高等学校计算机专业教材精选·算法与程序设计

C++程序设计教程

杨国兴 宋晏 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++程序设计教程

杨国兴 宋晏 编著

清华大学出版社
北京

内 容 简 介

本书从实际编程需要出发，通过大量实例介绍 C++ 语言以及面向对象程序设计方法，主要内容包括数据类型与表达式、C++ 控制语句、函数、数组、指针、类与对象、继承与派生、多态型、模板、输入输出流等。

本书可作为大专院校 C++ 程序设计或面向对象程序设计教材，同时也可供使用 C++ 进行程序开发的技术人员参考。

图书在版编目（CIP）数据

C++ 程序设计教程/杨国兴，宋晏编著. --北京：清华大学出版社，2012.12
高等学校计算机专业教材精选·算法与程序设计
ISBN 978-7-302-29007-0

Ⅰ. ①C…　Ⅱ. ①杨… ②宋…　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(2012)第 122852 号

责任编辑：白立军　顾　冰
封面设计：傅瑞学
责任校对：时翠兰
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**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
课件下载：http://www.tup.com.cn,010-62795954
印 装 者：北京嘉实印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：17.25　　**字　　数**：419 千字
版　　次：2012 年 12 月第 1 版　　**印　　次**：2012 年 12 月第1 次印刷
印　　数：1～3000
定　　价：29.50 元

产品编号：045999-01

前　言

C++是广泛使用的计算机程序设计语言之一。C++语言是由C语言发展而来，它保留了C语言的特点，同时又完全支持面向对象的程序设计。

本书重点介绍实际程序设计需要的知识，以及对于理解C++语言有重要帮助的内容，而不是面面俱到，纠缠语法细节。比如要记住运算符的优先级是非常不容易的，其实这些内容并不是很重要的，只要适当使用括号就可以解决问题，并且使用括号还可以提高程序的可读性。

在介绍C++语言的具体内容时，尽量使用较简单的例子，通过实例掌握语法知识。本书的所有例题都在Visual C++ 6.0环境下调试通过。

为了方便教师教学与学生学习，作者为使用本书的教师提供PowerPoint电子教案，方便教师根据具体情况进行有针对性的教学。

本书共分11章，主要内容如下：

第1章介绍C++与面向对象程序设计的特点，以及Visual C++ 6.0开发环境。

第2章介绍C++的基本数据类型、常用的运算符以及结构、联合、枚举等数据类型。

第3章介绍C++的各种控制语句，重点介绍选择结构与循环结构程序的实现。

第4章介绍C++函数的定义和使用、参数与返回值、递归调用、内联函数、函数重载、带默认参数值的函数以及变量的存储类别等。

第5章介绍数组的定义及使用，包括一维数组、二维数组、字符数组。

第6章介绍指针与指针变量的概念、指针变量的运算、指针与数组及指针与字符串的关系、动态内存分配等内容。

第7章介绍类的有关知识，包括类与对象的概念、构造函数与析构函数、类的组合、友元、类的静态成员以及对象数组与对象指针等内容。

第8章介绍继承与派生的有关内容，包括继承方式、派生类的构造与析构过程、多重继承与虚基类等。

第9章介绍运算符重载、虚函数以及抽象类等内容。

第10章介绍函数模板和类模板。

第11章介绍I/O流类库的层次结构、格式化输入输出、I/O流类库的文件输入输出功能等。

本书的编写工作主要由杨国兴、宋晏负责，严婷、谢永红、庄凤娟、杨国文、王国芳、庄莉等也参与了部分工作。

本书的编写得到了“十二五”期间高等学校本科教学质量与教学改革工程建设项目和北京科技大学教材建设经费资助。

由于作者水平有限，书中难免有不妥之处，恳请专家与读者批评指正。

作　者

2012年3月

目　录

第 1 章　C++ 与面向对象程序概述

计算机程序设计语言是人与计算机交流的工具，可以说没有程序设计语言和程序设计，计算机就不能使用，也就是说：程序（软件）是计算机的必要组成部分。计算机首先要求人们不断地在程序设计上付出大量的创造性劳动，然后才能享受到它的服务。

