C++ 程序设计实用教程

苏仕华　编著

清华大学出版社

C++程序设计实用教程

苏仕华　编著

清华大学出版社

北　京

内 容 简 介

C++是一种高效实用的程序设计语言，它既可进行过程化程序设计，也可进行面向对象的程序设计，已成为程序设计人员最广泛的使用工具。

本书全面系统地讲述了C++语言的基本概念、基本语法和编程方法，较详尽地讲述了C++语言面向对象的重要特征：类和对象、继承和组合、多态性和虚函数等。本书列举了丰富多样的例题，每章后面都配有形式不同的练习题。

本书不要求读者学过C语言。由于面向过程程序设计部分的思想和方法也适合C语言，只是在实现上与C语言有些差异，因此通过该部分也可以学习C语言编程。不过，已学过C语言的读者还必须重新学习过程设计这部分内容，因为这部分介绍了面向对象和面向过程所共有的许多设计方法。

本书是作者总结多年教学实践经验编写而成的，书中文字通俗易懂，内容由浅入深，突出重点讲解，注重实际应用，易于教学。本书不仅可作为高等院校C++语言课程的教材，同时也可作为自学C++语言的读者以及广大工程技术人员学习C++语言的教材或参考用书。

图书在版编目（CIP）数据

C++程序设计实用教程/苏仕华编著. —北京：清华大学出版社，2006.8

ISBN 7-302-13244-5

Ⅰ.C…　Ⅱ.苏…　Ⅲ.C语言-程序设计-教材　Ⅳ.TP312

中国版本图书馆CIP数据核字（2006）第068080号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：曾　刚
文稿编辑：鲁秀敏
封面设计：范华明
版式设计：杨　洋
印 装 者：清华大学印刷厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：21　**字数**：479千字
版　　次：2006年8月第1版　2006年8月第1次印刷
书　　号：ISBN 7-302-13244-5/TP·8369
印　　数：1～5000
定　　价：29.00元

前　言

计算机科学发展的每一步几乎都在软件设计和程序设计语言中得到充分的体现。软件是一个发展的概念，随着软件开发规模的不断扩大和开发方式的变化，人们开始将程序设计语言作为一门科学来对待。程序设计方法和技术在各个时期的发展不仅直接导致了一大批风格各异的程序设计语言的诞生，而且对计算机理论、硬件、软件以及计算机应用技术等多方面都产生了深远的影响。

从软件专业的角度讲，能胜任较低层次的软件开发是软件开发能力的基础。因此，在开始学习编程的阶段，不应该把重点放在更高层次的技巧上，而应该放在直接采用这种语言手工编程的基本功上。

本书把重点放在程序设计方法上，将内容划分为 3 大部分：基本语言、面向过程的程序设计和面向对象的程序设计。本书不要求读者学过 C 语言，基本语言和面向过程设计部分与 C 语言是完全兼容的，只是一些实现上的差异而已。如果没有学过 C 语言，通过这部分的学习将同时能学会 C 语言编程。但对已经学过 C 语言的读者，也还要学习这部分内容，否则对后面的面向对象编程的学习就会受到影响。因为这部分并不是完全重复 C 语言的内容，而是揉入了 C++语言的特征。

