高等学校计算机专业教材精选·算法与程序设计

C++程序设计

赵清杰 主编　　胡思康 宋 红 编著

清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

C++程序设计

赵清杰 主编
胡思康 宋 红 编著

清华大学出版社
北京

本书简介

本书系统地讲解了C++语言的基本语法及编程方法，介绍C++标准库特别是STL的主要组件及应用；内容包括C++语言概述及编程基础、函数与函数模板、类与类模板、运算符重载、继承与派生、多态、异常处理以及C++标准库等。针对初学者的特点，书中主要结合实例讲解基本概念和编程方法，力求通过简洁的实例让读者快速掌握C++语言，并能够熟练使用C++标准库进行程序设计。

本书内容全面、语言简洁、重点突出、实用性强，既适合于作为高等学校的教材，也适合作为培训班教材及自学参考书。

图书在版编目（CIP）数据

C++程序设计/赵清杰主编；胡思康，宋红编著. —北京：清华大学出版社，2008.9
（高等学校计算机专业教材精选·算法与程序设计）
ISBN 978-7-302-18297-9

Ⅰ. C…　Ⅱ. ①赵…　②胡…　③宋…　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(2008)第116784号

责任编辑：焦　虹
责任校对：李建庄
责任印制：杨　艳

出版发行：清华大学出版社　　**地　　址：**北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编：**100084
社　总　机：010-62770175　　**邮　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印　装　者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185×260　　**印　张：**15.25　　**字　数：**367千字
版　　次：2008年9月第1版　　**印　次：**2008年9月第1次印刷
印　　数：1～4000
定　　价：23.00元

出版说明

我国高等学校计算机教育近年来迅猛发展，应用所学计算机知识解决实际问题，已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在，很多高等学校都在积极探索符合自身特点的教学模式，涌现出一大批非常优秀的精品课程。

为了适应社会的需求，满足计算机教育的发展需要，清华大学出版社在进行了大量调查研究的基础上，组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材，把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生，使学生通过本套教材，学习计算机科学与技术方面的基本理论和基本知识，接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各校相应课程的主讲，有一定经验积累，且编写思路清晰，有独特的教学思路和指导思想，其教学经验具有推广价值。本套教材中不乏各类精品课配套教材，并力图努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则，书中的许多必须掌握的理论都将结合实例来讲，同时注重培养学生分析、解决问题的能力，满足社会用人要求。

(4) 易教易用，合理适当。本套教材编写时注意结合教学实际的课时数，把握教材的篇幅。同时，对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答，学生上机实验指导、教学网站等辅助教学资源，方便教学。

随着本套教材陆续出版，相信能够得到广大读者的认可和支持，为我国计算机教材建设及计算机教学水平的提高，为计算机教育事业的发展做出应有的贡献。

清华大学出版社

前　言

C++是广泛用于软件研发的大型语言，由于完备的功能和强大的解决现实问题的能力，所以极具学术研究价值和实际应用价值。

作者在多年教学的基础上，根据教学过程中反映的主要问题，对教材内容进行了合理组合与取舍，力求澄清概念上的误区，通过实例使读者尽快掌握C++语言的语法知识，把重点放在程序设计方法上，并加强对C++标准库的了解，掌握标准模板库的精华，对过程化编程思想、面向对象的编程技术和泛型编程技术有所了解，为今后的软件研发工作奠定基础。

本书具有如下特色：

(1) 将类和对象的概念提前介绍，从基本数据类型到结构再到类，可使读者尽快建立起"类"这种抽象数据类型的概念。在介绍对象的概念时，不用过于抽象的语言，而是具体针对计算机存储情况进行介绍。

(2) 与同类大部分教材相比，本书增加了指向成员的指针、成员函数地址获取、实现动态绑定的机制、函数对象等内容，加强了对C++标准库特别是对标准模板库STL的介绍。

