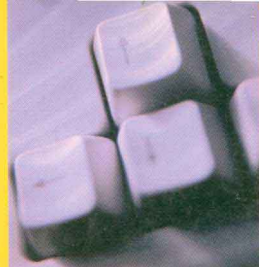




普通高等教育“十五”国家级规划教材

高等职业教育技能型紧缺人才培养培训系列教材



程序设计基础

——面向对象及C++实现

安志远 贾振华 主 编
张 昕 张燕文 副主编



高等教育出版社

普通高等教育“十五”国家级规划教材
高等职业教育技能型紧缺人才培养培训系列教材

程序设计基础

——面向对象及 C++ 实现

安志远 贾振华 主编

张 昕 张燕文 副主编

高等教育出版社

内容提要

本书是普通高等教育“十五”国家级规划教材,全书采用“任务驱动”的编写方式,语言通俗易懂,概念简洁、明确,内容由浅入深,循序渐进,具有较强实用性。

全书共计9章内容,包括:C++语言概述、类与对象、静态成员与友元、运算符重载、继承、虚函数和多态性、输入/输出流、模板和异常处理、综合实例等。

本书适合于高等职业学校、高等专科学校、成人高校、本科院校举办的二级职业技术学院的学生学习,也可供示范性软件职业技术学院、继续教育学院、民办高校、技能型紧缺人才培养使用,还可供本科院校、计算机专业人员和爱好者参考使用。

图书在版编目(CIP)数据

程序设计基础——面向对象及C++实现/安志远,贾振华

主编.—北京:高等教育出版社,2005.7

ISBN 7-04-016985-1

I.程... II.①安...②贾... III.①面向对象语言—程序设计—高等学校—教材②C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(2005)第056797号

策划编辑	冯 英	责任编辑	严 亮	封面设计	王凌波
版式设计	胡志萍	责任校对	王效珍	责任印制	杨 明

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100011
总 机 010-58581000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 北京蓝色畅想图书发行有限公司
印 刷 北京市联华印刷厂

网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>

开 本 787×1092 1/16
印 张 24.75
字 数 600 000

版 次 2005年7月第1版
印 次 2005年7月第1次印刷
定 价 29.90元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 16985-00

出版说明

为加强高职高专教育的教材建设工作,2000年教育部高等教育司颁发了《关于加强高职高专教育教材建设的若干意见》(教高司[2000]19号),提出了“力争经过5年的努力,编写、出版500本左右高职高专教育规划教材”的目标,并将高职高专教育规划教材的建设工作分为两步实施:先用2至3年时间,在继承原有教材建设成果的基础上,充分汲取近年来高职高专院校在探索培养高等技术应用性专门人才和教材建设方面取得的成就经验,解决好高职高专教育教材的有无问题;然后,再用2至3年的时间,在实施《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》立项研究的基础上,推出了一批特色鲜明的高质量的高职高专教育教材。根据这一精神,有关院校和出版社从2000年秋季开始,积极组织编写和出版了一批“教育部高职高专规划教材”。这些高职高专教材是依据1999年教育部组织制定的《高职高专教育基础课程教学基本要求》(草案)和《高职高专教育专业人才培养目标及规格》(草案)编写的,随着这些教材的陆续出版,基本上解决了高职高专教材的有无问题,完成了教育部高职高专规划教材建设工作的第一步。

2002年教育部确定了普通高等教育“十五”国家级规划教材选题,将高职高专教育规划教材纳入其中。“十五”国家级规划教材的建设将以“实施精品战略,抓好重点规划”为指导方针,重点抓好公共基础课、专业基础课和专业主干课教材的建设,特别要注意选择一部分原来基础较好的优秀教材进行修订使其逐步形成精品教材;同时还要扩大教材品种,实现教材系列配套,并处理好教材的统一性与多样化、基本教材与辅助教材、文字教材与软件教材的关系,在此基础上形成特色鲜明、一纲多本、优化配套的高职高专教育教材体系。

普通高等教育“十五”国家级规划教材(高职高专教育)适用于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院、继续教育学院和民办高校使用。

教育部高等教育司
2002年11月30日

前 言

随着计算机技术的快速发展,计算机应用逐渐普及和深入,同时计算机软件的需求也日益增长。C++语言是当前最流行和最实用的一种计算机程序设计语言,具有语言简洁、高效灵活、功能强大、运算符和数据类型丰富、可重用性好、扩展性强等优点。C++语言既支持面向过程的程序设计,也支持面向对象的程序设计。

面向对象的程序设计是当前最流行、最实用的软件开发手段。与传统的面向过程程序设计不同,面向对象程序设计认为现实世界所有事物都是由对象组成的,对象是由数据及数据上的操作构成的统一体。类是具有相同数据和相同操作的一组对象的抽象。通过继承和多态,实现了代码的可重用性及扩展性。程序中对象之间能够通过消息互相联系。这样,面向对象的软件开发方法,使开发软件的方法与过程尽可能接近人们认识世界解决问题的方法与过程,从而使描述问题的空间与解决问题的解决空间在结构上尽可能一致。

