普通高等教育“十三五”规划教材

C++程序设计案例教程

杨 平 编著

科学出版社

普通高等教育“十三五”规划教材

C++程序设计案例教程

杨 平 编著

科学出版社
北 京

内 容 简 介

本书是一本易学易用的C++面向对象程序设计教程，适合于已有C语言程序设计基础的读者。本书内容包括：C++语言概述、C++语言基础、类和对象、继承机制、虚函数与多态性、运算符重载、模板、输入和输出、标准模板库、C++综合应用实例。

本书以案例分析的方式讲解了C++的语言规则和编程方法，案例配有难度相同的“练一练”习题，学生可以边学边练，加强理解，提高兴趣；章后配有精心设计的课后习题，以便学生巩固本章知识点，提高程序设计的能力和综合运用知识的能力。

本书配有内容丰富的课程资源，课程网站上有教学视频、全部案例的源代码、演示文稿等；书中每个案例都配有视频讲解，读者可以通过扫描相关知识点的二维码，观看教学视频。

本书兼顾理论与实践，既可作为高等学校相关专业面向对象程序设计（C++语言）的教材，也可供编程爱好者自学使用。

图书在版编目(CIP)数据

C++程序设计案例教程 / 杨平编著. —北京：科学出版社，2016

（普通高等教育“十三五”规划教材）

ISBN 978-7-03-049422-1

Ⅰ. ①C… Ⅱ. ①杨… 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（2016）第167651号

责任编辑：冯 涛 王 惠 / 责任校对：陶丽荣

责任印制：吕春珉 / 封面设计：曹 来

科学出版社出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

双青印刷厂印刷

科学出版社发行 各地新华书店经销

*

2016年9月第 一 版 开本：787×1092 1/16

2016年9月第一次印刷 印张：14 1/2

字数：330 000

定价：34.00元

（如有印装质量问题，我社负责调换〈双青〉）

销售部电话 010-62136230 编辑部电话 010-62135397-2021

前　言

1. 本书的内容

本书是作者多年从事C++教学的经验总结，全面、系统地介绍了C++语言的基本概念和编程方法。为了便于教学实践，编者对各章节的内容和形式做了精心的设计。

全书共分为11章。第1章概述了面向对象程序设计的基本概念；第2章介绍了C++语言对C语言在非面向对象方面的扩充；第3～8章介绍了面向对象程序设计的基本方法，包括类和对象、继承机制、虚函数与多态性、运算符重载、模板、文件的输入和输出；第9章介绍了标准模板库的使用；第10章设计和实现了一个理财产品管理系统，将面向对象程序设计的知识和方法进行综合应用，有助于提高学生的实践技能；第11章设计和实现了一个基于图形界面飞镖打鸟的游戏，游戏案例的引入，增加了C++语言程序设计学习的趣味性。

本书在介绍知识点的同时，给出了有针对性的典型案例，并配有“练一练”习题。知识点部分是对该章节语法内容的介绍，精练到位；典型案例是针对该部分知识点精心设计的，既简洁又有代表性；“练一练”是针对知识点和案例配备的练习题，旨在逐步培养学生的学习兴趣和动手能力。

2. 本书的特色

本书是一本入门的面向对象程序设计（C++语言）教程。本书以案例为主线，语言叙述简明扼要，其主要特色有：

（1）内容精练，案例丰富、有趣味。

（2）每个案例都配有视频讲解，读者可以通过扫描的二维码观看教学视频，方便课前预习和课后复习。

（3）每个知识点都配有典型案例和“练一练”习题，习题内容源于作者多年的教学经验积累，难度适中，非常适合课堂练习和自学练习使用，便于学生巩固所学的知识。

（4）本书配有丰富的网络教学资源，课程网站上有教学视频、全部案例的源代码、演示文稿等（网址：http://sducc.sandau.edu.cn/）。

3. 致谢

感谢美国爱荷华大学倪军教授为本书做了细致的审稿和修订工作；感谢上海电子信息职业技术学院李小俊副教授为本书提出了很多宝贵建议；感谢张亮和张西蒙的全力支持，并协助完成了稿件的整理和校对；感谢金昱宏、陶培文、曾郑辉、项诗洋、徐璐伟、郑颖洁、徐倩倩、何乾等同学认真地试读和校对；感谢上海杉达学院信息学院全体教师

