



高职高专“十一五”规划教材·计算机类

C++程序设计

刘秋菊 主编

C++ CHENGXUSHEJI



 北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

高职高专“十一五”规划教材·计算机类

C++程序设计

刘秋菊 赵淑梅 主编
张 昆 秦国防 副主编

 **北京理工大学出版社**

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书共分为 13 章,从 C++语言基础开始介绍,由于 C++不是纯面向对象的语言,因此第 1~7 章着重介绍 C++基本语法、函数、数组和指针等 C++语言基础;第 8~12 章详细介绍 C++面向对象的概念,包括类、对象、派生类、继承、多态性和流类库等。最后一章介绍了程序设计的完整思路。全书提供实例分析,将复杂的概念用简洁浅显的语言来描述,力求做到深入浅出,使读者深刻领会面向对象程序设计思想,掌握面向对象编程的方法和要领。本书以应用为目的,注重应用能力的培养,各章结尾均配有习题。

本书适合作为高职高专院校计算机及相关专业学生学习 C++程序设计课程的教材,也适合作为相关技术人员的自学参考书。

版权专有 侵权必究

图书在版编目(CIP)数据

C++程序设计/刘秋菊,赵淑梅主编. —北京:北京理工大学出版社, 2007.2

高职高专“十一五”规划教材.计算机类

ISBN 978-7-5640-0912-0

I.C… II.①刘… ②赵… III.C 语言-程序设计-高等学校:技术学校-教材 IV.TP312

中国版本图书馆 CIP 数据核字(2007)第 009829 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(总编室) 68944990(批销中心) 68911084(读者服务部)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 北京市业和印务有限公司

开 本 / 787 毫米×1092 毫米 1/16

印 张 / 16.75

字 数 / 320 千字

版 次 / 2007 年 2 月第 1 版 2007 年 2 月第 1 次印刷

责任校对 / 陈玉梅

定 价 / 28.00 元

责任印制 / 母长新

图书出现印装质量问题,本社负责调换

前 言

随着计算机科学的发展，计算机技术已渗透到各学科的研究和应用之中。通过多年教学实践证明，C++语言作为教学与实践应用，都是比较好的选择。本书的编写极力倡导“由实例引入新知识——介绍新知识——综合新知识的应用”的新写作思路。

虽然 C++语言是从 C 语言发展而来，但是与 C 语言相比有很大的不同。本书采用循序渐进的编写方式，由结构化程序设计逐渐过渡到面向对象的程序设计，将重点放在编写程序的基本思路和对基本知识的掌握上，力求做到简单明了、通俗易懂、例题精练，把抽象的理论变得更具体、更实际。相信读者在学习本书的过程中会将学习变为乐趣，逐渐步入佳境，真正体会面向对象程序设计思想的精髓。

本书是作者结合多年的 C++语言教学经验编写而成的。本书第 1 章~第 7 章是基础部分，主要介绍 C++程序设计语言的基本要素、程序的结构和过程化基础；第 8 章~第 12 章是 C++的面向对象程序设计部分，主要讲述了 C++中的类、对象、继承、派生、多态、虚函数和文件流等核心内容；最后一章介绍了程序设计的步骤、内容和思路等，使学生通过程序设计进一步理解编写程序的思路和方法。

本书中所有程序全部在 Visual C++ 6.0 下运行通过，并提供所有程序源代码及相关文档。

本书由刘秋菊、赵淑梅担任主编，张昆、秦国防担任副主编。参加本书编写的还有侯丽敏、罗涛、彭斌、王仲英。全书由刘秋菊统稿。

本书的编写得到了郝小会的大力支持和帮助，他为本书的成稿付出了辛勤的劳动，并提出诸多好的建议，在此一并致以诚挚的谢意。由于编者水平有限，书中难免出现差错、疏漏之处，敬请同行专家及广大读者批评指正。如读者在使用本书的过程中有其他意见或建议，恳请向编者(bjzhangxf@126.com)踊跃提出宝贵意见。