本书凝聚了作者多年的C++程序设计教学和科研经验,具有很强的实用性和可读性。针对学生的学习规律以及在学习过程中容易出现的问题,采用“任务驱动”的编写方式,激发学生的学习兴趣。同时,在每章的开始部分给出了学习目标,有助于学生对重点的把握。每章之后都提供了精心设计、难易适度的实训题目,读者可独立思考自己解决问题,以提高学习的兴趣。

本书是普通高等教育“十五”国家级规划程序设计基础系列教材之一。在编写过程中,充分考虑到由《程序设计基础——逻辑编程及C++实现》作为前导课程,在《程序设计基础——面向对象及C++实现》中只需把逻辑编程作概要介绍,通过这样安排:一方面复习前导课程学过的内容(如果逻辑编程已经学过,就可以略过这一部分);另一方面对于那些没有学过逻辑编程的人,也可以获取面向对象程序设计所必需的基础知识。在《程序设计基础——面向对象及C++实现》一书中,以面向对象的基本思想和基本方法为主要内容,重点在面向对象基本特征和基本技术的学习上,同时培养学生以面向对象方法来分析、解决实际问题的能力。本书共分9章,第1章主要包括C++语言概述、基本数据类型和表达式、函数、数组、结构体和共用体、指针和引用、结构化程序设计等C++基础内容。第2~8章主要是面向对象的程序设计部分,内容包括:类和数据抽象、运算符重载、继承性、虚函数和多态性、C++输入输出流、文件处理、异常等。第9章列举了两个综合开发实例,促进学生对知识的全面理解 and 应用。

本书在编写过程中,注意实际应用和理论知识相结合,尽量避免枯燥的理论讲授,通过使用大量的具有代表性、实用性和趣味性的程序实例来帮助读者掌握理论知识,从而指导自己的实践,使读者能够尽快掌握C++语言的理论和程序开发技术。同时,力求语言通俗易懂,概念简洁明确,内容由浅入深,循序渐进,突出实用性。本书思路清晰,重点在于培养学生分析问题和解决问题的能力,使读者在学完本书后能够真正学以致用。

本书选材适当、定位准确、安排合理、实例典型丰富、重点突出,书中所给实例均在Visual C++ 6.0集成环境中调试通过。

本书适合作为高职高专计算机及其相关专业学习“C++程序设计”课程的教材,也可作为广

II 前言

大计算机应用人员和面向对象程序设计爱好者的自学参考书。

本书由安志远、贾振华担任主编,张昕、张燕文担任副主编。各章编写分工如下:第1、2、3章由安志远编写,第4、5、6章由贾振华编写,第7、8章由张昕编写,第9章由张燕文、贾振华共同编写,崔玉宝、赵辉、庄连英、李杰、赵丽艳等也参加了部分内容的编写工作以及大纲讨论、资料收集和程序调试工作。全书由安致远、贾振华负责统稿。

由于时间仓促,作者水平有限,书中难免有一些疏漏和不足之处,恳请读者批评指正。

编 者
2005 年 4 月

目 录

第 1 章 C++ 语言概述	1	2.1.1 定义类和对象	71
学习目标	1	2.1.2 访问对象成员	76
1.1 面向对象程序设计入门	1	2.1.3 this 指针	79
1.1.1 程序设计和程序设计语言	1	2.2 构造函数	80
1.1.2 简单的 C++ 程序	2	2.2.1 构造函数的含义	80
1.1.3 创建和运行 C++ 程序	4	2.2.2 带参数的构造函数	82
1.1.4 VC++ 集成开发环境	10	2.2.3 重载构造函数	86
1.2 数据类型和表达式	12	2.2.4 复制构造函数	88
1.2.1 基本语法成分	12	2.3 析构函数	91
1.2.2 基本数据类型	13	2.4 对象的指针和对象的引用	94
1.2.3 变量和常量	14	2.4.1 对象的指针	94
1.2.4 运算符和表达式	17	2.4.2 对象引用	97
1.2.5 数组类型	24	2.5 对象数组	100
1.2.6 结构体和共用体	26	2.6 常类型	103
1.2.7 指针和引用	30	2.6.1 常指针和常引用	103
1.3 程序控制结构	40	2.6.2 常成员函数	105
1.3.1 语句	40	2.6.3 常数据成员	106
1.3.2 选择语句	40	2.7 应用实例	107
1.3.3 循环语句	44	2.8 实训任务	112
1.3.4 转向语句	48	第 3 章 静态成员与友元	119
1.4 函数	50	学习目标	119
1.4.1 函数的定义和说明	50	3.1 静态成员	119
1.4.2 函数的调用	51	3.1.1 静态数据成员	120
1.4.3 函数的参数	54	3.1.2 静态成员函数	124
1.4.4 函数的嵌套调用和递归调用	56	3.1.3 静态成员使用	129
1.4.5 作用域和存储类型	59	3.2 友元	132
1.4.6 内联函数	63	3.2.1 友元函数	133
1.4.7 函数重载	64	3.2.2 友元类	137
1.4.8 字符串处理函数	66	3.2.3 友元的使用	140
1.5 实训任务	70	3.3 实训任务	145
第 2 章 类与对象	71	第 4 章 运算符重载	146
学习目标	71	学习目标	146
2.1 定义类和对象	71	4.1 概述	146