全书共分 12 章。第 1 章是绪论。它是全书的开篇，将分别介绍基于过程、面向对象程序设计的基本概念，引入面向对象的概念并介绍 C++程序和面向对象编程的基础知识。第 2 章~第 7 章侧重于基本语言和面向过程编程的概念。第 2 章是数据类型和表达式，它是 C++语言程序设计的基础。第 3 章是结构化程序设计，介绍几种基本结构的程序设计方法。第 4 章是数组和字符串，是典型的构造类型，并简要说明它们的使用方法。第 5 章是指针和引用。C++中引入了引用的概念，它为 C++的程序设计提供了新的舞台。第 6 章是函数。在 C++面向对象程序设计中，成员函数也是函数，所以本章也是学习面向对象编程的基础。第 7 章是结构类型。它是 C 语言中一个非常有用的构造类型，为了使读者能够更好地理解和掌握类和对象的概念，特增加了这一章，而在许多 C++教材中是没有写这方面内容的。第 8 章~第 12 章是 C++面向对象的程序设计。第 8 章是类和对象基础。这一章首先抽象对象的概念以便更好地描述对象，然后讨论对象的 3 大要素并介绍面向对象语言的 4 大特征。重点介绍在 C++中定义类、建立和使用对象的方法。第 9 章是类和对象的应用。进一步对类和对象的其他方面进行介绍，这些内容包括对象的指针、引用和数组以及常类型、子对象和堆对象等内容，本章将通过大量的例子来进一步讲述类和对象的应用。第 10 章是继承和派生类。将讨论 C++语言继承方面的语法特征和一般的使用方法。第 11 章是多态性和虚函数。由于多态性是一个与实现有关的概念，因而难于理解和掌握。这一章将重点集中于介绍运行时的多态性，并通过图解和大量程序实例帮助读者更好地理解多态性。第 12 章是

介绍C++的I/O流类库及其操作和应用。

本书在每一章中都介绍了许多相应的例题，这些例题程序都是在Visual C++ 6.0环境下运行通过的。另外，在每一章都配置了一定数量的习题。通过这些例题介绍和习题的练习，对学生掌握C++程序设计的方法、技术和技巧都会有很大的帮助。

在本书的编写过程中，得到了中国科学技术大学自动化系刘振安教授的大力支持和帮助，在百忙之中审阅了书稿并提出了许多宝贵的意见，特别表示感谢。另外，在本书的写作中参考了大量的图书资料，在此对这些作者表示衷心的感谢。刘燕华、修大成、苏晓凌、黄学俊、刘燕君、余华敏等参加了本书的部分章节的编写、例题和习题程序的编写调试以及文字的编辑等工作。

尽管我们尽了最大的努力，但因作者才疏学浅，错误之处在所难免，敬请广大读者和同行给予批评指正。

苏仕华 sush@ustc.edu.cn

2006年2月

于中国科学技术大学

目　录

第1章 绪 论

计算机科学与信息技术的飞速发展、日新月异，已极大地改变了我们的生活方式，使我们生活在一个以光速传递的时代。计算机的每一步发展几乎都在软件设计和程序设计语言中得到充分体现。随着软件开发规模的扩大和开发方式的变化，人们开始将程序设计语言作为一门科学来对待，程序设计方法和技术在各个时期的发展直接导致了一大批风格各异的程序设计语言的诞生，而 C++语言正是在这种大环境下诞生的一个典型的面向对象的程序设计语言。

本章将分别介绍程序设计语言的发展、面向过程、面向对象程序设计的基本概念等几个方面的内容，着重讲述 C++语言编程特点，包括 C++语言与 C 语言的关系、C++语言的词法规则和 C++语言程序的实现等，特别是有关面向对象的概念，包括什么是面向对象、面向对象语言的特点、面向对象和面向过程的区别等。通过对面向对象概念的学习和理解，将会进一步了解 C++语言的特点及其用处。

1.1 程序设计语言的发展

所谓软件开发就是对给定问题求解的过程。软件开发者将通过人的思维对该问题域客观存在的事物，以及对所要解决的问题产生正确的认识和理解，包括弄清事物的属性、行为及其彼此之间的关系并找出解决问题的方法。所以开发人员对问题域的认识是人类的一种思维活动。尤其是在软件开发过程中，要求人们对问题域的理解，要比日常生活中对它的理解更深刻、更准确。

用一种语言把人们对问题域中事物的认识、对问题及其解决方法的认识描述出来，而最终的描述必须使用一种能够被计算机识别的语言，即编程语言。

最基本的计算机语言是机器语言，机器语言就是使用二进制位来表示程序指令。虽然绝大多数计算机完成的功能基本相近，不同计算机的设计者也都可能会采用不同的二进制代码集来表示程序指令。所以不同的计算机使用的计算机语言并不一定相同。但有一点是相同的，那就是现代计算机都是以二进制代码的形式来存储和处理数据的。

