21世纪高等院校计算机应用规划教材

Visual C++ 程序设计教程

编著 赵璐 吕俊
李斌 吉祖鹏

南京大学出版社

21世纪高等院校计算机应用规划教材

Visual C++ 程序设计教程

编著 赵 璐 吕 俊
李 斌 吉祖鹏

南京大学出版社

图书在版编目(CIP)数据

Visual C++程序设计教程/赵璐等编著. —南京:南京大学出版社,2009.1
21世纪高等院校计算机应用规划教材
ISBN 978-7-305-05699-4

Ⅰ. V… Ⅱ. 赵… 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213483号

出 版 者 南京大学出版社
社 址 南京市汉口路22号 邮 编 210093
网 址 http://press.nju.edu.cn
出 版 人 左 健

丛 书 名 21世纪高等院校计算机应用规划教材
书 名 Visual C++程序设计教程
编 著 赵璐 吕俊 李斌 吉祖鹏
责任编辑 吴华 吴宜锴 编辑热线 025-83592146

照 排 南京南琳图文制作有限公司
印 刷 南京人民印刷厂
开 本 787×1092 1/16 印张20.25 字数487千
版 次 2009年1月第1版 2009年1月第1次印刷
ISBN 978-7-305-05699-4
定 价 39.50元

发行热线 025-83594756
电子邮箱 nupress1@public1.ptt.js.cn

前　言

由 Dennis Ritchie 于 20 世纪 70 年代创建的 C 语言以简洁、高效和良好的移植性将程序设计带入了一个新的时代，而 Bjarne Stroustrup 在 C 语言的基础之上设计的 C++语言，通过向 C 语言加入面向对象的特性为程序设计开创了一个新纪元。作为 C 语言的超集，C++为程序设计员提供了强大的功能，可以产生模块化程度高、重用性和可维护性好的程序。目前 C++在各个领域都得到了广泛的应用，成为最流行的程序设计语言之一。

本书面向程序设计初学者，以 Visual C++6.0 为编程环境介绍了 C++程序设计的基础知识。全书共 15 章，分两部分：第一部分为 VC++程序设计基础，共 8 章，循序渐进地介绍了 VC++的基本语法及结构化程序设计的基本知识和技巧；第二部分为面向对象的程序设计，共 7 章，介绍了面向对象的基本概念和 OOP 程序设计的基本知识。另外，本书在附录中收录了 ASCII 码表、C++关键字一览表、基本运算符优先级和结合性一览表以及常用函数简表，便于读者进行相关方面的查阅和参考。

作为程序设计的入门教材，编者在编写过程中力求从读者的角度出发，由浅入深地安排内容、简洁而准确地阐述概念，同时配以详实的图表。在例题的组织上，除了紧密围绕知识点、由简到繁地引入例题，更是对例题进行了简单的分析或点评，突出例题中的知识点和编程技巧，便于读者理解和学习。在每章内容后，还安排了大量的习题，这些习题从易到难地帮助读者在理解、掌握基本概念和知识点的基础上，一步一步提高编程能力。

本书涉及到的所有例题均在 VC++6.0 环境下编译运行过，稍加修改也可运行在其他的C++编译系统下。为了帮助读者更好地复习和练习，本书配套了一本实验

指导书供读者选用。

本书可作为普通高校程序设计课程的教材，也可作为程序设计初学者的自学用书。其中，第一部分作为C++程序设计的基础，建议重点讲解。第二部分的第9章（类和对象）、第12章（运算符重载）和第13章（继承和派生类）是重点；也是难点。若学时较少，第15章（C++流和文件流）可不讲授。

本书的编者都是长期工作在计算机程序设计语言教学一线的教师，有着丰富的教学经验，对程序设计语言的初学者的学习情况比较熟悉。因此编者将对教学过程的体会、经验和教训融会到了教材的编写中，力求为读者呈现一本实用而易于理解的教材，但由于编者的水平有限，错误和疏漏在所难免，敬请读者提出宝贵意见。

