高等学校计算机专业
面向项目实践规划教材

C++程序设计案例实践教程

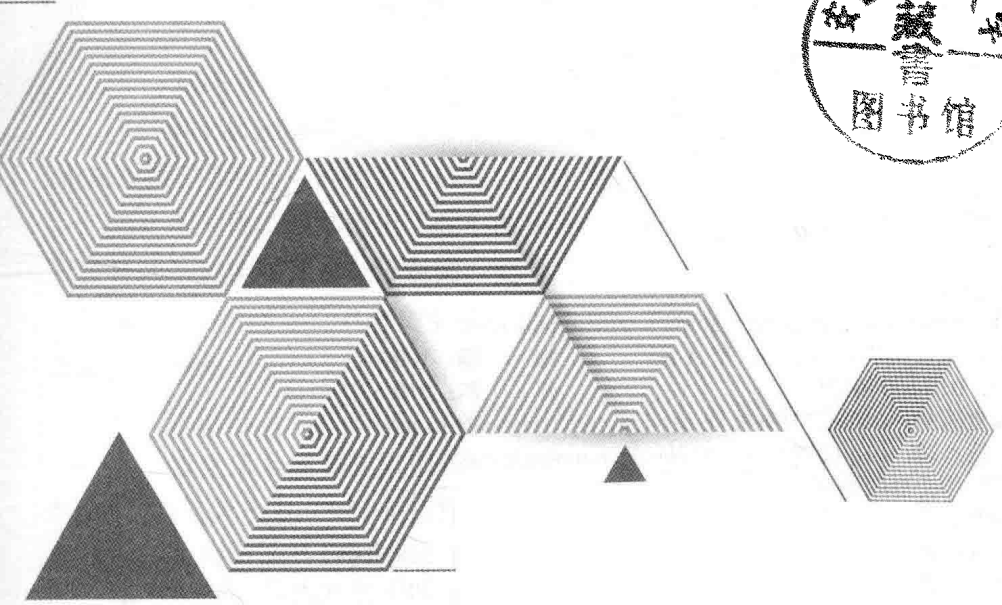
◎ 朱林 主编

清華大學出版社

高等学校计算机专业
面向项目实践规划教材

C++程序设计案例实践教程

◎ 朱林 主编



清華大學出版社
北京

内 容 简 介

本书从应用型人才培养的角度全面介绍了 C++语言程序设计的主要概念、基本语法及程序设计技巧等方面的内容，以简单实用为原则，讲解通俗易懂，行文流畅。在内容安排上由浅入深，让读者循序渐进地掌握 C++语言编程技术。

本书是多年来 C++教学内容和课程体系改革的综合成果，内容以面向工程实践和编程能力训练为主，具有较强的操作性和应用性，为 C++语言程序设计课程教学内容和课程体系改革构建了一个新的框架。本书从应用开发和数据存储的角度来写，贯穿全书安排两条线，一是应用线：用一个学生管理系统的开发贯穿全书，先从提示页面的显示再到增、删、改、查各功能的实现，先在 DOS 窗口下显示功能及操作，最后再用 MFC 和 ODBC 实现。二是数据存储线：先讲变量如何在内存中存储，再讲数组在内存中的存储，由物理相邻的存储结构（数组）的缺点过渡到与指针结合的逻辑相邻的存储结构——链表，讲解链表如何解决操作数组时存在的缺点，然后在输入输出流中讲述数据存放在文件中该如何操作，对比与数据存放在内存中（数组、链表）的不同，然后阐述文件存储是一种最初的数据仓库模型，从而引入简单的 Access 数据库用法，为学生学习以后的数据库知识打下基础，同时方便用户使用 MFC 和 ODBC 开发基于界面的管理系统。

本书可作为应用型高校工科类专业的“C++语言程序设计”“面向对象程序设计”“程序设计导论”等课程的教材，也可以作为各专业学生和工程技术人员进行项目编程时的教材及参考书籍。

图书在版编目（CIP）数据

C++程序设计案例实践教程 / 朱林主编. —北京：清华大学出版社，2018
（高等学校计算机专业面向项目实践规划教材）
ISBN 978-7-302-51265-3