(3) 采用好的编程习惯，如声明变量或对象时进行初始化等；使用不带后缀的标准库头文件；类定义时，采用以行为为中心的书写方式，以增强程序的可读性。

(4) 本书内容全面，语言简洁，重点突出；所列举的例题实用有趣，习题具有一定难度。

(5) 使用本书不需要有良好的C语言基础。

本书共分9章。

第1章简单介绍C++语言的特点以及C++标准库的主要构成。通过一个简单例子让初学者建立对C++程序的初步认识，然后介绍C++程序的编辑、编译、连接与运行，并介绍Visual C++ 6.0开发环境下建立标准C++控制台应用程序的步骤。

第2章的目的是让读者掌握C++语言的基本概念和基本语法，内容包括C++的词法规则、数据类型、表达式及语句、预处理命令、名字空间等。首先通过例子说明相应的语法知识，在此基础上使读者能够编写出简单的C++程序。

第3章主要介绍函数的定义与声明，包括函数调用、函数的参数传递及返回类型，inline函数，函数重载，带默认参数值的函数定义与使用，函数模板定义、重载与专门化等内容。

第4章介绍类与类对象的定义，详细讨论类的构造函数与析构函数，特别是构造函数的不同重载形式；并且介绍赋值成员函数、static成员及const成员，指向成员的指针，以及组合类、友元，类模板的定义、专门化等内容。

第5章讨论运算符重载的概念、规则及两种重载形式，给出几种特殊运算符的重载方法与应用实例，介绍函数对象的概念及应用。

第6章介绍与类的继承有关的一些概念，如继承与派生、基类与派生类、向上类型转换、单继承与多继承以及三种继承方式等，着重讨论在不同继承方式下基类成员的访问控制问题，讨论派生类的构造函数与析构函数，特别是复杂情况下子对象的构造与析构的顺序问题，分析继承与组合的区别，讨论多继承中可能存在的歧义性及解决方法。

第 7 章介绍静态绑定、动态绑定、虚函数、抽象类等概念，详细讨论虚函数以及动态绑定的实现机制，通过实例分析纯虚函数与抽象类的作用。

第 8 章主要介绍 C++ 语言的异常处理机制及带异常声明的函数，通过实例分析从对象的成员函数抛出异常的几种情况，介绍标准库中定义的异常类型。

第 9 章在对 C++ 标准库主要组件、特别是标准模板库的内容做更深入介绍的基础上，将通过更多的实例来说明如何使用标准库。

可为读者免费提供电子版的课件和习题答案，胡思康老师主编的《C++ 程序设计上机指导与习题答案》可与本书配套使用。

本书由赵清杰主编。王法胜、吴月吟等参与了本书的部分编写工作。

作者在本书的写作过程中参考了大量资料，大部分列入了书后的参考文献之中，还有一些没有收入其中，特别是部分内容参考了互联网上的共享资源，在此对这些作者表示衷心地感谢。

最后感谢您选用此书，欢迎您对本书内容提出宝贵意见和建议。

作者联系方式为：zhaoqingjie@tsinghua.org.cn

赵清杰

2008 年 8 月

目　录

第1章　C++语言概述

本章简单介绍C++语言的特点以及C++标准库的主要构成。首先通过一个简单例子让初学者建立对C++程序的初步认识；然后介绍C++程序的编辑、编译、连接与运行方法，以及在Visual C++ 6.0开发环境下建立标准C++控制台应用程序的步骤。本章涉及不少概念或名词，读者没有完全理解也无关紧要，通过后面的学习会逐渐理解的。

1.1　C++语言的特点

20世纪80年代，AT&T贝尔实验室的Bjarne Stroustrup博士研发出了C++语言，他的出发点是为了让编程人员能够更容易、更快捷地编写出高质量的程序。后来，C++语言得到了广泛应用。经过多年的不断完善和发展，1998年国际标准化组织公布了C++语言的国际标准，2002年又进行了修订，形成了今天的国际标准C++语言。国际标准（包括语法和程序库）的建立，简化了C++语言的使用以及在不同平台之间的移植工作，使用户可以根据爱好选用不同的C++编译器。