编　者

2008年11月

目　录

1 C++概述

1.1 C++简介

自从有了计算机，也就有了计算机的编程。最初的计算机编程语言是机器语言，即直接使用机器代码(二进制)编程。机器代码的执行效率很高，但编程繁琐、易错，程序的可读性和可维护性较差。为了提高编程效率，人们引入了助记符，出现了汇编语言，虽然程序的可读性和可维护性得到了很大的改善，但是汇编语言同机器语言并没有本质的区别，编程时涉及太多的细节，程序的编写也与具体的计算机相关。

早期的计算机由于速度慢、内存小，所以衡量程序质量高低最重要的指标是机器执行的效率。但随着计算机技术的发展，机器硬件的性能大幅度提高，程序的复杂度也在增加，程序的可读性和可维护性渐渐成为衡量程序质量高低的最重要的指标。在这样的形势下，高级语言应运而生。1954 年，出现了世界上第一种计算机高级语言，用于科学计算的 Fortran 语言。随着计算机的进一步推广应用，大量的高级语言开始出现，如 Basic、Algol、Pascal、Cobol、C 等，其中 C 语言是使用最广泛、影响最大的语言之一。

1972 年，美国贝尔实验室的 Dennis M. Ritchie 成功开发了 C 语言。最初 C 语言是专为计算机专业人员设计的，主要用来编写 UNIX 操作系统。随着 C 语言的不断改进，它的功能越来越丰富，加上使用灵活、移植性好，不仅具有高级语言的优点，又具有低级语言的很多特点，C 语言迅速被广泛接受，成为程序设计的主流语言。

在 C 语言创建之后不久，出现了新的概念：面向对象的程序设计(OOP)。面向对象的理念很快吸引了程序员的注意，因为它提供了一种强大的新方法来完成程序设计工作。随着程序变得越来越大，其复杂度也在增加，因此需要采取一些措施来处理这种复杂性。OOP 提供了一种解决方案，OOP 将复杂的大程序划分为功能性的单元(对象)，这样做使得复杂的系统分解为容易管理的部分。

1983 年，贝尔实验室的 Bjarne Stroustrup 在 C 语言的基础上引入了面向对象的概念，在 C 语言中加入了 OOP 需要的新的关键字和语法，创建了 C++(读作 C Plus Plus)语言。C++语言增强了 C 语言的能力，使得程序员能够改进编写程序的质量，并易于程序代码的复用。C++语言的 ISO 标准已在 1997 年 11 月被一致通过，1998 年 8 月被正式批准。

C++是 C 的超集，保留了 C 语言的所有功能，在实现 OOP 的同时也可以兼容运行 C 语言的程序，这使得 C++一经推出就立刻被程序员所接受。随着功能的不断完善，C++语言已经成为最流行的计算机语言之一。

1.2 C++程序示例

编写程序是为了解决问题，解决问题的方法和步骤称为算法。算法只有按照指定的模式表达才可以被计算机理解并执行，不同的程序设计语言对应的表达模式是不一样的。下面介绍一个简单的C++程序。

【例1-1】 计算两个整数的和。

程序如下：

```
/*例1-1计算两个整数的和*/
#include<iostream.h>
void main()
{
    int a,b,sum;
    cout<<"请输入被加数和加数:"<<endl;
    cin>>a>>b;
    sum=a+b;                                    //求两数之和
    cout<<"两数之和为:"<<sum<<endl;
}
```

这个程序不长，但是从数学角度来看，语句行“sum=a+b;”已经可以解决问题了。那么其他代码行的作用又是什么呢？这就和前面谈到的“表达模式”有关了。

1.2.1 程序框架

所有的C++程序都必须按照一个固定的格式书写，只有按照这样的格式书写，计算机才能“读懂”并予以执行。

1. 包含头文件

一般来说，程序都会从包含头文件开始书写，如例1-1中的“#include<iostream.h>”。头文件，可以理解成由系统预先编写的程序文件，用来实现常用的功能，如输入/输出操作、一些常用的数学函数等等。这样，用户在需要这些功能的时候就可以直接使用而不需要再自行编程，从而节省了编程时间，提高了编程的效率。但用户在使用这些系统提供的程序段时需要通过包含头文件的方式即“#include<头文件名>”的形式预先向系统说明，如例1-1中需要用到输入/输出操作，就必须包含和输入/输出功能相关的标准输入/输出流的头文件“iostream.h”。关于包含头文件的详细情况请参考第四章。