4.2 运算符重载的一般规则	149	6.2 虚函数	267
4.3 用成员函数或友元函数 重载运算符	149	6.2.1 虚函数的定义	267
4.3.1 用成员函数重载运算符	149	6.2.2 虚函数与动态联编	269
4.3.2 用友元函数重载运算符	153	6.2.3 虚函数的重载	275
4.4 一些特殊操作符的重载	156	6.2.4 虚析构函数	281
4.4.1 重载++与--运算符	157	6.3 纯虚函数和抽象类	286
4.4.2 重载赋值运算符	167	6.3.1 纯虚函数	286
4.4.3 重载[]和()运算符	172	6.3.2 抽象类	290
4.4.4 重载插入(<<)和提取(>>) 运算符	176	6.4 多态	295
4.5 类型转换	179	6.4.1 多态的含义	295
4.5.1 构造函数用于类型转换	179	6.4.2 多态的应用	301
4.5.2 转换运算符用于类型转换	182	6.5 实训任务	305
4.6 应用实例	185	第7章 输入/输出流	308
4.7 实训任务	192	学习目标	308
第5章 继承	193	7.1 输入/输出流的概念	308
学习目标	193	7.2 标准流和流操作	309
5.1 基类和派生类	193	7.2.1 基本输入/输出操作	310
5.1.1 继承和派生的概念	193	7.2.2 输入/输出格式控制	314
5.1.2 派生类的定义	194	7.3 文件处理	318
5.1.3 类的继承方式	199	7.3.1 文件的打开和关闭	318
5.2 单继承	209	7.3.2 文本文件的访问	320
5.2.1 成员访问权限在继承中的控制	209	7.3.3 二进制文件的访问	322
5.2.2 单继承下的构造函数和 析构函数	212	7.3.4 文件的随机访问	323
5.3 多继承	227	7.4 实训任务	326
5.3.1 多继承的声明	227	第8章 模板和异常处理	327
5.3.2 多继承的构造函数和析构函数	232	学习目标	327
5.3.3 二义性问题	236	8.1 模板	327
5.4 虚基类	242	8.1.1 模板的概念	327
5.4.1 虚基类的声明	242	8.1.2 函数模板	327
5.4.2 虚基类及其派生类的构造函数	245	8.1.3 类模板	330
5.5 应用实例	249	8.2 异常处理	332
5.6 实训任务	256	8.2.1 异常处理的基本概念	332
第6章 虚函数和多态性	260	8.2.2 异常处理的实现	333
学习目标	260	8.3 实训任务	335
6.1 静态联编和动态联编	260	第9章 综合实例	336
		9.1 实例一	336
		9.2 实例二	352
		参考文献	384

第 1 章 C++ 语言概述

学习目标

- 理解程序设计的概念
- 掌握 C++ 程序的开发过程
- 掌握 C++ 的运算符和表达式
- 掌握 C++ 的数据类型,包括基本数据类型和构造数据类型
- 熟练掌握基本控制语句 if、switch、while、do~while、for 的使用
- 掌握转向语句 break、continue 的使用
- 熟练掌握函数的使用

1.1 面向对象程序设计入门

1.1.1 程序设计和程序设计语言

程序设计也是一门工程设计,它是根据要解决的问题,使用某种程序设计语言,设计出能够完成这一任务的计算机指令序列。

程序设计语言是人与计算机进行交流的一种形式语言,是人利用计算机分析问题、解决问题的一个基本工具。就如同人类社会生活中,自然语言是人与人之间用来表达意思、交流思想的工具一样。自然语言是由字、词、句法和语法等构成的一个系统;而计算机程序设计语言是由字、词和语法等构成的指令系统。

最初,程序开发人员使用的程序设计语言是原始的计算机指令,即能够被计算机直接识别的一系列二进制数,称为机器语言。

在机器语言的基础上,人们设计出汇编语言,它可以将机器语言用一些易于人们记忆和阅读的助记符来表示,如 ADD、SUB、MOV 等。计算机运行汇编程序时,首先将用助记符写成的源程序转换成机器能够识别的指令,然后再运行机器指令程序,得到所要的结果。

随着计算机应用的发展,人们开发出像 Fortran、Basic、Pascal、C、C++ 等高级程序设计语言,用一条语句可以完成大量任务。高级语言使程序的编写更像英语指令,可以包含常用的数学符号。从程序员的角度来看,高级语言比机器语言和汇编语言要强得多,其中 C 和 C++ 是使用最为广泛的高级语言。

在使用计算机程序设计语言设计程序时,要达到的目标是:在保证程序正确的前提下,力求程序可读性强、容易维护、移植性好。程序的可读性强是指程序要有良好的书写风格和用简单易懂的语句编写程序,书写风格包括语句的对齐、规范的注释等。容易维护是指当业务规则发生变化时,要求以最小的开销就可以对程序功能进行更改或增加。移植性好是指编好的程序可以在