编 者

目 录

第1章 C++语言概述..... 1

1.1 C++语言及其发展 1

1.2 程序设计方法 1

1.2.1 面向过程程序设计方法 1

1.2.2 面向对象程序设计方法 2

1.3 C++程序的结构 3

1.3.1 简单的C++程序 3

1.3.2 C++程序的结构 4

1.4 Visual C++ 6.0 开发环境简介 7

1.4.1 Visual C++ 6.0 开发环境简介 7

1.4.2 C++语言程序的编写和运行 7

1.5 C++语言的特点 11

习 题 11

第2章 数据类型、运算符和表达式 13

2.1 数据类型 13

2.1.1 数据类型的划分 14

2.1.2 基本数据类型的存储 15

2.2 常量和变量 15

2.2.1 常量 16

2.2.2 变量 17

2.2.3 引用 18

2.2.4 符号常量与常值变量 19

2.3 数值表达式 20

2.3.1 无运算符的表达式 20

2.3.2 算术运算符 20

2.3.3 位运算符 20

2.3.4 赋值运算符 20

2.3.5 复合赋值运算符 21

2.3.6 增1减1运算符 21

2.3.7 求字节数运算符 22

2.3.8 运算符的优先级和结合性 22

2.3.9 类型的自动转换与强制转换 22

2.4 逻辑型数据和逻辑表达式 23

2.4.1 逻辑型数据的表示与存储 23

2.4.2 逻辑表达式 23

2.4.3 逻辑型与其他数据类型的 关系 24

2.4.4 条件运算符与条件表达式 24

2.5 自定义类型修饰符 25

2.6 逗号运算符与逗号表达式 25

2.7 实 训 26

2.7.1 实训1 26

2.7.2 实训2 27

习 题 27

第3章 C++基本语句 30

3.1 声明语句 30

3.2 输入/输出语句 30

3.2.1 输入语句 30

3.2.2 输出语句 31

3.2.3 常用的控制符 32

3.3 分支语句 33

3.3.1 if语句 33

3.3.2 switch语句 37

3.4 循环语句 39

3.4.1 while语句 40

3.4.2 do...while语句 41

3.4.3 for语句 43

3.4.4 循环语句的嵌套 45

3.5 转向语句 46

3.5.1 goto语句 46

3.5.2 break语句 47

3.5.3 continue语句 48

3.5.4 return语句 48

3.6 实 训 49

3.6.1 实训1 49

3.6.2 实训2 50

3.6.3 实训3 51

习 题.....	52	5.7.3 实训 3.....	92
第 4 章 数组.....	55	习 题.....	94
4.1 一维数组.....	55	第 6 章 指针.....	98
4.1.1 一维数组的定义及初始化.....	55	6.1 指针与指针变量的概念.....	98
4.1.2 数组元素的引用及基本操作.....	57	6.1.1 指针的概念.....	98
4.2 二维数组.....	59	6.1.2 指针变量的定义和初始化.....	98
4.2.1 二维数组的定义及初始化.....	59	6.1.3 指针的运算.....	100
4.2.2 数组元素的引用及基本操作.....	61	6.2 指针与数组.....	101
4.3 字符数组与字符串.....	61	6.2.1 指针与一维数组.....	101
4.3.1 字符数组的定义及初始化.....	61	6.2.2 指针与二维数组.....	103
4.3.2 字符串.....	62	6.2.3 动态数组与动态存储分配.....	106
4.3.3 字符串处理函数.....	63	6.2.4 指针数组.....	109
4.4 实 训.....	66	6.3 指针与字符串.....	110
4.4.1 实训 1.....	66	6.3.1 字符指针.....	110
4.4.2 实训 2.....	67	6.3.2 字符指针与字符数组.....	111
4.4.3 实训 3.....	69	6.4 指针与函数.....	112
习 题.....	70	6.4.1 指针参数.....	112
第 5 章 函数.....	73	6.4.2 返回指针值的函数.....	113
5.1 函数的定义和声明.....	73	6.4.3 指向函数的指针.....	114
5.1.1 函数的定义.....	73	6.4.4 作为参数传递的函数.....	115
5.1.2 函数的声明.....	75	6.5 实 训.....	116
5.2 函数的调用.....	76	6.5.1 实训 1.....	116
5.2.1 函数的调用格式及调用方式.....	76	6.5.2 实训 2.....	118
5.2.2 函数的递归调用.....	77	习 题.....	119
5.3 函数调用中的参数传递.....	79	第 7 章 结构与联合.....	121
5.3.1 传值参数.....	80	7.1 结构的定义.....	121
5.3.2 数组参数.....	80	7.2 结构变量的定义和初始化.....	122
5.3.3 引用参数.....	82	7.2.1 用结构类型名定义变量.....	122
5.3.4 可选参数.....	83	7.2.2 定义结构类型的同时定义 变量.....	122
5.4 内联函数.....	84	7.2.3 定义匿名结构类型的同时 定义变量.....	123
5.5 函数重载.....	84	7.3 结构成员的访问.....	124
5.6 函数和变量的作用域.....	86	7.4 结构与函数.....	125
5.6.1 函数的作用域.....	87	7.5 结构与指针.....	127
5.6.2 变量的作用域和生存期.....	87	7.6 联合.....	128
5.7 实 训.....	90	7.6.1 联合的定义和访问.....	129
5.7.1 实训 1.....	90		
5.7.2 实训 2.....	92		