2. 主函数main()

函数是C++程序最基本的程序架构，是功能上相互独立的程序段。所有的C++程序都是由若干个函数构成的，但每一个程序必须有也只能有一个主函数——“main”函数。“main”函数一般存放程序的主要代码，是程序运行的起点。“main”函数的基本格式如下：

```
void main( )                                    //函数的首部定义
{                                               //函数的开始
```

```
    语句;                                    //函数中的语句序列
    ……;
}                                            //函数的结束
```

函数的第一行是函数的首部定义，用来定义函数的返回值、函数名和参数表，读者现在不用深究具体的规则和含义，只需要按照此处定义形式来定义“main”函数即可。函数名“main”后的一对空括号“()”一定不能少。

大括号对“{}”标示了函数体的开始和结束，必须成对出现。函数体是指用来实现函数功能的程序代码，由若干条语句组成，每一条语句必须以分号“;”结尾。

包含了头文件，定义了“main”函数框架后，系统已经可以辨识这个C++程序了，但要让计算机顺利解决问题，还需要编写适当的函数体。对应不同的问题，实现函数体的程序肯定是不一样的，但是，程序中涉及到的很多基本元素却是大致相同的。下面，结合例1-1的程序介绍一下C++程序代码中一些最基本的元素。

1.2.2 程序基本元素

1. 变量

例1-1的程序是为了计算两个整数的和。程序需要解决的不是某两个指定整数的求和，而是任意两个整数的求和问题。所以，在程序的求和式中，被加数不是某个特定的整数，而是一个符号a，同样，加数、和也分别用符号b和sum表示。这样，在程序运行的时候，对a、b指定不同的值，sum就能得到相应的和，从而保证了程序的通用性。符号a、b、sum就叫做变量，在程序运行的过程中，变量的值可以发生变化。

变量在使用之前需要先定义，通过定义向系统说明该变量对应的是哪一类数据。如例1-1中的语句“int a,b,sum;”表示定义三个整型变量a、b、sum，其中“int”用来说明变量的类型为整型。如需要定义带小数的变量，则可用“float”定义。关于变量定义的具体规则将在第二章中进一步介绍。

2. 表达式

用运算符将变量等数据连接起来的式子叫做表达式，如例1-1中求和语句包含的式子“sum=a+b”就是表达式。其中“a+b”叫做算术表达式，用算术符号“+”连接了变量a和b，这和数学中的概念是一致的。数学中的减法对应C++中的算术符号为“-”，乘法的对应符号是“*”。数学中的除法在C++中的表示有些特殊，在后续章节中再进行讨论。

除了算术表达式“a+b”之外，求和语句行中还有一个表达式叫做赋值表达式，即“sum=a+b”。C++中的符号“=”和数学中的等于概念不同，它叫做赋值号，是个动态的概念，表示把赋值号右边的值赋给赋值号左边的变量。这里的赋值表达式是将变量a、b的和赋给变量sum。

3. 输入/输出

在例1-1中可以看到cin开头和cout开头的语句，从字面上就可以看出这是和输入、输出相关的语句。前面提到过标准输入/输出流头文件“iostream.h”，事实上，cin、cout就是

这个头文件中预先定义好的用于实现输入、输出的对象名，在学习对象的概念之前，我们可以简单地把 cin、cout 理解为用以实现输入/输出功能的语句。

cin 和符号“>>”配合使用，用于将用户从键盘输入的值依次赋给输入符号">>"后的变量，变量通过这种方式来获得初值。cout 和符号“<<”配合使用，用于将符号“<<”后的输出项依次输出至显示器，一般用来输出程序的运行结果及一些提示信息。输出项为“endl”时表示换行。

