102  成绩：90
```

C++程序的结构和书写格式归纳如下：

（1）一个 C++程序可以由一个程序单位或多个程序单位构成。每一个程序单位作为一个文件。在编译程序时，编译系统分别对各个文件进行编译，因此，一个文件是一个编译单元。

（2）在一个程序单位中，可以包括以下几个部分：

1）预处理命令。例 1-1～例 1-4 程序中都包括#include 命令。

2）全局声明部分（在函数外的声明部分）。在这部分中包括对用户自己定义的数据类型的声明和程序中所用到的变量的定义。

3）函数。函数是实现操作的部分，因此函数是程序中必须有的和最基本的组成部分。每一个程序必须包括一个或多个函数，其中必须有一个（而且只能有一个）主函数（main 函数）。

但是并不要求每一个程序文件都必须具有以上预处理命令、全局声明部分、函数三部分，可以缺少某些部分（包括函数）。

（3）一个函数由两部分组成：

1）函数首部，即函数的第一行，包括函数名、函数类型、函数属性、函数参数（形参）名、参数类型。

注：一个函数名后面必须跟一对圆括号，函数参数可以缺省，如 int main()。

2）函数体，即函数首部下面大括号内的部分。如果在一个函数中有多个大括号，则最外层的一对{ }为函数体的范围。

函数体一般包括：①局部声明部分 （在函数内的声明部分）。包括对本函数中所用到的类型、函数的声明和变量的定义。注：对数据的声明既可以放在函数之外（其作用范围是全局的），也可以放在函数内（其作用范围是局部的，只在本函数内有效）。②执行部分。由若干个执行语句组成，用来进行有关操作，以实现函数的功能。

（4）语句。包括两类：一类是声明语句，另一类是执行语句。C++对每一种语句赋予一种特定的功能。语句是实现操作的基本成分，显然，没有语句的函数是没有意义的。C++语

句必须以分号结束。

（5）一个 C++程序总是从 main 函数开始执行的，而不论 main 函数在整个程序中的位置如何。

（6）类（class）是 C++新增加的重要的数据类型，是 C++对 C 语言的最重要的发展。有了类，就可以实现面向对象程序设计方法中的封装、信息隐蔽、继承、派生、多态等功能。在一个类中可以包括数据成员和成员函数，它们可以被指定为私有的（private）和公用的（public）属性。私有的数据成员和成员函数只能被本类的成员函数所调用。

（7）C++程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行上。C++程序没有行号，也不像 FORTRAN 或 COBOL 那样严格规定书写格式（语句必须从某一列开始书写）。

1.3　C++对 C 在非面向对象的扩充

C++既可以用于面向过程的程序设计，也可以用于面向对象的程序设计。C++继承了 C 语言在面向过程领域的大部分功能和语法规定，并在此基础上做了大量扩充，主要在以下几方面增强了程序的灵活性和实用性。

1.3.1　注释

一个好的、有使用价值的源程序都应当加上必要的注释，以增加程序的可读性，有时，为了方便调试，暂时将某些源代码注释。C++还保留了 C 语言的注释形式，可以用“/*……*/”对 C++程序中的任何部分作块注释。在“/*”和“*/”之间的全部内容作为注释。

用“//”作注释时，有效范围只有一行，即本行有效，不能跨行。而用“/*……*/”作注释时有效范围为多行。只要在开始处有一个“/*”在最后一行结束处有一个“*/”即可。因此，一般习惯是：内容较少的简单注释常用“//”，内容较长的注释常用“/*……*/”。

以下两条语句是等价的：

```
x=y+z;  /* This is a comment */
x=y+z;  //This is a comment
```

1.3.2　C++数据输入输出

C++采用标准输入流对象 cin 从标准设备（默认键盘）上进行数据的输入，采用标准输出流对象 cout 把数据输出到标准设备（默认屏幕）上。下面仅给出 cin、cout 的基本用法，进一步地理解可参考本书第 6 章。

1. cout 用法

cout 经典用法：cout<<输出项 1<<输出项 2<<……<<输出项 n，输出项可以是任何可输出的类型，如字符串、常量、变量、有值函数、转义字符等，用 cout 输出数据可以是一项或多项。在连续输出多个数据时，应注意在数据之间加插入间隔符。如：

```
int x1=66;
float x2=34.1;
double x3=56.12;
cout<<x1<<x2<<x3<<600;
```

其中的 cout 语句将在屏幕上输出“6634.156.12600”，看不出什么数据，若改为

```
cout<<"x1="<<x1<<","<<"x2="<<x2<<","<<"x3="<<x3<<endl<<600<<endl;
```

则输出的结果为

```
x1=66,x2=34.1,x3=56.12
600
```

显然要清晰得多，有时可将上述语句分多行来写，让每一次用 cout 都有一个主题。

```
cout<<"x1="<<x1<<","<<"x2="<<x2<<","<<"x3="<<x3<<endl;
cout<<600<<endl;
```

2. cin 用法

cin 经典的用法：cin>>变量 1>>变量 2>>…>>变量 *n*，注意在>>后面只能出现变量名或数组元素，其中变量 1，…，变量 *n* 可以是内置数据类型如 int、char、float、double 等。在一条 cin 语句中同时为多个变量输入数据。输入数据的个数应当与 cin 语句中变量个数及数据类型相同，各输入数据之间用一个或多个空白（包括空格键、回车键、Tab 键）作为间隔符，全部数据输入完成后，按 Enter 键结束。如：

```
int x1;
double x2;
char x3;
cin>>x1>>x2>>x3;
```

应从键盘输入：“6 23.5b”，则 x1 为 6，x2 为 23.5，x3='b'，这时要注意 x2 和 x3 间的数据间隔不能用空格，否则 x3 的值就成了空格字符，这主要和 x3 为字符数据类型有关。

【例 1-5】 一个完整的 C++程序。

```
#include <iostream>
using namespace std;
int main()
{  char name[20];
   cout <<"Hello,your name: ";
   cin >>name;
   cout<<name<<"Nice to meet you!"<<endl;
    return 0;
 }
```

由上可知，C++的输入输出比 C 语言的输入输出简单易用。

1.3.3 灵活的变量说明

在一对{ }之间的块内的任何地方（除 while、do/while、for 和 if 结构的条件外）都可声明变量，其作用域为从定义位置起到块的末尾。看如下程序段：

```
float fun(int x,int y)
{   for (int i=0;i<10;i++)          // i 是 for 循环内的局部变量,循环外不起作用
      { int sum=0;                  // 循环体内也可对变量 sum 进行说明
        sum=sum+i;
        cout<<"sum="<<sum;
      }
      int z=0;                      // 使用变量 z 时才说明它
      z=x+y;
 }
```