7.6.2 联合的应用	130	9.4 虚基类	176
7.7 实训	131	9.4.1 虚基类的定义	176
7.7.1 实训 1	131	9.4.2 虚基类的初始化	178
7.7.2 实训 2	133	9.5 实训	179
习题	134	习题	182
第 8 章 类与对象	135	第 10 章 多态性	184
8.1 类与对象的定义	135	10.1 多态性	184
8.1.1 类的定义格式	136	10.1.1 多态的概念	184
8.1.2 类的定义和使用说明	136	10.1.2 编译时的多态性和运行时的多态性	184
8.1.3 对象的定义及访问	138	10.2 运算符重载	185
8.2 类的作用域	140	10.2.1 为什么要进行运算符重载	185
8.3 类的成员函数	142	10.2.2 运算符重载的规则	186
8.3.1 构造函数	142	10.2.3 运算符重载的方法	186
8.3.2 析构造函数	144	10.3 虚函数	191
8.4 友元函数	147	10.3.1 虚函数的定义	191
8.5 类的静态成员	148	10.3.2 虚函数与重载函数的比较	193
8.5.1 静态数据成员	148	10.4 抽象类与纯虚函数	194
8.5.2 静态成员函数	150	10.4.1 纯虚函数	194
8.6 实训	152	10.4.2 抽象类	194
8.6.1 实训 1	152	10.5 实训	196
8.6.2 实训 2	154	习题	198
习题	156	第 11 章 模板	201
第 9 章 继承与派生	158	11.1 函数模板的定义和使用	201
9.1 继承与派生类	158	11.1.1 函数模板的定义	201
9.1.1 派生类的定义	158	11.1.2 函数模板的使用	202
9.1.2 派生类的继承方式	161	11.2 类模板的定义和使用	204
9.2 派生类的构造函数和析构造函数	166	11.2.1 类模板的定义	204
9.2.1 派生类的构造函数和析构函数的执行顺序	166	11.2.2 类模板的使用	205
9.2.2 派生类的构造函数和析构函数的构造规则	168	11.3 实训	207
9.3 多重继承	171	11.3.1 实训 1	207
9.3.1 多重继承的概念	171	11.3.2 实训 2	208
9.3.2 多重继承的声明	171	习题	209
9.3.3 多重继承的构造函数和析构造函数	172	第 12 章 C++输入输出流	212
9.3.4 多重继承中的同名问题	174	12.1 C++流	212
		12.2 格式控制	214
		12.2.1 用 ios 类成员函数进行格式	

控制	215	习 题	231
12.2.2 ios 类中的其他成员函数	216	第 13 章 C++程序设计实训	233
12.2.3 格式控制操作符	220	13.1 实训 1 电话簿管理程序	233
12.3 文件的 I/O 操作	221	13.2 实训 2 模拟网上购书的结账功能	241
12.3.1 文件与文件流	221	附录	252
12.3.2 文件流对象的建立	222	附录 1 C++运算符的优先级和结合性	252
12.3.3 文件的打开与关闭	222	附录 2 常用字符与 ASCII 码对照表	253
12.3.4 文本文件的 I/O 操作	224	附录 3 2006 年全国计算机等级考试	
12.3.5 二进制文件的 I/O 操作	225	二级 C++考试大纲	254
12.4 实 训	227	参考文献	258
12.4.1 实训 1	227		
12.4.2 实训 2	228		