的大力支持。本书的出版由上海杉达学院“应用型试点专业人才培养方案建设（信息学院）”项目经费资助。感谢读者选用本书，欢迎读者对本书的内容提出意见和建议。

编者电子邮箱：brightyang@126.com。

编　者

2016 年 6 月

目　录

第1章

概　述

学习目标

- 了解C++的背景；
- 了解面向过程和面向对象两种程序设计方法的不同；
- 掌握面向对象程序设计的基本特征；
- 掌握Visual Studio环境下，C++应用程序的开发过程。

1.1 C++语言背景和应用领域

C语言产生于20世纪70年代，是一种面向过程的结构化程序设计语言。C++语言是从C语言发展演化而来的，为了强调它是C语言的增强版，就采用了C语言中的自加运算符“++”，1983年正式命名为C++。C++语言兼容了C语言的语法，增加了面向对象的概念。相比较而言，C++语言的语法更简单，功能更强大。

C++语言生成的目标代码质量高，编写的程序运行效率高，所以C++语言在现代软件领域中占据着举足轻重的地位。C++语言在许多应用领域有很大的优势，如游戏开发、网络软件开发、服务端开发、嵌入式开发、系统级开发、数字图像处理和虚拟现实仿真等。而且，随着信息化、智能化、网络化进程的加快，C++语言的应用也会越来越多，正在各个应用领域发挥重要作用。

1.2 C++语言程序

1.2.1 一个简单的C++程序

C++程序介绍和简单程序示例

【案例1-1】求两个数的和。

```
/*一个示范程序  1_1.cpp*/
```

```
#include<iostream>
using namespace std;
int main()
{
    int x,y;
    cout<<"输入两个整数: "<<endl;          //终端屏幕上输出（显示）字符串
    cin>>x>>y;                              //从键盘输入两个整数给 x 和 y
    int z;
    z = x + y;
    cout<<x<<"+"<<y<<"="<<z<<endl;  //终端屏幕上输出字符串和变量
    return 0;
}
```

程序的运行结果为：

```
输入两个整数:
3 4
3+4=7
```

程序的分析说明：

1. #include<iostream>

该头文件是 C++语言库中自带的输入/输出（简称 I/O）流文件，iostream 文件设置了 I/O 相关环境，定义了标准输入/输出流的对象 cin 与 cout 等。

2. using namespace std; 语句

C++语言经过较长时间的发展和标准化的过程，形成了两个版本，一个是最初开发者设计的传统的 C++，另一个是 ANSI/ISO（国际标准化委员会）创建的标准 C++，这两个版本的核心内容基本相同。表 1-1 为两个标准中的常用头文件对照说明，注意头文件名称的写法。

表 1-1 传统的 C++和 ANSI/ISO 标准 C++的头文件

传统的 C++	标准 C++	内 容
fstream.h	fstream	支持文件的流输入/输出
iomanip.h	iomanip	支持改变流的状态和格式
iostream.h	iostream	支持标准流和多字节标准流的输入和输出
math.h	cmath	各种数学函数
stdlib.h	cstdlib	许多杂的函数，包括内存分配、类型转换等
string.h	cstring	字符串处理函数
—	string	为字符串类型提供支持和定义
time.h	ctime	日期和时间函数库

所以，输入/输出的头文件可以采用#include<iostream> using namespace std;，也可以采用#include<iostream.h>。本书的源程序大多数采用 ANSI/ISO 标准 C++的库。

3. main()的返回值

一个C++程序必须有唯一的main()函数，以便操作系统调用。一般main()需要返给系统一个值，这个简单的C++程序中int main()指定返给系统一个整型数，main()函数的最后一句“return 0;”用于将0返回给操作系统。这通常被理解为程序在运行期间没有任何错误并且已经按照预先的方式完成，这是结束C++程序最常用的方法。关于函数的定义和使用，将在后面章节中详细介绍。

4. cin和cout

iostream文件中定义了两个流对象：istream cin;和ostream cout ;，其中，cin用于从键盘输入数据，cout用于将内存数据输出（显示）到屏幕上。简单格式如下：

```
cin>>变量名;
cout<<变量或者常量;
```