C++语言是支持多种编程思想的程序设计语言。它不仅支持过程式程序设计（procedural programming），还支持基于对象的程序设计（object-based programming）、面向对象的程序设计（object-oriented programming）以及泛型程序设计（generic programming）。

过程式程序设计通过一组算法建立问题的模型，程序的构成就是“程序＝算法＋数据”，这里的算法（或者操作/方法/过程）通过函数实现。C语言就是一门支持过程式程序设计的语言。C++语言能够兼容C语言，这样就可以保证原来的C库函数可以继续使用。C++语言在C语言的基础上做了很多改进。例如对类型要求更加严格，输入输出更加方便，增加了新的运算符，允许函数重载（overloading）和运算符重载，增加了引用（reference）类型，提出内嵌（inline）概念，提供常类型关键字const等。总之，在支持过程式程序设计方面，C++语言比C语言更安全，功能更强，使用更方便。

C++语言支持基于对象的程序设计。基于对象程序设计的主要特征是抽象和封装。它允许将数据和处理数据的操作封装在一起，形成抽象数据类型（abstract data type）。在C++语言中一般把这种抽象数据类型称为类（class），把占据一块内存、具有唯一地址的类的实例（instance）称为该类型的对象（object）。外界通过对象的公有接口与对象发生联系，调用对象的成员函数又称为向这个对象发送消息（sending a message）。基于对象的程序设计就是建立对象和向对象发送消息的过程。

C++语言支持面向对象的程序设计。抽象、封装、继承与多态是面向对象程序设计的主要特征。在类与对象的基础上，C++语言通过继承（inheritance）和动态绑定（dynamic binding）机制扩展了抽象数据类型。通过从基类派生出子类，用户无需修改原有的程序就可以重用已有的资源，并能够进一步修改或增添功能。动态绑定机制通过定义虚函数，使处于不同类层次的同名函数，在程序运行时才被决定应该调用哪一个。这种性质称为多态性

(polymorphism),该性质进一步提高了软件的可重用性和可扩展性。

C++ 语言支持泛型程序设计。泛型程序设计的思想是,将算法从特定的数据类型中抽象出来,使算法通用于不同的数据类型。泛型程序在代码重用、组织性、可维护性以及效率等方面表现出很强的优势。C++ 的模板语法机制,为范型程序设计奠定了关键基础。C++ 标准模板库 STL(Standard Template Library)就是以模板的形式定义了常用的数据结构和泛型算法。

1.2 C++ 标准库简介

C++ 标准库(standard library)是标准规格的一部分,它定义了一些可直接使用的函数、类、对象等。这些定义分别放在不同的头文件中,使用时只要包含相应的头文件即可。

以前的 C 头文件和 C++ 头文件名都是以".h"作为后缀的,如<stdio.h>和<iostream.h>。新的C++ 标准头文件名不带后缀,以字母 c 开头的标准头文件(如<cstdio>)等价于原来的 C 头文件。新标准头文件中的类和函数大都是基于模板定义的。为了叙述简单,本书在介绍标准库内容时有时会省略"模板"二字。

标准头文件中定义的标识符(如类名、函数名、对象名)都归属于名字空间 std,使用时要加前缀"std::"进行限制,否则要使用 using 声明或 using 指令。

下面简单介绍 C++ 标准库的主要内容。在第 9 章中将进行更详细的介绍。

C++ 标准库主要包括流类、string、容器类、使用容器的泛型算法、数值运算、C 标准库、语言支持功能等,其主要特点如下:

- C 标准函数库　基本保持了与原有 C 函数库的良好兼容。
- 语言支持　提供程序运行所需的功能,如动态存储分配、运行时类型识别等。
- 诊断　提供程序诊断和报错的功能,包括异常处理、断言(assertion)等。
- 通用工具(general utilities)　为 C++ 标准库的其他部分提供支持,也可以在自己的程序中调用相应的功能。
- 本地化　使体现自然语言或文化差异的特征本地化,如日期输出格式、货币表示符号等。
- 串(string)　用于表示和处理文本,比基于 char * 的字符串具有更佳的性能。
- 流类　提供对输入输出的基本支持,保持了以前流类库的功能,但是被模板化了;对继承层次结构也做了部分修改,增强了抛出异常的能力。
- 数值运算　包含一些数学运算功能,支持复数运算。
- 标准模板库(Standard Template Library,STL)　主要包括容器、算法、迭代器等,这部分提供了最常见的数据结构以及操作这些数据结构的基本算法。

通过图 1-1 可以简单说明 C++ 标准库的构成。在 C++ 标准库中,STL 占据相当重要的地位,它包含了计算机科学领域中许多常用的数据结构和基本算法。

初学者容易把 C++ 语言与 C++ 的开发环境(如 Visual C++)弄混,认为后者是前者的升级版本,这是错误的。C++ 是一门计算机语言,而 Visual C++ 是编译调试 C++ 程序的一种软件开发工具,类似的开发工具还有很多。我们要学习的是 C++ 这门计算机语言的语法知识、编程方法以及 C++ 标准库的使用。

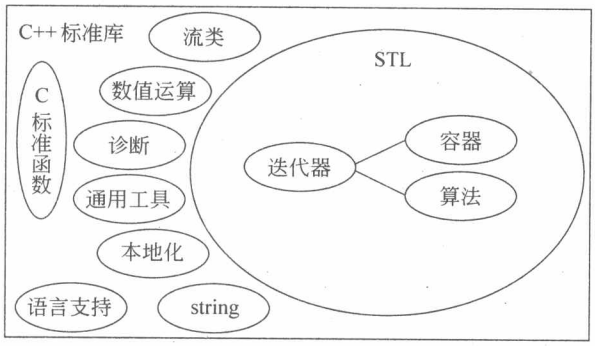

图 1-1　C++ 标准库的构成

基于标准 C++ 编写的程序，不受开发环境的影响，可移植性好。来自不同企业的开发环境一般也为用户提供一个基础类库，如 Microsoft Visual C++ 的 MFC（Microsoft Foundation Class Library）、Borland C++ Builder 的 VCL（Visual Component Library）等。此外，软件开发人员也可以编写自己的类库。

本书只涉及 C++ 标准库，对不同的开发环境及其类库不做具体介绍。在编译调试 C++ 程序时，应对自己所用的软件开发环境有大致的了解。

1.3　简单的 C++ 程序

进行程序设计时，主要是进行数据描述和数据处理两方面的工作。数据描述是把要处理的信息描述成计算机可以接受的数据形式，数据处理是指对数据进行输入、计算、存储、维护、输出等操作。下面通过例 1-1 说明如何编写计算圆面积的 C++ 程序。

```
//*********************************************************
//例 1-1. 计算圆面积的 C++程序
//ex1-1.cpp
//数据描述:半径和面积均为实数
//数据处理:(a)从键盘输入半径 r;(b)计算面积=πr²;(c)向屏幕输出半径和面积
//*********************************************************
#include<iostream>                          //包含标准头文件 iostream
using namespace std;                        //声明可以直接使用 std 中的标识符
//---------------------------------------------------------------------
int main()
{
    double r=0.0;                           //定义半径 r,初值赋为 0.0
    double area=0.0;                        //定义面积 area,初值赋为 0.0
    const double pi=3.14159;                //定义常数 pi=3.14159
    cout<<"Please input radius:"<<endl;     //向屏幕输出"Please input radius:"
    cin>>r;                                 //从键盘输入一个数值,并赋给 r
    area=pi*r*r;                            //计算圆的面积
    cout<<"radius="<<r<<endl;               //输出半径,endl 表示换行
    cout<<"area="<<area<<endl;              //输出面积
```

```
    return 0;                          //如果主函数没有显式返回语句,则标准 C++默认返回 0
}
```