Ⅰ. ①C… Ⅱ. ①朱… Ⅲ. ①C++语言-程序设计-高等学校-教材 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2018）第 212813 号

责任编辑： 贾　斌
封面设计： 刘　键
责任校对： 焦丽丽
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
印 装 者： 三河市君旺印务有限公司
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　张：** 24.5　　**字　数：** 598 千字
版　　次： 2018 年 10 月第 1 版　　**印　次：** 2018 年 10 月第 1 次印刷
印　　数： 1～1500
定　　价： 59.00 元

产品编号：071598-01

前言

FOREWORD

目前，很多高校向应用型高校转型，应用型人才的培养在高等教育中占的比重越来越大，本教材以案例教学的方式讲解C++语言程序设计，可以进一步提高学生的实践应用能力、开拓创新能力，能够促进应用型高校的转型步伐。

本书是多年来C++教学内容和课程体系改革的综合成果。内容以面向工程实践和编程能力训练为主，具有较强的操作性和应用性，为C++教学内容和课程体系改革构建了一个新的框架。本书从应用开发和数据存储的角度来写，贯穿全书安排两条线：

（1）应用线：用一个学生管理系统的开发贯穿全书，先从提示页面的显示再到增、删、改、查各功能的实现，先在DOS窗口下显示功能及操作，最后使用MFC和ODBC实现。

（2）数据存储线：先讲变量如何在内存中存储，再讲数组在内存中的存储，由物理相邻的存储结构（数组）的缺点过渡到与指针结合的逻辑相邻的存储结构——链表，讲解链表如何解决操作数组时存在的缺点，然后在输入输出流中讲述数据存放在文件中该如何操作，对比与数据存放在内存中（数组、链表）的不同，然后阐述文件存储是一种最初的数据仓库模型，从而引入简单的Access数据库的用法，为学生学习以后的数据库知识打下基础，同时方便用户使用MFC和ODBC开发基于界面的管理系统。

本书具有以下特点：

（1）本书内容广泛、案例丰富，其中的例题、习题及实践案例都来源于一线教学。

（2）本书按照读者在学习程序设计中遇到的问题来组织内容，随着读者对程序设计的理解和实际动手能力的提高，内容由浅入深地向前推进。

（3）本书每章后都给出了相应的案例实践，给出技能训练要点和任务实现，这些代码不仅能够与理论知识点无缝对接，而且短小精练，方便读者尝试完成。

（4）本书以学生信息管理系统项目案例贯穿始终，每章中的知识点则使用独立的案例，并辅以实例输出和任务实现。

（5）课后练习题覆盖面广，种类多样，方便读者巩固理论与实践知识。

本书特别适合培养应用型人才高校的工科类专业使用，可以作为“C++语言程序设计”“面向对象程序设计”“程序设计导论”等课程的教材，也可以作为各专业学生和工程技术人员进行编程时的教材及参考书籍。

本书在编写过程中得到了清华大学出版社和同行专家、学者的大力支持和帮助，在此表示衷心的感谢。此外，本书的编写参考了部分书籍和报刊，并从互联网上参考了部分有价值的材料，在此向有关的作者、编者、译者和网站表示衷心的感谢。

本书配有电子教案，并提供程序源代码，以方便读者自学，请到清华大学出版社官方网站下载。

由于编者水平有限，书中难免有不妥之处，敬请读者和专家批评、指正。

朱　林

2018 年 3 月

目录

CONTENTS

第 1 章

C++概述

1.1 C++语言简介

C++语言是一种应用广泛、支持多种程序设计范型的高级程序设计语言。C++语言是在 C 语言的基础之上发展起来的，它既适合于编写面向过程的程序，也适合于编写面向对象的程序，所以在一定程度上将其称为半面向过程半面向对象的语言。

自 1946 年第一台电子数字计算机 ENIAC 问世以来，随着计算机技术的高速发展，计算机应用领域不断扩大，程序设计语言作为计算机应用的一种工具也得到不断充实和完善。每年都有新的程序设计语言问世，同时，老的程序设计语言也在不断地更新换代。