不同的计算机和操作系统上都能运行,并且运行的结果一样。

程序语言的发展,总是从低级到高级,从具体到抽象,直到可以用人的自然语言来描述。

1.1.2 简单的 C++ 程序

【例 1.1】一个简单的 C++ 程序。

```
/*
    源文件名:Ch1_01.cpp
    功能:给出矩形的宽和高,计算矩形的周长和面积,使用结构化方法实现
*/
#include <iostream.h>
void main()

    int width,height,girth,area;
    cout<<"请输入矩形的宽和高:";           //提示输入信息
    cin>> width>> height;                   //输入两个数据
    girth= (width+ height) * 2;              //计算矩形周长
    area= width* height;                     //计算矩形面积
    cout<<"矩形:宽 = "<< width<<" 高 = "<< height<<endl; //输出数据
    cout<<"周长 = "<< girth<<endl;
    cout<<"面积 = "<< area<<endl;
```

程序运行后,屏幕显示:

```
请输入矩形的宽和高: 20 15
矩形:宽 = 20 高 = 15
周长 = 70
面积 = 300
```

下面来分析例 1.1 的程序结构。

(1) “/* */”是程序的注释部分,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,称为块注释。

(2) #include <iostream.h>是一条编译预处理命令,声明该程序要使用 iostream.h 文件中的内容,iostream.h 文件中包含了输入 cin 和输出 cout 的定义。编译时系统将头文件 iostream.h 中的内容嵌入到程序中该命令位置。C++ 中编译预处理命令都以 # 开头。C++ 提供了 3 类编译预处理命令:宏定义命令、文件包含命令和条件编译命令。例 1.1 中出现的 #include <iostream.h>是文件包含命令,其中尖括号内是被包含的文件名。

(3) 程序中定义了一个主函数 main(),其中 main 是函数名,void 表示该函数返回值类型。

程序执行从主函数开始。一个 C++ 的程序可以包含多个文件,每个文件又可以包含多个函数。函数之间地位是相互平行、相互独立的。一个 C++ 程序,必须有一个且只能有一个主函数 `main()`。执行程序时,系统先从主函数开始运行,其他函数只能被主函数调用或通过主函数调用的函数所调用,函数可以嵌套调用,即在一个函数中调用另外一个函数。主函数可以带参数,也可以不带参数。函数在调用之前,必须先定义好,定义函数要按照系统规定的格式进行,后面再详细介绍。

由 `{}` 括起来的内容是主函数 `main()` 的函数体,其中左大括号“`{`”表示函数的开始,右大括号“`}`”表示函数的结束。函数体部分由许多 C++ 语句组成,这些语句描述了函数的功能实现。

(4) 函数体中的“`int width,height,girth,area;`”语句是变量声明,通知系统为变量分配存储空间。

(5) 函数体中的其他语句用于完成相应功能,是函数体的重要组成部分,包括 4 条输出语句:`cout` 是标准输出流对象,指定显示器为标准输出设备;“`<<`”是 `cout` 中的插入运算符,表示把它后面的数据在输出设备上输出显示;双引号中的内容要原样输出,变量要显示具体的值;`endl` 表示回车换行,“`;`”表示语句结束,C++ 规定语句必须要用分号“`;`”结尾。

由以上分析可以看出,一个 C++ 程序的基本结构包括:以“`#`”开头的若干个编译预处理命令,将程序所需要的头文件包含进来;然后是定义主函数和其他函数,当然函数也可以在程序的起始部分先利用函数原型进行声明,以后再进行定义;用大括号“`{}`”括起来的部分就是函数体部分,函数体部分主要包括各种各样的语句和注释信息,这部分应该是程序的主体部分,占的比重也最大。

【例 1.2】用面向对象方法实现一个简单的 C++ 程序。

```
/*
    源文件名:Ch1_02.cpp
    功能:给出矩形的宽和高,计算矩形的周长和面积,使用面向对象方法实现
*/
#include <iostream.h>
#include <iostream.h>
class Rectangle{
private:
    int width,height;
public:
    void setWH(int w,int h){width=w;height=h;}
    int getWidth(){return width;}
    int getHeight(){return height;}
    int getGrith(){return (width+height)*2;}
    int getArea(){return width*height;}
};
void main()
{
    class Rectangle r1,r2;
```

```
r1.setWH(10,8);
cout<<"矩形 1:宽 = "<< r1.getWidth()<<" 高 = "<< r1.getHeight()<<endl;
cout<<"周长 = "<<r1.getGrith()<<endl;
cout<<"面积 = "<<r1.getArea()<<endl;
r2.setWH(30,20);
cout<<"矩形 2:宽 = "<<r2.getWidth()<<" 高 = "<< r2.getHeight()<<endl;
cout<<"周长 = "<<r2.getGrith()<<endl;
cout<<"面积 = "<<r2.getArea()<<endl;
```