4. 分隔符和标点符号

分隔符是用来分隔单词或程序正文的，它用来表示某个程序实体的结束和另一个程序实体的开始，最常见的分隔符是空格。其他的 C++分隔符包括回车键、Tab 键和标点符号。

C++中使用的标点符号共有 9 个：

#	(	)
{	}	,
:	;	…

分隔符的具体使用在后续章节的相关部分再作进一步的介绍，但目前读者要注意的是：C++中出现的标点符号都是西文标点符号。

5. 注释

注释一般用来对程序或语句的功能进行说明，注释的目的是便于阅读程序。注释不属于 C++的语句，在程序编译阶段，注释将被忽略。

C++中，添加注释方法的方法如下：

/ * 注释 * /

或

//注释

前一种注释方法一般单独成一行或一段，用来对整个程序或一段代码进行注释说明。比如例 1 - 1 中的第一行：

/ * 例 1 - 1 计算两个整数的和 * /

第二种注释方法一般跟在语句行的后面，用来对该行语句进行注释说明。比如例 1 - 1 中用到的主函数定义语句行：

void main( )　　//函数的首部定义

在编写程序的时候，要养成添加注释的习惯。适当的注释不仅可以方便阅读程序，对于程序的修改也有很大的帮助。

运行例 1 - 1 程序后，屏幕上首先输出以下一行提示信息：

请输入被加数和加数：

如用户输入：

2　3↙

则屏幕输出：

两数之和为：5

经过变量定义、变量输入、表达式计算和结果输出几个步骤，例 1 - 1 的程序编写完成

了。而用户编写的C++程序又是如何输入计算机，怎样被执行的呢？

1.3 C++程序的实现

C++程序的输入及执行都是在相应的C++程序开发环境中完成的，在介绍具体的开发环境之前，先来看一下C++程序从输入到最后运行所要经历的步骤。

1.3.1 C++程序的实现步骤

(1) 输入代码，生成源程序。所有的高级语言编写的程序输入计算机后都是以源程序的形式保存的。C++的源程序是扩展名为“.cpp”的文件。

(2) 对源程序进行编译，生成目标程序。用高级语言编写的源程序计算机是不能识别和执行的，必须通过特定的语言处理程序将源程序代码翻译成二进制形式的代码。这个翻译的过程叫做“编译”，而编译后得到的源程序的二进制形式叫做“目标程序”，是扩展名为“.obj”的文件。编译还要进行词法检查和语法检查，凡是有词法和语法错误的源程序是不能生成目标程序的，只有改正了发现的错误，重新编译通过后才能生成目标程序。

(3) 连接目标文件，生成可执行文件。编译后会出现一个或多个目标文件，利用专门的连接程序将多个目标文件和程序涉及到的库文件连接后，将生成一个可直接运行的可执行的二进制文件，扩展名为“.exe”。

(4) 运行可执行文件，得到运行结果。如果结果不正确，要分析、改正源程序，重新编译、连接后再运行可执行程序，直到得到正确的运行结果。

C++程序的实现过程如图1-1所示。

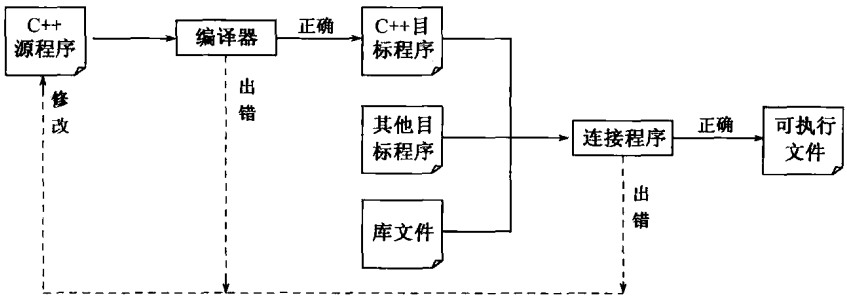

图1-1 C++程序实现步骤示意图

1.3.2 C++上机环境

许多软件厂商都提供了自己的C++集成开发环境，称为C++ IDE。著名的有Borland公司的C++ Builder，IBM公司的Visual Age For C++，Microsoft公司的Visual C++等等。其中，Visual C++6.0(以下简称VC6.0)集源程序的编写、编译、连接、调试、运行，以及应用程序的文件管理于一体，是当前PC机上最流行的C++程序开发环境。