在早期的计算机应用中，实现的程序都是用机器语言编写的。为了实现应用程序的编写，程序员需要记住各种操作的机器语言指令。同时，为了存取数据，程序员还必须记住所有数据在内存中的存储地址。因此，这种需要记住大量的编码指令来编写程序的方法不仅难以实现，而且非常容易出错。直接使用机器语言来编写程序是一种相当复杂的手工劳动，它要求使用者熟悉计算机的有关细节，一般的工程技术人员难以掌握。

汇编语言的出现简化了编程人员的工作。汇编语言的主要特征是可以用助记符来表示

每一条机器指令。由于汇编语言比机器语言容易记忆，编程效率就比机器语言前进了一大步，但汇编语言程序的大部分语句还是和机器指令一一对应的，与机器的相关性仍然很强。

用汇编语言编好的程序还需要由相应的翻译程序，将其翻译成计算机可执行的机器语言程序。用程序设计语言写成的程序称为源程序，这种源程序可以在具有该种语言编译系统的不同计算机上使用。源程序必须翻译成机器语言才能执行，逐条翻译并执行的翻译程序称为解释程序，而将源程序一次翻译成目标程序然后再执行的翻译程序称为编译程序，例如 BASIC 语言就有解释程序和编译程序两种，而 FORTRAN、C 和 C++等都只有编译程序。

高级语言起始于 20 世纪 50 年代中期，因为它们更接近自然语言和数学语言，所以用它们编写的程序可读性强，交流起来也比较方便。最早问世的高级语言是 FORTRAN，60 年代初期，BASIC、COBOL、LISP 和 ALGOL 相继出现。70 年代以来，随着结构化程序设计思想的日益深入，使得这段时期问世的几种程序设计语言的控制结构大为简化，比较有代表性的有 PASCAL 和 C 语言等，它们均属于面向过程的程序设计语言。从 60 年代起，ALGOL、PL/1 与 SNOBOL4 逐渐消失。尽管 PASCAL 语言的许多结构用在 Ada 语言中，但在 70 年代早期它已度过了鼎盛时期，逐渐成为教学用语言。目前仍在使用的较老的语言都经历了阶段性的修改以便反映来自其他计算机领域的明显影响。例如 FORTRAN 90 和 Ada 95。LISP 更新为 Scherne LISP，以后又改为 Common LISP。这些语言又称为面向过程的语言。较新的语言如 Delphi 是从 PASCAL、C++是从 C 演化而来的混合型面向对象语言。

面向对象语言是比面向过程语言更高级的一种高级语言。面向对象语言的出现改变了编程者的思维方式，使设计程序的出发点由着眼于问题域中的过程转向着眼于问题域中的对象及其相互关系。这种转变更加符合人们对客观事物的认识，因此，面向对象语言更接近于自然语言，是人们对于客观事物更高层次的抽象。

1.2 计算机求解问题的过程

在搞清楚机器语言和高级语言的基本概念之后，如何按照某种高级语言规则来编写程序解决实际应用问题，在计算机上又如何运行用高级语言编写的程序等，是每个程序设计人员所必须了解和掌握的。

编写程序的过程就是解决问题的过程。不同的程序设计人员可能采用不同的方法来解决问题。如果采用了好的解决问题的方法，不仅可以解决实际问题，而且思路清晰易懂，可移植性和适应性强。因此，要想成为一个优秀的程序设计人员，必须学会解决问题的方法。常用解决问题的几个步骤为：问题需求分析、设计解决方案（算法）和编程解决问题。

开发设计一个程序解决实际问题，首先要从分析问题开始，然后设计解决问题的算法，再用高级语言编写程序语句（编码），最后在计算机中运行程序，得出结果。

问题分析是程序设计最重要的一步。问题分析就是要彻底弄清问题需求，找出需要的输入、输出和必要的数据处理等。如果问题比较复杂，可将其分解成若干个子问题，然后再对每个子问题进行需求分析。