上述程序运行后,屏幕上显示:

```
Please input radius:
```

如果从键盘输入:

```
3↵
```

屏幕上将显示:

```
radius=3
area=28.2743
```

这里 main 是函数名,称为主函数。函数体用一对花括号括住,int 表示主函数的返回值类型为整型。一个 C++ 程序可以由多个文件构成,但必须包含且只能包含一个主函数。C++ 程序从 main()函数的第一个"{"开始,依次执行后面的语句。如果在执行过程中遇到其他函数,则调用其他函数;调用完后返回,继续执行下一条语句,直到最后一个"}"为止。在标准 C++ 中,如果 main()函数没有显式提供返回语句,则默认返回 0。

程序由语句构成,每条语句由";"作为结束符。注意语句中的引号、分号等应采用英文模式,如果输入的是中文模式则会出错。

cin 和 cout 是系统预定义的流类对象,这里知道 cin 表示键盘、cout 表示屏幕就行了。"<<"表示输出(如输出到屏幕),">>"表示输入(如从键盘输入)。这些对象和操作都是在标准库中定义的。

预处理命令"#include <iostream>"的作用是:将头文件"iostream"中的代码嵌入该命令所在的位置。注意新标准 C++ 的头文件名没有后缀,而带后缀".h"的头文件是老版本的。老版本的库中没有定义名字空间 std。

使用#include 命令时,如果包含的是 C++ 系统头文件,则用一对尖括号将文件名括起来,目的是告诉编译器直接到系统目录下寻找;如果包含的是用户自己定义的头文件,则用一对双引号将文件名括起来,目的是告诉编译器先搜索当前目录,如果找不到再搜索系统目录。

std 是标准 C++ 预定义的名字空间,其中包含了对标准库中函数、对象、类等标识符的定义,包括对 cin、cout、endl 的定义。程序中 using 指令的作用是:声明 std 中定义的所有标识符都可以直接使用。如果没有"using namespace std;"这句声明,则要在 cin、cout、endl 的前面加上"std::"进行限制。

在 C++ 语言中,有两种注释方法供选择使用。一种以"//"开始,后面直到本行末尾的文字为注释;另一种以"/*"开始,以"*/"结尾,中间的文字均为注释,可以包含多个行。注释的作用是对程序进行注解和说明,以便于阅读。编译系统在对源程序进行编译时不理会注释部分,所以注释内容不会增加最终产生的可执行代码的大小。

利用输入/输出流类,可以很方便地进行数据的输入与输出。比如在屏幕上输出以八进制、十六进制等格式表示的数据,参见例 1-2。

```
//*********************************************************
//例 1-2. 数据的输出
//ex1-2.cpp
//*********************************************************
#include <iostream>          //包含标准头文件 iostream
using namespace std;         //声明可以直接使用 std 中的标识符
//---------------------------------------------------------------------
int main()
{
    cout<<"a number in decimal:"<<dec<<15<<endl;          //以十进制输出
    cout<<"in octal:"<<oct<<15<<endl;                     //以八进制输出
    cout<<"in hex:"<<hex<<15<<endl;                       //以十六进制输出
    cout<<"a floating-point number:"<<3.14159<<endl;      //输出浮点数
    cout<<"a char:"<<'A'<<endl;                           //输出字符 A
    return 0;
}
```

运行结果：

```
a number in decimal: 15
in octal: 17
in hex: f
a floating-point number: 3.14159
a char: A
```

1.4 程序的编辑、编译、连接与运行

要得到一个用 C++ 语言设计的可执行文件，通常需要经过编辑、编译、连接几个步骤。首先用编辑器编辑一个 C++ 源程序，如前面的 ex1-1. cpp；然后利用 C++ 编译器对源程序进行编译，形成目标文件 ex1-1. obj；再通过连接程序将目标文件变成可执行文件。在多文件应用程序(如一个程序中包含多个. cpp 文件)中，编译过程会产生多个目标文件，连接时要将多个目标文件以及需要的库文件连在一起，最后生成一个扩展名为 exe 的可执行文件。