VC6.0集成开发环境，被划分成四个主要区域：菜单和工具栏、工作区窗口、代码编辑

窗口和输出窗口，如图 1－2 所示。

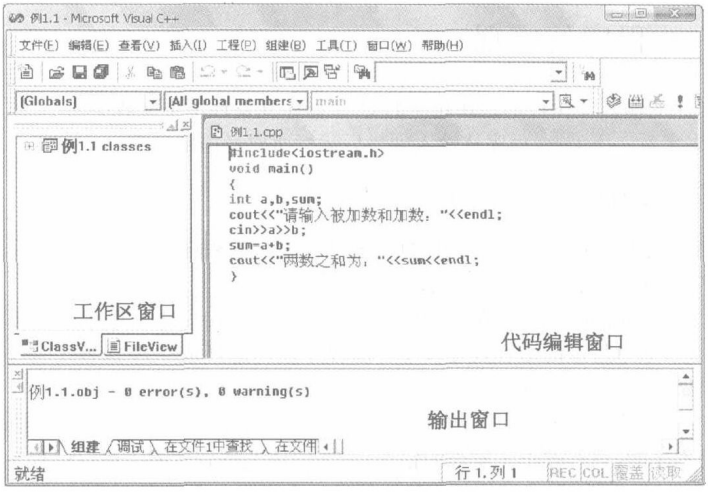

图 1－2　VC 6.0 集成开发环境

菜单和工具栏提供用户编辑、编译和运行程序等操作。新建或打开文件选择“文件”菜单中的相关子菜单，程序的编译、连接和运行选择“组建”菜单下的相关子菜单。

C++是用项目的形式来管理应用程序的，每一个应用程序都会对应于一个项目。工作区窗口是以视图的方式显示当前项目中的所有文件以及所有的类，方便用户管理。

代码编辑窗口供用户输入和编辑源程序。输出窗口用来输出编译时发现的语法错误，只有当输出窗口输出为“0 error(s)，0 waring(s)”时，才表示编译顺利完成可以进行连接了。

这里大致介绍 VC6.0 的集成环境，具体的上机步骤可参考本书配套的实验指导书。

习题一

1. C++程序中必须有也只能有一个的函数是什么函数？

2. 一个 C++程序从编辑到运行需要经过哪几步，每一步生成的文件是什么？

3. C++程序如下，参照第 1.2.2 节的内容，分析每一行分别对应哪些基本的程序元素。并分析程序的运行结果。

```
#include<iostream.h>
void main()
{
    int a,b,c;
    a=5;
    b=10;
    c=a*b;
    cout<<"a*b="<<c<<endl;
}
```

4. 现需要编写一个 C++程序，根据键盘输入的长方形的高和宽，计算并输出长方形的周长和面积。

长方形的高、宽、周长、面积分别用变量 h、w、c、s 表示，请试着将程序补充完整。

```
#include<iostream.h>
void main()
{
    float h,w,s,c;
    cout<<"请输入长方形的高:";
    cin>>_____;
    cout<<"请输入长方形的宽:";
    cin>>w;
    s=_____;
    c=_____;
    cout<<"长方形的面积=";
    cout<<_____<<endl;
    cout<<"长方形的周长=";
    cout<<_____<<endl;
}
```

5. 上机运行第 4 题对应的程序，验证程序的正确性。

2 C++语言基础

简单地说，程序就是用某种计算机语言来解决问题的方法和步骤。而解决问题的根本就是对已知的相关数据进行处理的过程。要处理怎样的数据，该如何处理，是首先要解决的问题。这一章，就将讨论C++中的数据以及和数据相关的概念。

2.1 标识符

标识符是一个字符序列，用来标识操作、变量、函数、数据类型等。任何程序都离不开标识符，不可能出现没有标识符的程序。在例1-1的程序中，include、a、main、int等都是标识符，分别用来标识包含头文件的预处理操作、变量名、主函数名和整数类型。