运行结果:

```
矩形 1:宽 = 10 高 = 8
周长 = 36
面积 = 80
矩形 2:宽 = 30 高 = 20
周长 = 100
面积 = 600
```

程序首先说明一个矩形类 Rectangle,它是对所有矩形的抽象,类中包含两个数据成员 width,height 表示矩形的宽和高,成员函数 setWH(int w,int h)用于设置具体矩形的宽和高,成员函数 getWidth()、getHeight()用于获取当前矩形的宽和高,成员函数 getGrith()、getArea()用于计算矩形的周长和面积。从类的定义可以看出,数据成员表示事物的属性,成员函数表示事物的行为、动作或操作。

主函数中定义了类 Rectangle 的两个具体矩形 r1 和 r2,r1 和 r2 称为类 Rectangle 的实例或对象。主函数中通过调用成员函数输出两个矩形的宽、高、周长和面积。

面向对象方法编程重点就是要把待解决问题进行抽象,用类的形式表示出来。

在编写 C++ 程序时,要注意书写格式,尽量遵循以下基本原则:

(1) 一般情况下一行只写一条语句。短语句可以一行写多条,长语句可以分成多行来写。分行原则是不能将一个单词分开,用双引号括起来的字符串最好也不要分开,如果一定要分开,有的编译系统要求在行尾加上续行符“\”。

(2) C++ 程序书写时要尽量提高可读性。为此,采用适当地缩进格式书写程序是非常必要的,表示同一类内容或同一层次的语句要对齐。例如,一个循环的循环体中的各语句要对齐,同一个 if 语句中的 if 体内的若干条语句或 else 体内的若干条语句要对齐。

(3) C++ 程序中大括号“{}”使用较多,其书写方法也较多,建议用户要养成使用大括号“{}”的固定风格。例如,每个大括号占一行,并与使用大括号的语句对齐,大括号内的语句采用缩进 4 个字符的格式书写,如例 1.1 所示。

1.1.3 创建和运行 C++ 程序

C++ 源程序要经过编辑、编译、连接、运行 4 个环节,才能产生输出结果。例如,要编制一个名为 Ch1_01 的程序,其操作流程如图 1.1 所示。

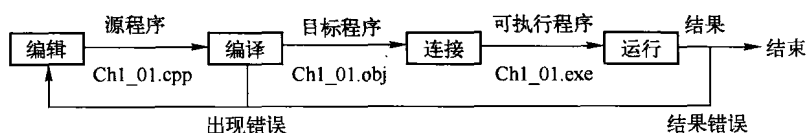


图 1.1 C++ 程序操作流程

1. 编辑

编辑是在一定的环境下进行程序的输入和修改过程。C++ 程序可以事先在纸上写好,也可以在编辑环境下直接输入到计算机中。用某种计算机程序设计语言编写的程序称为源程序,保存后生成程序文件,C++ 源程序默认文件扩展名为“.CPP”。C++ 源程序也可以使用计算机所提供的各种编辑器进行编辑。

2. 编译

编辑好的源程序不能直接被计算机所理解,源程序必须经过编译,生成计算机能够识别的机器代码。通过编译器将 C++ 源程序转换成二进制机器代码的过程称为编译,这些二进制机器代码称为目标代码。这些目标代码保存在以“.OBJ”为扩展名的目标文件中。

编译阶段要进行词法分析和语法分析,又称源程序分析。这一阶段主要是分析程序的语法结构,检查 C++ 源程序的语法错误。如果分析过程中发现有不符合要求的语法错误,就会及时报告给用户,将错误类型显示在屏幕上。

3. 连接

编译后生成的目标代码还不能直接在计算机上运行,其主要原因是编译器对每个源程序文件分别进行编译,如果一个程序有多个源程序文件,编译后这些源程序文件还分布在不同的地方,因此,需要把它们连接在一起,生成可以在计算机上运行的可执行文件。即使 C++ 源程序只有一个源文件,这个源文件生成的目标程序也还需要系统提供库文件中的一些代码,故也需要连接起来。

连接工作一般由编译系统中的连接程序来完成,连接程序将由编译器生成的目标代码文件和库中的某些文件连接在一起,生成一个可执行文件。可执行文件的默认扩展名为“.EXE”。

4. 运行

一个 C++ 源程序经过编译和连接后生成了可执行文件,就可以在 Windows 环境下直接双击该文件运行该程序,也可以在 Visual C++ 6.0 的集成开发环境下运行。

程序运行后,将在屏幕上显示运行结果或显示提示用户输入数据的信息。用户可以根据运行结果来判断程序是否有算法错误。在生成可执行文件之前,一定要保证编译和连接不出现错误和警告,这样才能正常运行。因为程序中有些警告虽然不影响生成可执行文件,但有可能导致结果错误。

下面以一个简单 C++ 程序设计为例来说明 C++ 程序的上机实现过程。

(1) 启动 Visual C++ 6.0。

用户可以单击“开始”菜单,在“程序”选项中选择“Microsoft Visual Studio 6.0”级联菜单下的“Microsoft Visual C++ 6.0”,或在桌面上双击“Microsoft Visual C++ 6.0”的快捷方式,就会启动 Visual C++ 6.0,打开如图 1.2 所示的集成开发环境。