C++ 是在 C 语言的基础上发展起来的，它继承了 C 语言的简洁、高效等特点，同时 C++ 是一种面向对象的程序设计语言，完全支持面向对象的程序设计。

本章主要介绍计算机程序设计语言的发展、面向对象程序设计与 C++ 程序设计的特点以及 Visual C++ 6.0 开发环境。

1.1　程序设计语言的发展

程序设计的任务就是用计算机懂得的语言（即程序设计语言）编写程序，然后交给计算机去执行。自从第一台电子计算机诞生以来，人们对计算机程序设计语言的研究就一直没有停止过。计算机程序设计语言的发展大致经过了三个阶段，即机器语言、汇编语言和高级语言。

1.1.1　机器语言

由计算机硬件系统可以识别的二进制指令组成的语言称为机器语言。计算机设计者把计算机可以完成的动作编辑成一个指令表，每种动作赋予一个二进制代码，并为机器的每种动作设计一种通用的格式：由指令码和内存地址组成的指令。一条指令就是一个固定长度的由指令码和地址码组成的二进制位串，这就是计算机唯一可以读懂的语言，一般称作机器语言。

程序员把要计算机完成的任务分解为一系列机器语言指令表（或指令系统）包括的“动作”，以指令序列的形式写出来，这就是机器语言程序设计。

1949 年，冯·诺依曼（von Neumann）研制的 EDVAC 计算机，首次把上述指令序列形式的程序与数据一同置于磁芯存储器中，从那时开始，计算机通过识别置于存储器中的由 0 和 1 组成的二进制指令序列，来依次执行指定的操作，这种工作方式一直延续到今天。

在计算机应用最初的十几年中，大多数计算机程序就是用机器语言编写的。这种语言虽然十分简单，机器可以直接识别，但对于程序员来说却很不方便。完成一个简单的计算公式也要编写几十条指令，编程工作枯燥而烦琐，程序冗长而难读，调试、修改、移植和维护都是难题。因此，早期的计算机应用不广，编程的专业性极强，人们逐渐感到用机器语言编程是计算机应用向前发展的瓶颈。

1.1.2　汇编语言

虽然机器语言可以被计算机直接识别，但对程序员来说却太不方便。于是，一种汇编系

统程序问世了，这种程序的功能是把一种“汇编语言”编写的程序翻译为“机器语言”形式的程序。

汇编语言是用人们比较习惯的符号来代替指令编码，例如用 ADD 来代替 001 表示加法操作，用 Move 来代替 010 表示数据移动。用符号代替二进制地址表示参加操作的数据，这样大大减少了编程工作的困难。后来又改进为“宏汇编语言”，一条宏汇编指令可以代替多条机器指令。人们用汇编语言或宏汇编语言写程序，通过汇编系统(assembler)把它们翻译成计算机唯一“看”得懂的机器语言程序，然后再令其执行。

使用汇编语言编程比使用机器语言编程要容易，另外由于汇编语言指令与机器语言指令基本上是一条对一条或一条对几条，所以汇编系统的程序开发也不太复杂。因此，汇编语言编程很快取代了机器语言编程。到了 20 世纪 60 年代，机器语言编程已经比较少了，汇编语言逐渐取代机器语言，成为主要的编程语言。

汇编语言和机器语言都属于低级语言，这是因为其语言的结构都是以面向机器的指令序列形式为主，与人的习惯语言方式距离较远，所以它们的共同缺点是：

- 依赖于机器，可移植性差。
- 代码冗长，不易于编写大规模程序。
- 可读性差，可维护性差。

1.1.3 高级语言

对于程序员来说，虽然汇编语言比机器语言方便很多，但仍然没有解决计算机编程难的基本问题。后来以 FORTRAN 和 ALGOL 60 为代表的高级语言逐渐流行。到了 20 世纪 70 年代，新一代的高级语言 Pascal 和 C 问世了。