在 C 语言推出之前，操作系统等系统软件主要是用汇编语言编写的，汇编语言的好处是能对硬件直接进行操作，速度快、效率高。但由于汇编语言依赖于计算机硬件，且写法较为底层，因此程序的可移植性和可读性较差。为了使语言的编写和可读性更接近于人的语言习惯，同时也为了提高高级语言的速度和效率，1973 年，贝尔实验室的 Thompson 和 Ritchie 开发了 C 语言，并用它重写了 UNIX 的大部分代码。C 语言具有以下的特点：

（1）C 语言是一种结构化的程序设计语言，语言本身比较简洁，使用起来比较灵活方便。

（2）它具有一般高级语言的特点，又具有汇编语言的特点。除了提供对数据进行算术、逻辑运算外，还提供了二进制整数的位运算。用 C 语言开发的应用程序，不仅结构性较好，且程序执行效率高。

（3）程序的可移植性好。在某一种计算机上用 C 语言开发的应用程序，其源程序基本上可以不作修改，在其他型号和不同档次的计算机上重新编译连接后，即可完成应用程序的移植。

（4）程序的语法结构自由度大。精通 C 语言的程序设计者正是利用这一特点，设计出高质量的通用应用程序。对于初学者来说，掌握 C 语言并不是一件容易的事。往往是源程序编译时容易通过，程序运行时出错，且这种错误不易解决。

随着 C 语言应用的不断推广，C 语言存在的一些不足也开始流露出来。例如，C 语言对数据类型检查的机制比较弱、缺少支持代码重用的结构，随着计算机应用面的推广和软件工程规模的扩大，难以适应开发特大型的程序、软件维护困难等。在使用它做较大型项

目编程的时候，由于数据的共享和函数的调用错综复杂，所以在维护方面会出现大量的问题。面向对象的技术就是在这种情况下产生的，1980 年，贝尔实验室的 Bjarne Stroustrup 博士及其同事对 C 语言进行了改进和扩充。在保持了 C 语言简洁、高效前提下，克服了 C 语言存在的不足，并把 Simula67 中类和对象的概念引入到 C 中。1983 年由 Rick Maseitti 提议将改进后的 C 命名为 C++（C Plus Plus）。后来又把运算符重载、引用、虚函数等功能加入到 C++中，使 C++的功能日趋完善，使之可以支持面向对象的程序设计，同时，它既可以支持 DOS 下的程序设计，也可以用来开发 Windows 环境下的应用程序。

C++除了继承 C 语言的一些特点之外，还具有以下特点：

（1）C++是 C 语言的一个超集，它基本上具备了 C 语言的所有功能。

用 C 语言开发的源程序代码可以不作修改或略作修改后，就可在 C++的集成环境下编译、调试和运行。这对推广或进一步开发仍有使用价值的软件是极为重要的，可节省人力和物力。

（2）C++是一种面向对象的程序设计语言。

C++从 C 语言中继承了过程化编程的高效性，并集成了面向对象程序设计的功能。除了 C++标准库中提供的大量功能。还有许多商业 C++库也支持数量众多的操作系统环境和专门的应用程序。编写出的面向对象的程序可大大增强程序的可读性和可理解性，使得各个模块的独立性更强、更好，程序代码的结构性更加合理。这对于设计和调试大的应用软件是非常重要的。

（3）用 C++语言开发的应用程序，扩充性强、可维护性好。

首先，在应用软件的开发过程中，对要解决的实际问题有一个认识、理解，再进一步认识和理解，直至客观地弄清楚问题本质的过程。这种认识和理解的过程，往往伴随着可能需要改变程序的结构或功能，这就要求应用软件具有较强的可扩充性。其次，对于任何一个已开发的应用软件，随着时间的推移和应用的深入，常要求增加或扩充新的功能、改进某些功能或修正程序故障。这均要求所设计的程序具有可扩充性和可维护性的特点。所以，对于较大应用程序的设计，这一特点非常重要。