(2) 在“文件”菜单下选择“新建”命令,将弹出“新建”对话框,如图 1.3 所示。

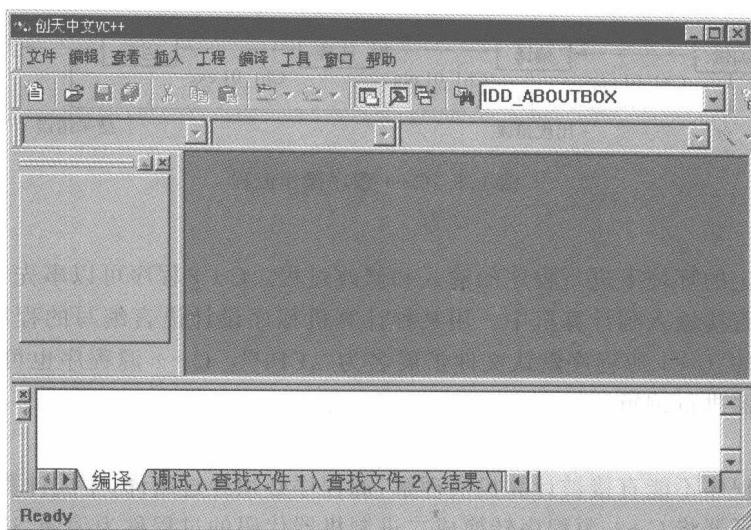


图 1.2 Visual C++ 6.0 的集成开发环境

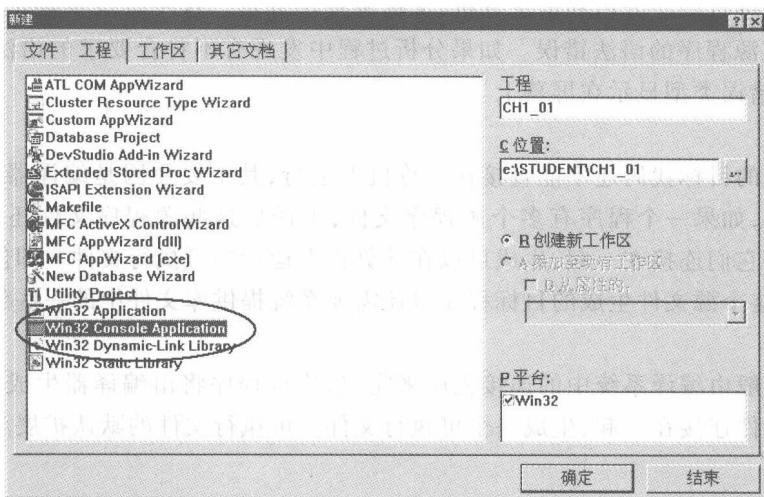


图 1.3 “新建”对话框

在该对话框中选择“工程”标签下的“Win32 Console Application”选项,可创建一个基于控制台应用程序的工程;在“工程”编辑栏内输入创建的工程名称(CH1_01);在“位置”编辑栏内输入工程的保存位置(E:\STUDENT),确保单选按钮“R 创建新工作区”被选定,其他选项选择默认值,单击“确定”按钮将弹出“Win32 Console Applition”项目步骤对话框,如图 1.4 所示。

(3) 在“Win32 Console Applition”项目步骤对话框中,选择要创建应用程序的类型,共有 4 个选项:“An empty project.”(一个空工程)、“A simple application.”(一个简单应用程序)、“A “Hello, World” application.”(一个“世界,你好”应用程序)和“An application that support MFC.”(一个支持 MFC 的应用程序)。不同的选项意味着系统自动生成的代码会有所不同。