在经过认真仔细的问题分析之后，下一步就是设计算法解决问题。如若需要解决的问题被分解为若干个子问题，那么每个子问题都需要分别设计算法。在前面已经说过，解决一个问题的算法可能有多种，但算法首先应是正确的，算法的优劣好坏与否也是需要分析验证的。

当算法分析设计并证明是正确的以后，就可以使用某种高级语言编写源程序代码，以实现解决问题的算法，源程序必须符合程序设计语言的规则。程序设计完成之后，可利用计算机系统中的文本编辑器输入编辑该源程序代码，然后通过系统提供的高级语言编译程序来编译检查源程序代码的语法和句法是否正确，若有错则必须修改错误，再进行编译检查，直至完全正确并生成目标程序为止。接着，由连接程序将目标程序和所用到的其他程序连接在一起生成可执行程序。最后就是执行程序，给出结果。综上所述，计算机解决问题的全过程可用图 1.1 来表示。

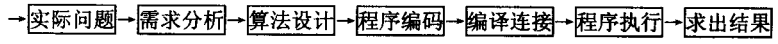

图 1.1　计算机解决问题的流程图

如前所述，通常在编写源程序代码之前，需要做大量的准备工作。经过认真分析和精心设计的源程序代码不仅有利于及早发现错误，而且程序有很强的适应性，可以增强程序的可读性和易修改性。所以，通过本书的学习，读者不仅可以掌握 C++程序设计的技术，还可以学到许多分析问题、解决问题的方法。在书中的每一章中都会有实际问题的设计用例，在这些用例中不仅介绍问题的程序设计，而且介绍分析问题、解决问题的方法和步骤，以帮助读者理解和掌握书中所涉及到的各种概念。

【例 1.1】已知一个三角形的 3 个顶点的坐标，求这个三角形的面积。

1. 问题分析

要想计算三角形的面积，就必须知道三角形的 3 条边长，而边长可以通过坐标求得。由坐标求边长和由边长求面积的计算公式如下：

假定 3 个坐标点分别为 $a(x1,y1)$、$b(x2,y2)$、$c(x3,y3)$，所以 3 条边长可用下面的公式求出：

$$ab=\sqrt{(x1-x2)^2+(y1-y2)^2}$$

$$ac=\sqrt{(x1-x3)^2+(y1-y3)^2}$$

$$bc=\sqrt{(x2-x3)^2+(y2-y3)^2}$$

求面积之前先计算边长和的一半：$s=\dfrac{ab+ac+bc}{2}$

求面积的计算公式为：$area=\sqrt{s(s-ab)(s-ac)(s-bc)}$

2. 算法设计

由上述分析，可得出求给定3个顶点坐标点的三角形的面积的算法描述如下：

（1）输入3个顶点的坐标。

（2）计算三角形的3条边的边长。

（3）计算3条边长和的一半值，求三角形的面积。

（4）输出三角形的面积。

3. 程序编码

根据上述问题分析和算法描述，很容易写出求解问题的C++源程序代码。由于求三角形的边长要计算3次，因此可编写一个计算边长的函数。因为在程序中要用到输入/输出流和数学函数，所以在程序的开始处必须要包含相关的以.h为扩展名的头文件。编写其源程序如下：

```
#include<iostream.h>     //包含输入/输出流
#include<math.h>         //包含数学函数的头文件
double edge(double x1,double x2,double y1,double y2)
{   //求三角形的边长
      double len;
      len=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));    //求边长
      return len;
}
void main( )
{
      double x1,x2,x3,y1,y2,y3,s,area,ab,ac,bc;     //说明变量
      cin>>x1>>x2>>x3;
      cin>>y1>>y2>>y3;                              //输入坐标值
      ab=edge(x1,x2,y1,y2);                         //求边长
      ac=edge(x1,x3,y1,y3);
      bc=edge(x2,x3,y2,y3);
      s=(ab+ac+bc)/2;                               //求边长和的一半
      area=sqrt(s*(s-ab)*(s-ac)*(s-bc));            //计算面积
      cout<<"area= "<<area<<'\n';                   //输出三角形面积
}
```