例如：

```
int x;
double y;
char z;
cin>>x>>y>>z;
cout<<"x="<<x<<"y="<<y;
```

关于cout的很多常用格式设置方法，将在第8章进行详细介绍；“cout<<”和“cin>>”带自定义类型变量的用法将在第6章介绍。

5. endl

endl在iostream文件中定义，也是std名字空间中的一个对象。向标准输出发送endl，类似于在控制台中按回车键，可以理解为换行输出。

【练一练1-1】编写C++风格的程序，在屏幕上输出“Hello World!”。

1.2.2 C++程序的构成

一个C++程序可以由一个文件或多个文件构成。在一个程序文件中，常常包括以下几个部分。

1. 预处理命令

在程序开头出现的以“#”开头的命令称为预处理命令。预处理命令分为3类：宏定义、文件包含和条件编译。#include称为调用文件包含的预处理命令，iostream是一个C++库文件，该文件中具体定义了标准输入/输出流的相关数据格式及其操作规范。例如，一个C++程序中需要用到输入流对象cin和输出流对象cout，可以用#include将其加入程序头部。

2. 输入和输出

一个程序常需要有输入和输出。

```
cout<<"输入两个整数: "<<endl;          //输出字符串
cin>>x>>y;                              //输入两个整数给 x 和 y
cout<<x<<"+"<<y<<"="<<z<<endl;          //输出字符串和变量
```

3. 函数

函数是用来管理程序的结构，特别是功能的模块结构。人们把一个特殊的功能编成一个函数，把多个常用或类型相近的函数存放在一起，组成一个函数库。这些特定的函数库通常被存放在一个特定文件中，以便多次灵活调用。一个 C++程序往往由若干个文件组成，每个文件又由若干个具有一定功能的函数组成。函数与函数之间既相对独立，又可以相互调用。一个 C++程序中有且仅有一个主函数 main()，执行程序时，系统必先执行主函数，并通过主函数来调用其他函数。有关函数的详细介绍可参考 2.5 节。

4. 变量

变量是程序运行时存储在内存中的数值，便于程序调用。使用变量之前一定要说明其类型。例如：

```
int a,b;      //a、b 为整型变量
char ch;      //ch 为字符型变量
```

在多数编译器中，C 语言中使用的变量必须定义在程序的首部；而 C++语言中使用的变量可以定义在程序的任意位置，只需在使用之前定义即可。因此在定义和使用变量方面，C++语言比 C 语言灵活得多。

5. 语句

函数是由若干条语句组成的。C++程序中的语句必须用分号结束。例如：

赋值语句：a=2-4;　c=a+b;

空语句：;

分支语句：if (3>2) a=3; else a=2;

循环语句：for(i=1; i<10; i++)　sum=sum+i;

6. 注释

注释是程序员为某一条语句或一段代码所做的必要说明，其目的是提高程序的可读性。注释一般有两种类型：序言性注释和解释性注释。序言性注释用在程序的开头，用于说明程序或文件的一些概要信息，包括程序或文件的名字、用途、编写时间、作者等；解释性注释用在程序中间，用于对某个变量、某条语句或某段代码进行解释。

注释有两种格式：“//”是单行注释；“/*……*/”是多行注释。一般来说，采用单行

注释比较清晰、规范，常用于序言性注释、变量和语句功能注释；而多行注释比较灵活，它不但可以用于段落注释、变量和语句功能注释，还可以用来调试程序。例如，把还没有确定的程序段给注释掉，便于逐步调试程序。

1.3 程序设计方法

在软件开发方法中，当前发展最成熟、应用最广泛的程序设计方法有两种：面向过程的程序设计方法和面向对象的程序设计方法。两者的关键区别在于程序设计的理念，前者侧重于程序按逻辑计算过程逐渐展开，将所有的数据以统一的数据结构进行处理；后者则侧重于根据数据属性和特性进行分析和处理，将所有的数据进行归类后，按其类别给予不同的处理。

下面简单介绍这两种程序设计方法。

1.3.1 面向过程的程序设计方法

面向过程的程序设计方法的主要原则可以概括为：自顶向下，逐步求精，模块化。

（1）自顶向下：程序设计时，先考虑主体（全局目标），后考虑细节（具体问题）。

（2）逐步求精：将复杂问题细分成若干个小问题，逐个求解。

（3）模块化：将程序要解决的总目标分解为若干个目标，再进一步分解为具体的小目标，每个目标称为一个模块。

在面向过程的程序设计方法中，程序可表示为：程序=数据结构+算法。这种程序设计方法，是将所求解问题中的数据定义为不同的数据结构，以功能为中心进行设计，通常用一个函数实现一个功能。通过分析确定解决问题所需要的步骤，然后通过函数将这些步骤实现，程序运行时依次调用函数。面向过程的程序设计方法存在一个函数与所处理的数据分离的问题，带来安全隐患。