此例中选择第 1 个选项,创建一个空工程。然后单击“完成”按钮,将出现“新建工程信息”对话框,如图 1.5 所示。该对话框给出了新创建工程的简单信息。

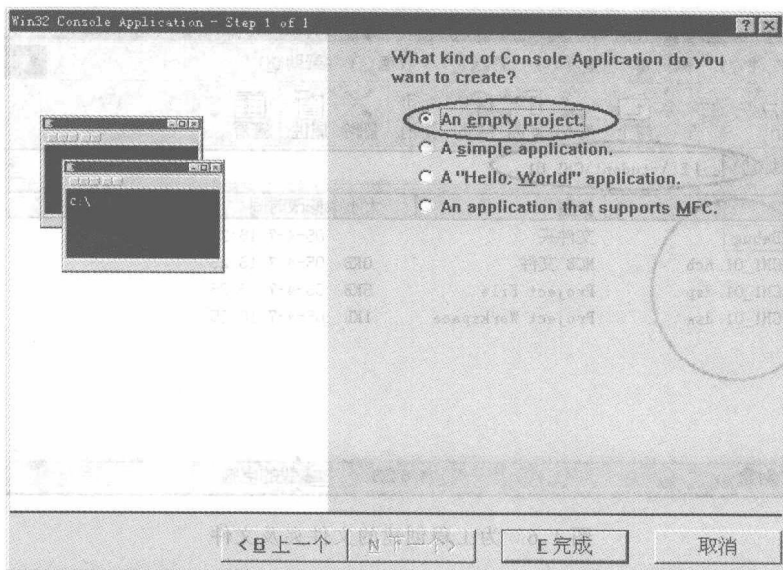


图 1.4 “Win32 Console Applition”项目步骤对话框

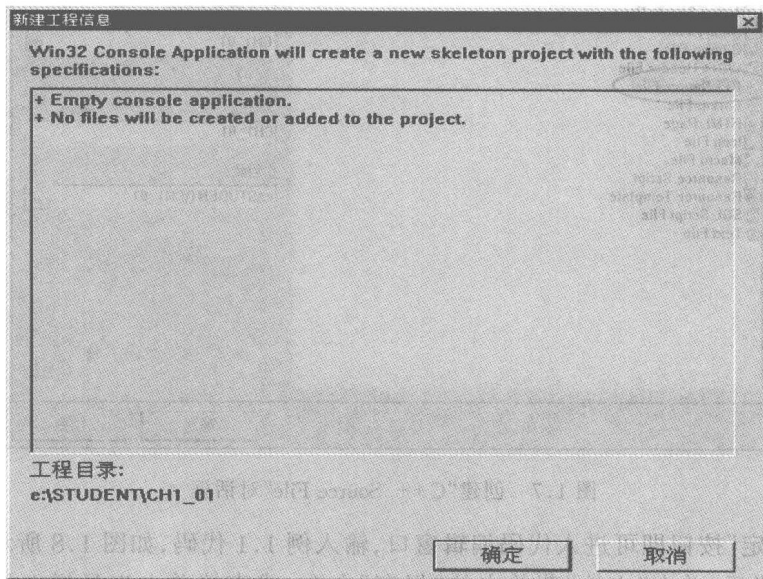


图 1.5 “新建工程信息”对话框

(4) 单击“确定”按钮,创建空工程的工作结束,回到 Visual C++ 6.0 的集成开发环境。

至此,系统为工程 CH1_01 自动创建了“E: \ STUDENT \ CH1_01”文件夹,并在其中生成了“CH1_01.dsp”、“CH1_01.dsw”、“CH1_01.ncb”文件和“Debug”文件夹,如图 1.6 所示。

(5) 在“文件”菜单下选择“新建”命令,在出现的对话框中选择“文件”标签,选择“C++ Source File”选项,创建一个 C++ 源程序。然后在“文件”编辑栏内输入创建的 C++ 源程序名(如 CH1_01);在“目录”编辑栏内输入 C++ 源程序的保存位置(如 e: \ STUDENT \ CH1_01)。如图 1.7 所示。