另外，在程序中使用到的所有变量也必须做到先声明再使用。在编写好源程序之后，用计算机系统提供的文本编辑器或语言本身提供的编辑器将其输入计算机中，并编译检查，直到修改正确后，再连接运行得出结果。这就是要用计算机解决实际问题的全过程。当然，解决一个真正的大的实际应用问题，比这个过程要复杂得多。

1.3 面向过程的程序设计方法

随着应用需求的不断扩大和变化，软件生产的方法和效率远远跟不上社会发展的需要。

软件工作者开始考虑如何提高软件质量和生产效率，即使用系统方法代替个人经验、智慧、技巧等，使建立软件系统的过程遵从一系列规范化阶段，包括需求分析、概要设计、详细设计、实现、组装测试、运行和维护等。这就将软件设计工作推进到软件工程时代。

结构化程序设计被称为软件发展中的第三个里程碑，其影响将比前两个里程碑（子程序、高级语言）更为深远。结构化程序设计的概念，最早是由 Dijkstra 提出的。

Tom.De Marcod 在《结构化分析与系统规格说明》教材中提出基于模型的软件工程概念，认为对于复杂软件系统的创建，必须首先为它们建立系统的书面模型。另一个有影响的软件理论是 Niklans Wirth 提出的“算法+数据结构=程序”。将软件划分成若干个独立的可以单独命名并分别编写的部分，称之为模块。模块化设计思想使软件能够有效地管理、维护、分析和处理复杂问题。在 20 世纪 80 年代，软件工作者普遍接受了模块化程序设计方法，接下来就是争论如何建立模块。有人认为最佳途径是使用函数，有人认为每个模块只应容纳一个数据结构，有人认为每个模块只应做一件事，还有人提出了事件驱动概念。这些代表性的解决方案也促进了程序设计语言的发展。

C 语言是结构化程序语言，它的程序设计特点就是函数设计。所谓函数，就是模块的基本单位，是对处理问题的一种抽象。例如，函数 sqrt(x)就是一个求平方根值的功能抽象。给定一个 x 的值，就可以得出这个值所对应的平方根值。如 sqrt(16)=4，sqrt(9)=3，16 和 9 为函数 sqrt 的参数。把一切逻辑功能完全独立的或相对独立的程序部分都设计成函数，并让每一个函数只完成一个功能。这样，一个函数就是一个程序模块，程序的各个部分除了必要的信息交流之外，互不影响。相互隔离的程序设计方法就是模块化程序设计方法。C 语言的这种程序结构化和模块化设计方法，特别适合于大型应用程序的开发。它解决了过去组成大型系统时所产生的多文件的组织与管理问题。

在程序规模比较大时，一般是根据结构化程序设计方法将程序划分成多个源文件。在编译该程序时，可以按一个个源文件为单位分别进行编译并产生与之对应的目标文件，然后再用链接程序把所生成的多个目标文件链接成一个可执行文件。C 语言的这种编译过程称为分块编译。

C 语言的这种分块编译处理方式就可以使一个程序同时由多个人进行开发，为大型软件的集体开发提供了有力的支持。分块编译的优点还在于修改一个源文件中的程序后，并不需要把整个程序的所有文件重新编译，这就大大节省了时间。

因为结构化程序设计方法是以函数过程和数据结构为中心的，所以面向过程的程序设计的关键是考虑如何使用结构化设计方法，使程序模块化。由此可见，现代计算机是面向算法的自动机，算法通过程序告诉计算机如何执行算法，构成程序的符号系统是语言，它是描述算法的工具。计算机语言（程序设计语言）的发展过程就是其功能不断完善、描述问题的方法逐步接近人类思维方式的过程。在计算机语言的发展过程中，新技术和新思想也不断出现。

因为 C++是混合型语言，所以可以使用 C++编译器所提供的对象，设计出更好的面向过程的软件系统。