（4）C++适用的应用程序范围极广。

C++几乎可用于所有的应用程序，从字处理应用程序到科学应用程序，从操作系统组件到计算机游戏等。C++还可用于硬件级别的编程，例如实现设备驱动程序。

1.2 本章知识目标

（1）了解 C++语言的基本概念和特点。

（2）了解 C++应用程序的基本结构，学会编写简单的 C++应用程序并了解编程中需要注意的问题。

（3）学会使用 C++的开发环境，会在开发环境下对应用程序进行编译、构建、运行等操作。

（4）学会 C++语言输入输出的写法，会使用输入输出语句做简单的输出测试及实现应

用程序的界面。

1.3　程序

其实，日常生活中人们在不断地创造程序并执行程序，只不过并没有明确地意识到而已。举个例子，要用全自动洗衣机洗衣服，应该怎么做呢？

第一步把脏衣服扔进洗衣机；

第二步打开上水的水龙头并安装好电源插头；

第三步放入洗衣粉；

第四步按下洗衣机的开始按钮；

第五步等待洗衣机洗完衣服（当然，这段时间不妨去干点别的事情）。在洗衣机提示洗完的蜂鸣声响了以后，就可以从洗衣机中拿出干净衣服去晾晒了。

上面所描述的五个步骤，就是人们洗衣服的“程序”。也许不同的人使用的步骤并不完全一样，例如将第一步和第二步互换一下，也同样能将衣服洗干净，所以干一件事的“程序”可以不唯一，这也是计算机程序的一个特点。

对于计算机来说，程序就是由计算机指令构成的序列。计算机按照程序中的指令逐条执行，就可以完成相应的操作。实际上计算机自己不会做任何工作，它所做的工作都是由人们事先编好的程序来控制的。程序需要人来编写，使用的工具就是程序设计语言。

1.4　C++程序基本结构

程序结构是程序的组织结构，是指该程序语言特定的语句结构、语法规则和表达方式，其内容包括代码的组织结构和文件的组织结构两部分。只有严格遵守程序结构的规则，才能编写出高效、易读的程序，否则写出的代码将晦涩难懂，甚至不能被正确编译运行。

本章通过一个简单程序向读者介绍 C++程序的基本结构，同时也说明 C++程序中简单的输入输出操作方法，以方便后续章节中的讲解。

例 1-1　一个简单的 C++程序。

```
/*第一部分*/
//这是一个演示程序
/*第二部分*/
#include <iostream.h>
/*第三部分*/
void main()
{
    cout<<"This is a c++ program";
}
```

C++程序通常会包括示例中所示的三部分。

第一部分是整个文件的注释部分，注释内容是为了增加程序的可读性，系统不编译注释内容，自动忽略从“/*”到“*/”之间的内容。C++中以“//”开头直到本行结束的部分也是注释。与“/*……*/”的区别在于“//”只能注释一行，不能跨行，这种注释也称为行注释，而“/*……*/”注释可以跨行，称为多行注释或块注释。

第二部分是预处理部分。它是在编译前要处理的工作。这里是以#include 说明的头文件包含代码#include <iostream.h>，它指示编译器在预处理时将文件 iostream.h 中的代码嵌入到该代码指示的地方。其中#include 是编译指令。头文件 iostream.h 中声明了程序需要的输入输出操作的信息，在 C++中，用标准输入设备（键盘）和标准输出设备（显示器）进行输入输出时，使用输入输出流中的对象 cin（输入）和 cout（输出）来完成，它们的定义就属于头文件 iostream.h，所以不使用#include <iostream.h>就不能使用上述的输入输出对象。需要注意的是，在 Visual Studio 中，有时还会看到#include <iostream>的引入方式，这也是 Visual Studio 中鼓励使用的方式。但是采用这种方式时，还需要用 using namespace std; 引入 std 命名空间。

在编译源程序时，先调用编译预处理程序对源程序中的编译预处理指令进行加工处理后，形成一个临时文件，并将该临时文件交给 C++编译器进行编译。由于编译预处理指令不属于 C++的语法范畴，为了把编译预处理指令与 C++语句区分开来，每一条编译预处理指令单独占一行，均用符号#开头。根据编译预处理指令的功能，将其分为三种：文件包含、宏和条件编译。