与汇编语言和机器语言相比，高级语言更接近人类的自然语言，当然计算机也不能直接识别高级语言编写的程序，要通过编译程序将高级语言编写的程序翻译成机器语言程序(这一过程称为编译)，再让计算机运行。

高级语言的发展经历了高级语言编程的初级阶段、结构化程序设计阶段和面向对象程序设计阶段。

在高级语言刚出现的一段时期里，计算机的主要应用领域是数值计算，程序的规模通常也不是很大，随着计算机和计算机高级语言应用的不断发展，需要编写一些规模大，复杂度高，使用周期长，投入人力、物力多的大型程序，程序设计的目标把可靠性、可维护性的要求放在了比高效率更重要的位置上。高级语言编程初级阶段的程序设计方法已经不能满足程序需要不断扩大的要求。

为了解决这一问题，出现了结构化程序设计方法，结构化程序设计思想认为：

(1) 好程序的标准首先是它的可读性和可维护性，其次才是高效率；所谓可读性是指改善程序书写的静态结构，好的语言风格，结构清晰，符合人的阅读习惯，以及注意编程格式，给程序及程序中的变量和函数一个有意义的命名，增加必不可少的注释。

(2) 结构化程序设计主要采用自顶向下、逐步求精的设计方法和单入口单出口的控制结构。

(3) 自顶向下、逐步求精的设计方法符合人们解决复杂问题的普遍规则，可提高软件开发的成功率；由此开发出的程序具有清晰的层次结构，易于阅读理解及易于修改、调试和

扩充。

(4) 程序出问题的更主要原因，是来自程序执行中动态结构的混乱。结构化程序设计最主要的目标是尽可能地使程序运行的动态结构，与程序书写的静态结构相对地比较一致。

虽然结构化程序设计方法有很多优点，但仍然是一种面向过程的程序设计方法。它将数据和处理数据的过程分离为相互独立的部分，当数据结构发生变化时，相应的处理过程也要改变，随着程序规模的增大和复杂程度的提高，使得程序维护的成本迅速增加。

面向对象的程序设计方法将数据和处理数据的过程封装在一起，形成一个有机的整体(即类)，更符合人们通常的思维习惯，使开发的软件产品易重用，易修改，易测试，易维护，易扩充。

1.2 面向对象程序设计的特点

1.2.1 面向对象程序设计的基本概念

1. 类(class)

类描述了一组具有相同特性(数据元素)和相同行为(函数)的对象。如汽车、树、书和复数等都是类。不管哪一个汽车，虽然颜色、性能等不同，但都具有一些相同的特征，如都有最高时速、百公里耗油量、载重量等，因此将它们划分为一个类(汽车类)。对于书来说都具有书名、作者、出版社、出版日期、定价和页数等属性，因此将它们划分为一个类(书类)。

2. 对象(object)

对象是现实世界实际存在的事物，是类的一个具体实例，如某一辆具体的汽车、某一棵树、某一本书都是一个对象。

面向对象程序设计中的对象是系统中用来描述客观事物的一个实体，它是用来构成系统的一个基本单位。对象由一组属性和一组行为构成。

3. 属性(attribute)

类中的特性(数据)称为类的属性，如汽车的颜色、最高时速、载重量等是汽车类的属性，书类中的书名、作者、单价等是书类的属性。不同的类具有不同的属性。

4. 方法(method)

类中的行为(函数)称为类的方法，如汽车类可以有加速方法、刹车方法，转向方法等。不同的类具有不同的方法，既不能让树做汽车可以做的事情，也不能让汽车做树可以做的事情，如不能让树加速、转向等，也不能给汽车浇水让其长大。

1.2.2 面向对象程序设计的特点

封装、数据隐藏、继承和多态是面向对象程序设计的主要特征。

1. 封装和数据隐藏

封装和数据隐藏的思想来源于现实世界，如在电视机发生故障需要更换一个晶体管时，修理人员不需要自己去做一个晶体管，而是在已经做好的晶体管中找到一个合适的使用，因此对于维修人员来说，他并不关心晶体管内部的结构，他也没有必要为了使用晶体管而了解晶体管内部的工作原理，他关心的只是晶体管的各种参数及其与外界的连接。也就是说将