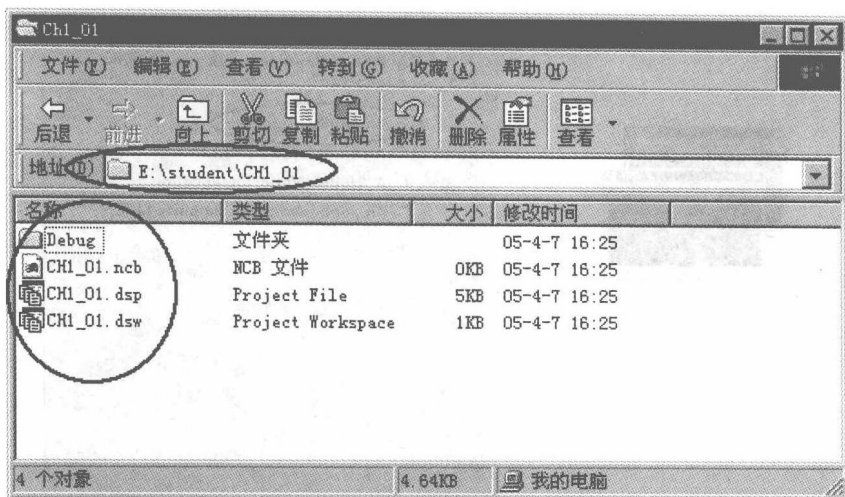


图 1.6 为工程创建的文件夹及文件

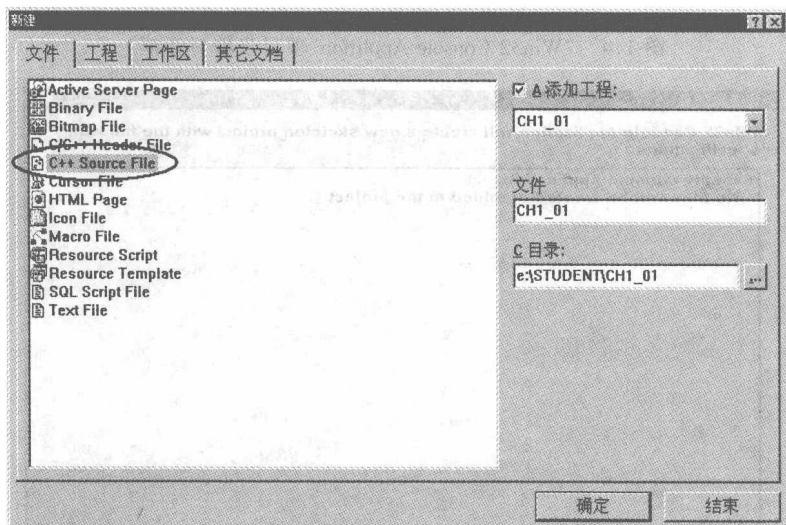


图 1.7 创建“C++ Source File”对话框

(6) 单击“确定”按钮即可进入代码编辑窗口,输入例 1.1 代码,如图 1.8 所示。

代码输入完毕后,单击“文件”菜单中的“保存”命令,或直接单击工具栏上的“保存”工具按钮,保存 C++ 源程序,文件名为“CH1_01.cpp”。

(7) 选择“编译”菜单下的“编译 CH1_01.cpp”项或单击微型编译工具栏中的“编译程序”工具按钮,对 C++ 源程序“CH1_01.cpp”进行编译。如果编译有错误,需要回到程序编辑窗口重新编辑,直到没有错误为止。此时将在信息输出窗口中显示“CH1_01.obj-0 error(s), 0 warning(s)”的信息。

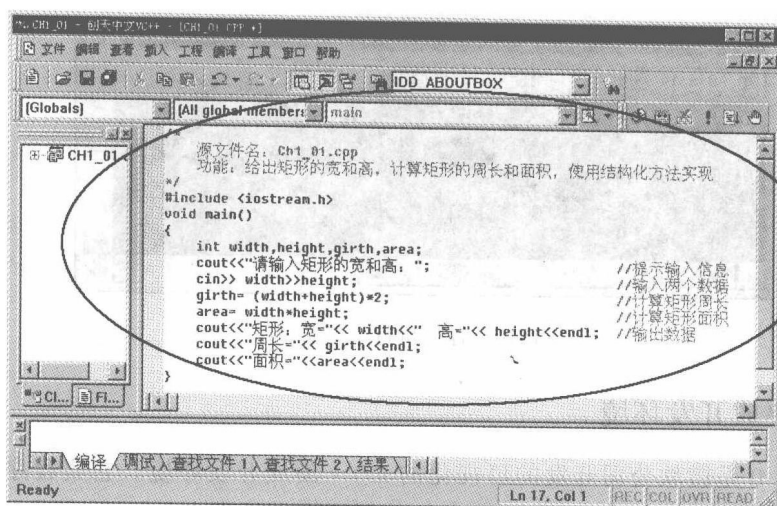


图 1.8 代码编辑窗口

(8) 编译完成后,选择命令:“编译/构建 CH1_01.exe”或单击微型编译工具栏中的“构建程序”工具按钮,如果没有连接错误,将生成可执行文件“CH1_01.exe”。结果如图 1.9 所示。

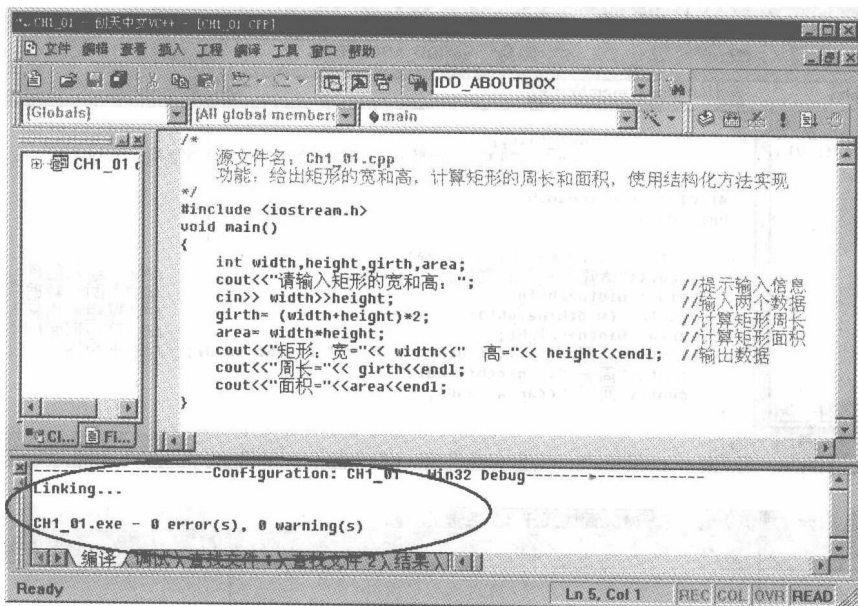


图 1.9 生成可执行文件界面

(9) 用户可以在 Windows 环境下直接用鼠标双击“CH1_01.exe”文件,或者单击微型编译工具栏中的“执行程序”工具按钮,都可以执行程序。

程序运行后,将弹出一个窗体要求用户输入两个整数,从键盘输入 60 和 40,中间用空格分隔,回车。屏幕显示程序的运行结果,如图 1.10 所示。