文件包含是在一个源程序文件中的任一位置可以将另一个源程序文件的全部内容包含进来。include 编译预处理指令可实现这一功能。该编译预处理指令的格式为：#include <文件名>，编译预处理部分的内容将在后面章节中详细介绍。

第三部分是代码的主要部分，它实现了一个函数，结构如下：

```
void  main()
{
    …

}
```

程序中定义的 main()函数又被称为主函数，其中 main 是函数名，void 表示该函数的返回值类型为空。任何程序必须有一个且只能有一个主函数，且程序的执行总是从主函数开始，其他函数只能被主函数调用或通过主函数调用的其余函数调用。关于函数定义及调用的内容会在后面章节做详细介绍。

程序中的代码 cout<<"This is a C++ program";在执行后屏幕上会出现 This is a C++ program 这句话，即 C++输出语句中双引号中的内容会被原样输出。

1.5 C++程序的调试与运行

针对一个实际问题，用 C++语言设计一个实用程序时，通常要经过如下五个开发步骤。

（1）用户需求分析。根据要解决的实际问题，分析用户的需求，并用合适的方法、工具进行详细描述。

（2）根据用户需求，设计 C++源程序。利用 C++的集成环境或某一种文本编辑器将设计好的源程序输入到计算机的一个文件中。文件的扩展名为 cpp。

（3）编译源程序，并产生目标程序。在编译源程序文件时，若发生语法或语义错误时，要修改源程序文件，直到没有编译错误为止。编译后，源程序会产生目标文件。在 PC 上，目标程序文件的扩展名为 obj。

（4）将目标文件连接成可执行文件。将一个或多个目标程序与库函数进行连接后，产生一个可执行文件。在 PC 上，可执行文件的扩展名为 exe。

（5）调试程序。运行可执行程序文件，输入测试数据，并分析运行结果。若运行结果不正确，则要修改源程序，并重复以上的过程，直到得到正确的结果为止。

1.5.1　用 Visual C++开发环境运行程序

Visual Studio 6.0 为用户提供了良好的可视化编程环境，程序员可以利用该开发环境轻松地使用 C ++源代码编辑器、资源编辑器和内部调试器，并且可以创建项目文件。Visual C++开发环境不仅包含编译器，而且还包含了许多有用的组件，通过这些组件的协同工作，可以在 Visual C++集成环境中轻松地完成创建源文件、编辑资源，以及对程序的编译、连接和调试等各项工作。

（1）启动 Visual C++开发环境。

成功地安装了 Visual C++以后，可以在“开始”菜单中的“程序”选项中选择Microsft Visual Studio 6.0 级联菜单下的Microsft Visual C++6.0 命令，启动 Visual C++，进入 Visual C++ 6.0 的集成环境，如图 1-1 所示。

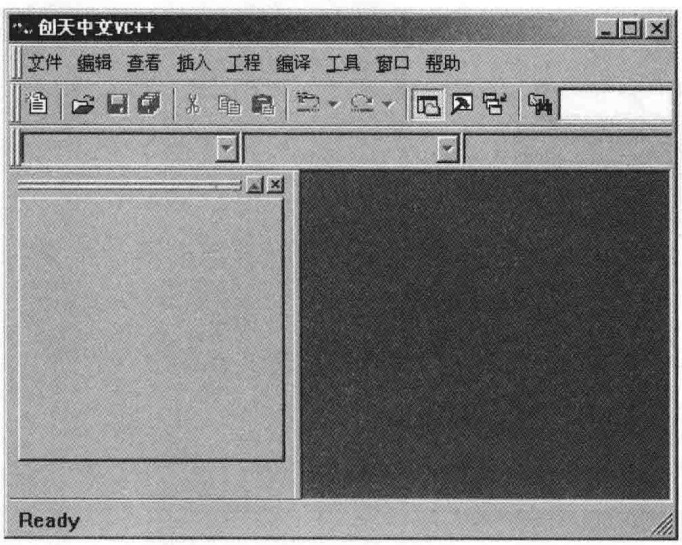


图 1-1　Visual C++界面

（2）创建项目。

若开始一个新程序的开发，可以先用 AppWizard（应用程序向导）建立新工程项目。