晶体管的内部数据和工作原理封装在晶体管内部，从而实现内部数据的隐藏，为使用晶体管的技术人员提供了极大的方便。

类似地，在面向对象程序设计中，也可以通过创建类（包含属性与方法）实现封装和数据隐藏。如创建汽车类，将汽车的内部数据和方法封装在一起，实现数据隐藏，使用汽车类的程序员并不需要掌握汽车类内部的具体细节，只需要了解汽车类的外部接口就可以了，为程序代码的重用提供了方便。

2. 继承

当需要设计新型汽车时，通常可以有两种选择，一种是从头开始设计，另一种是在已有的相近型号的基础上进行一些改进，即在已有型号的基础上加入一些新的特性，称之为继承，即新型车继承了原有车的特性，又增加了一些新的特性。

面向对象程序设计通过类的继承实现代码重用，即在已经存在类的基础上，定义一个新的类，新类继承已有类的属性和方法，并可以增加自己新的属性和方法。如人类有姓名、性别、年龄等属性，学生除了具有人类的所有属性之外，还有学号、成绩等属性。这样定义了人类以后，再定义学生类时，只要让学生类从人类继承，而不需要在学生类中重复声明姓名、性别、年龄等属性。

3. 多态

在面向对象的程序设计中，多态有两种类型，即静态多态和动态多态。静态多态是通过函数重载实现的，即多个函数可以具有相同的函数名，函数调用时可以根据参数的个数和类型确定调用哪一个函数；动态多态则是通过虚函数实现的。

1.3 C++ 语言的特点

C++ 是在 C 语言的基础上发展而来的，同时支持面向对象的程序设计，它主要具有以下特点：

（1）C++ 继承了 C 语言的所有特点。包括语言简洁、紧凑，使用方便、灵活；拥有丰富的运算符；生成的目标代码质量高，程序执行效率高；可移植性好等。

（2）对 C 语言的某些方面进行了一定的改进。如引入 const 常量和内联函数，取代 C 语言中的宏定义；引入 reference（引用）概念等。

（3）支持面向过程和面向对象的方法。在 C++ 环境下既可以进行面向对象的程序设计，也可以进行面向过程的程序设计。

C++ 完全支持面向对象的程序设计，包括数据封装、数据隐藏、继承和多态等特征。

1.4 简单的 C++ 程序

C++ 源程序文件的扩展名是 cpp，头文件的扩展名是 h。C 语言的输入输出是通过函数实现的，C++ 的输入输出是通过 I/O 流类实现的。为了在后面章节的程序中进行输入和输出，下面简单介绍如何使用流类对象 cin 和 cout 实现简单的输入和输出，有关这部分的详细内容可以参考第 11 章。

cin 和 cout 分别是 C++ 预先定义的输入流类对象和输出流类对象，在程序中可以直接

使用。

1.4.1 cout 标准输出

使用 cout 进行输出的格式如下：

cout<<待输出的内容 1<<待输出的内容 2…；

我们也称符号＜＜为插入运算符，即将其后面的数据插入到输出数据流中。由于 cout 和插入运算符＜＜都是在系统提供的头文件 iostream. h 中声明的，因此要包含该头文件。

例 1.1 使用 cout 进行简单的输出。

```
#include<iostream.h>
void main()
{
    cout<<"您好"<<endl;
    cout<<"这是一个简单的 C++程序"<<endl;
}
```

程序运行结果：

```
您好
这是一个简单的 C++程序
```

第 1 行是编译预处理指令，将文件 iostream. h 中的代码嵌入到该指令所在的位置，main()是主函数，C++ 程序都是从主函数开始运行的。函数中的语句由一对大括号括起来，每一语句由分号“;”结束。cout 是一个输出流类对象，通过插入运算符＜＜输出数据，其中 endl 是换行符号，表示后面的输出要在下一行上。