标识符必须按照一定的规则进行命名：

(1) 标识符由英文字母(包括大小写字母)、数字(0～9)和下划线(_)组成，必须以字母或下划线开头；

(2) 标识符中字符的个数不能超过255个；

(3) 严格区分大小写，即cout和Cout、COUT表示不同的标识符。

2.1.1 关键字

关键字，又称为保留字，是系统预先定义的具有特定含义的标识符。如例1-1的程序中用到的标识符void、int都属于C++的关键字。关键字一般都由小写字母组成，或以下划线开头、下划线和小写字母共同组成。注意不要将小写关键字写成大写或大小写混合，如int是关键字，而Int和INT都不是关键字。

标准C++中共定义了63个关键字，另外还定义了11个运算符关键字。其中和VC++兼容的是43个关键字，而VC++又增加了19个专用关键字。读者不一定会用到所有的关键字，一些常用、重要的关键字会在后续章节中进行详细的介绍。完整的关键字列表请参考附录。

2.1.2 用户自定义标识符

除了关键字外，用户可以自定义标识符用来命名变量、函数等。用户自定义的标识符除要遵守标识符的命名规则外，还需要注意以下几点：

(1) 所有的关键字都不可用于用户自定义标识符；

(2) 在定义标识符时，虽然语法上允许用下划线开头，但是，最好避免定义用下划线开头的标识符，因为编译器常常定义一些下划线开头的标识符；

(3) 有些标识符虽然不是关键字，但C++语言总是以固定的形式用于专门的地方，也不

能把它们当作用户自定义标识符使用，以免造成混乱。这样的标识符有 include、define 等。

(4) 为了增加程序的可读性，使程序更加清晰易懂，标识符应该尽量有意义。比如表示年可以用 year，表示长度可以用 length，表示累加和可以用 sum 等。

2.2 数据类型

编写程序的最终目的就是处理数据。数据在计算机中都是以二进制的 0、1 组合的形式表示的，而许多数据转换为二进制后的 0、1 组合形式是相同的。比如说，字母“A”转换后是“01000001”，而数字“65”转换后也是“01000001”，那么计算机又如何分辨这两个相同的二进制串从而进行不同的处理呢？这就需要在输入数据的同时说明数据的类型，这样，计算机就能在存储数据的时候加以区别，使得在需要的时候进行正确的处理。

所以，只有在交付给计算机需要处理的数据同时，配合适当的数据类型才能让计算机正确地识别和恰当地处理数据。

C++把数据类型划分为基本的数据类型和复合的数据类型(又称为构造的数据类型)，具体的划分如图 2-1 所示。

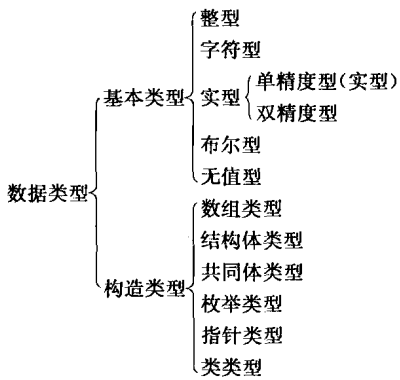

图 2-1 C++数据类型划分

本章重点讨论基本的数据类型，构造类型将在后续的章节中陆续介绍。

基本类型中，整型、实型、双精度型均属于数值类型。根据数值使用的不同场合，可以有正负的区别和表示范围的不同。C++定义了四个关键字来对这些情况加以标示和区别。分别是：

有符号(signed)　　无符号(unsigned)

长(long)　　短(short)

这四个关键字可以单独作为前缀修饰数据类型，也可以组合修饰数据类型，修饰后对应的数据在计算机中占用的字节数和取值范围都将有所不同。

字符型数据是用来表示英文字母和数字字符的。在计算机中以 ASCII 码的形式加以存储。而 ASCII 码本身是一个取值在 0～127 之间的整数，所以，字符型数据也可以用来存储取值较小的整数，这时可以用“signed”或“unsigned”关键字进行修饰。