随着问题规模与复杂性的增长，面向过程的结构化程序设计方法存在着以下明显的不足。

（1）数据安全性存在问题。

（2）可维护性及可重用性差。

（3）图形用户界面的应用程序很难用面向过程的程序设计方法来描述和实现，开发和维护也很困难。

1.3.2 面向对象的程序设计方法

在面向对象的程序设计方法中，对象是数据类型和算法的封装体。对象之间存在各种联系，但它们之间只能通过消息进行通信。程序可表示为：程序=对象+消息。

在面向对象的程序设计中，首先对所处理的数据进行类别的处理。这就需要对数据按其数据属性进行归类，理清类与类之间的关系，以及每个类所包含的数据特性和

相应的处理方法。因此，在面向对象的程序设计中着重于类的设计。由此可见，类是面向对象程序设计的基本模块，通过类的设计，以及类的对象创建来完成实体的建模任务。

面向对象的程序设计方法具有以下优点。

（1）程序模块间的关系更为简单，程序模块的独立性、数据的安全性方面比较完善。

（2）通过继承与多态性，可以大大提高程序的可重用性，使得软件的开发和维护都更为方便。

1.3.3 两种程序设计方法的比较

根据前面所述的两种程序设计过程的理念差异，对两种程序设计方法的比较如下。

（1）面向过程的程序设计方法是先确定算法，再确定数据类型；而面向对象的程序设计方法是先确定数据类型，再确定运算。

（2）在面向过程的程序设计中，习惯于建立数据类型存放数据并定义方法（函数）来操作数据；而在面向对象的程序设计中，则构造一个对象模型，将数据与方法组织在一起。

【案例 1-2】给定一个正方形边长，求其周长和面积。下面分别采用面向过程的程序设计方法和面向对象的程序设计方法进行思考。

1. 采用面向过程的程序设计方法进行思考

（1）确定正方形周长和面积的算法，也就是计算的公式。

（2）编写两个方法（函数）分别计算正方形的周长和面积。

（3）求周长的方法（函数）和求面积的方法（函数）都需要一个参数（正方形的边长），确定其数据类型。

```
/*采用面向过程程序设计方法求正方形的周长和面积  1_2_1.cpp*/
#include<iostream>
using namespace std;
int getPerimeter(int l)                          //计算周长的方法
{
    return 4*l;
}
int getArea(int l)                               //计算面积的方法
{
    return l*l;
}
int main()
{
    int a;                                       //正方形的边长
    cout<<"请输入正方形的边长"<<endl;
    cin>>a;
```

两种程序设计方法

```
    cout<<"周长"<<getPerimeter(a)<<endl; //通过调用周长的函数计算周长
    cout<<"面积"<<getArea(a)<<endl;      //通过调用面积的函数计算面积
    return 0;
}
```

程序的运行结果为：

```
请输入正方形的边长
3
周长 12
面积 9
```

2. 采用面向对象的程序设计方法进行思考

（1）一个具体的正方形实体可以看成正方形类型的一个对象。

（2）一个正方形对象有一个状态（边长）和两个计算功能（简称行为，求周长和求面积）所组成。

（3）将所有正方形的共性抽取出来，设计一个正方形类。

（4）通过正方形对象的行为，就可以求出某个具体的正方形对象的周长和面积。

```
/*采用面向对象程序设计方法求正方形的周长和面积 1_2_2.cpp*/
#include<iostream>
using namespace std;
class Square
{
private:
    int line;                    //正方形的边长
public:
    void input()                 //输入正方形边长的方法
    {
        cout<<"请输入正方形的边长"<<endl;
        cin>>line;
    }
    int getPerimeter()   //计算正方形周长的方法
    {
        return 4*line;
    }
    int getArea()                //计算正方形面积的方法
    {
        return line*line;
    }
    void show()                  //输出正方形信息的方法
    {
        cout<<"周长"<<getPerimeter()<<endl;
        cout<<"面积"<<getArea()<<endl;
    }
```