1.4.2 cin 标准输入

使用 cin 输入数据的格式如下：

cin>>保存数据的变量 1>>保存数据的变量 2…；

符号＞＞也称为提取运算符，即将输入数据流中的数据提取出来。由于 cin 和符号＞＞也都是在系统提供的头文件 iostream. h 中声明的，因此在使用时要包含该头文件。

例 1.2 使用 cin 进行简单的输入。

```
#include<iostream.h>
void main()
{
    int a,b;
    cin>>a>>b;
    cout<<a<<","<<b<<endl;
}
```

程序运行后，在键盘上输入 10　20，程序输出如下：

```
10, 20
```

在主函数 main()的最开始定义两个整型(int 表示整数)变量 a 和 b。cin 是一个输入流类对象,通过运算符>>输入数据,如果要输入多个数据,中间用空格分开。最后一行利用 cout 输出变量 a 和 b 的值。

1.5 Visual C++ 6.0 编程环境简介

为方便后面的上机练习,下面简单介绍 Visual C++ 6.0 开发环境以及程序的编辑、编译、运行和程序的调试等方法。

1.5.1 Visual C++ 6.0 界面介绍

Visual C++ 6.0 是运行于 Windows 环境下的交互式可视化集成开发环境。Visual C++ 6.0 的主界面如图 1.1 所示。

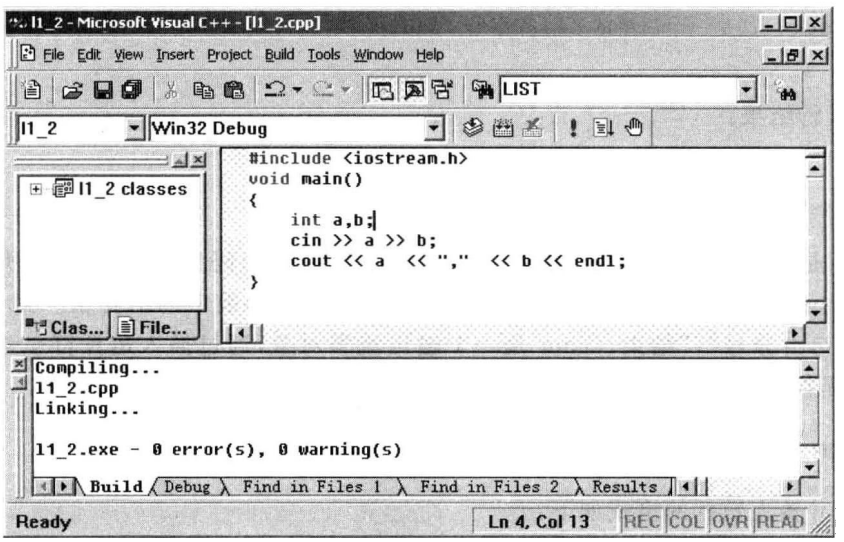

图 1.1 Visual C++ 6.0 集成开发环境

Visual C++ 6.0集成开发环境的主界面由标题栏、菜单栏、工具栏、工作区(Workspace)窗口、客户区、输出窗口和状态栏组成。主窗口的最上端为标题栏,用于显示应用程序名和当前打开的文件名,图 1.1 中程序名是 L1_2,当前打开的文件是 L1_2.cpp。标题栏的下面为菜单栏和工具栏。工具条下面为工作区和客户区,一般工作区位于左侧,客户区位于右侧。工作区包含三个标签,即 ClassView 标签、ResourceView 标签和 FileView 标签。ClassView 标签用于显示当前打开项目所包含类的信息;ResourceView 标签用于显示当前打开项目所包含的资源信息,包括对话框、图标、位图、菜单等资源(本书的例题不处理资源,因此不需要此标签);FileView 标签用于显示当前打开项目所包含各种文件的信息,包括源程序文件、头文件、资源文件等。客户区用于文本编辑、资源编辑等,图 1.1 显示当前正在编辑 L1_2.cpp 文件的内容。工作区和客户区的下面为输出窗口,用于输出编译信息、调试信息和查找结果信息等。最底部是状态栏。