在编译和连接时，一般会对程序中的错误进行检查，并将查出的错误显示在屏幕上。编译阶段查出的错误是语法错误，连接时查出的错误是连接错误。在编辑、编译、连接的过程中一般需要反复调试与修改才能得到正确的可执行文件。

现在的程序开发一般都使用集成开发环境。目前，Windows 环境下的常用 C++ 集成开发环境主要有 Microsoft Visual C++ 6.0、Visual C++ 7.0(即. net)以及 Borland C++ Builder 等。在 Linux 环境下可以采用 gcc 编译源程序。注意较早发布的 Visual C++ 6.0 编译系统在对标准 C++ 的支持上存在一些问题，请下载 Visual C++ 6.0 Service Pack 6 安装。下载地址为：http://msdn2. microsoft. com/en-us/vstudio/aa718359. aspx。

下面以例 1-1 中的程序为例，介绍在 Visual C++ 6.0 开发环境下，建立标准 C++ 控制台应用程序的步骤：

(1) 启动 Visual C++ 6.0 开发环境。

(2) 创建一个新项目。

- 单击 File 菜单中的 New 选项，显示 New(新建)对话框。
- 单击 Project(项目)标签，在 Project 选项卡中，选择 Win32 Console Application (Win32 控制台应用程序)。在 Location(位置)文本框中为新项目指定一个文件夹，在 Project Name(项目名称)文本框中为项目输入一个名字(如 myproject)，单击 OK 按钮。
- 在弹出的 Win32 Console Application-Step 1 of 1 对话框中选择 An Empty Project 单选项，然后单击 Finish(完成)按钮。
- 最后在 New Project Information 对话框中单击 OK 按钮，完成项目的建立。

(3) 建立 C++ 源程序文件。

- 选择菜单命令 Project|Add to Project|New，弹出 New 对话框。
- 在 New 对话框的 Files 选项卡中选择 C++ Source File，并填入文件名称(如 ex1-1)。单击 OK 按钮，完成新建 C++ 源程序文件。这时在 myproject 文件夹下就多了一个 ex1-1.cpp 文件。如果在 New 对话框的 Files 选项卡中选择 C/C++ Header File，则可以新建 C++ 头文件。用同样的方法，可以在 myproject 中新建多个.h 文件和.cpp 文件。如果选择菜单命令 Project|Add to Project|Files，则可以将已有的文件加入 myproject。

(4) 编辑 C++ 源程序文件内容。

- 在 ex1-1.cpp 文件编辑窗口中输入例 1-1 中的 C++ 代码。
- 选择菜单命令 File|Save 保存这个文件。

同一个项目可以包含多个头文件和.cpp 源程序文件。在例 1-1 中只包含了一个.cpp 文件。

(5) 编译、连接与执行。

- 选择菜单命令 Build|compile ex1-1.cpp，对源程序进行编译，产生目标文件 ex1-1.obj。如果正确输入了源程序，此时便成功生成了目标文件 ex1-1.obj；如果程序有语法错误，则屏幕下方的状态窗口中会显示错误信息；根据这些错误信息对源程序进行修改后，重新选择菜单命令 Build|compile ex1-1.cpp，对源程序进行编译。如果项目中有多个源程序(.cpp)文件，则可以分别进行编译。
- 选择菜单命令 Build|Build myproject.exe，可以对多个目标文件进行连接，例 1-1 只含一个目标文件。如果连接正确则生成可执行文件 myproject.exe。
- 选择菜单命令 Build|Execute myproject.exe 运行程序，在提示下输入一个半径值，观察屏幕上的显示内容。

(6) 关闭工作空间。

选择菜单命令 File|Close Workspace 关闭工作空间。

1.5 小　　结

C++ 语言是一门支持多种编程方法的程序设计语言。它不仅支持过程式程序设计，还支持基于对象的程序设计、面向对象的程序设计以及泛型程序设计。