第 1 章 C++语言概述

1.1 C++语言及其发展

C++语言是从 C 语言发展而来的，是 C 语言的扩展。最初的 C 语言是贝尔实验室为了描述和实现 UNIX 操作系统而设计的。后来 C 语言不断改进，由于其具有功能丰富、表达力强、使用灵活、可移植性强等特点，引起了人们的关注，使 C 语言从实验室走向了社会。

C++是由贝尔实验室在 C 语言的基础上开发成功的。它既保留了 C 语言原有的精华，又增加了面向对象的机制，使程序的结构更加清晰、更容易维护和扩充，同时又不丧失其高效性。C++支持面向对象的概念，如对象、类、属性和方法、派生类与继承等，因此 C++又叫做“带类的 C”，成为继 C 语言之后另一个令人不可忽视的“风云人物”。

1.2 程序设计方法

1.2.1 面向过程程序设计方法

程序是为解决某一问题而编写的语句的集合，编写程序的过程就是程序设计。概括地说，程序设计就是分析问题、编写程序和调试程序的过程。在最初的程序设计步骤中，人们把解决问题的过程看成是加工数据的过程，基于这种方法的程序设计称为面向过程程序设计。

面向过程的程序设计者认为，每个程序都要完成一些规定的功能，每个功能的实现是通过对数据进行一系列的加工而实现的。因此，程序设计包括设计数据结构及设计算法两部分。

在进行算法和程序设计时，如果不加规范，可能会使程序执行转来转去，从而造成程序结构的混乱，给阅读、修改和调试程序造成困难。为了克服程序设计的随意性，迪杰斯特拉(Dijkstra)提出了结构化程序结构的思想、概念和标准，希望通过程序静态结构的良好性来保证程序的正确性；同时，Wirth 也提出了自顶向下、逐步求精的程序设计方法；面向过程程序设计的基本程序结构有 3 种：顺序结构、选择结构、循环结构。面向过程程序设计，将程序定义为：

数据结构+算法

两者分开设计，以算法为主。

1.2.2 面向对象程序设计方法

面向过程的程序设计将数据和操作分成两个相互独立的实体，用户必须时刻注意所处理的数据，对不同数据采用相同的处理或对相同的数据采取不同的处理，都要重新编程，程序的重用性不好。基于可重用性的思想是指建立一些具有已知特性的部件，在需要的时候可以插入到程序中，这是一种模仿硬件组合方式的做法。当工程师需要一个新的晶体管时，他不用自己发明，只要到仓库去找即可。用面向对象程序设计(OOP)方法解决问题就是直接考察与研究问题域，从中抽象出所关心的对象的本质特征，将它们自然地表达成软件系统中的对象。对象代表着待处理问题中的一个实体，这些实体之间的关系也通过软件系统中的对象关系自然地表达出来，从而以自然的方式表示问题域，完成问题解。从概念上讲，一个对象是包含数据，又包含处理这些数据的功能单元，若干个对象和它们之间的相互关系形成了程序，完成了问题解。因此，面向对象程序设计把程序定义为：

对象+消息

对象是数据和操作的封装体。面向对象的程序设计方法吸取了人们日常生活中进行逻辑思维所采取的封装和数据隐藏、继承、多态性技术。

1. 封装和数据隐藏

封装是指把数据和实现操作的代码集中起来放在对象内部，并尽可能隐蔽对象的内部细节。对象好像是一个不透明的黑盒子，表示对象属性的数据和实现各个操作的代码都被封装在黑盒子里，从外面是看不见的，更不能从外面直接访问或修改这些数据及代码。使用一个对象时，只需知道它向外界提供的接口形式，而无需知道它的数据结构细节和实现操作的算法。例如，当一个技术员要安装一台电脑时，他将各个部件组装起来，当他想要一个声卡时，不需要用原始的集成电路芯片和材料去制成一个声卡，而是购买一个他所需要的某种功能的声卡。技术员关心的是声卡的功能，并不是声卡内部的工作原理。声卡是自成一体的。这种自成一体性称为封装性。无需知道封装单元内部是如何工作就能使用的思想称为数据隐藏。声卡的所有属性都封装在声卡中，不会扩展到声卡之外。因为声卡的数据隐藏在其电路板上。

2. 继承和重用

继承在现实生活中是一个很容易理解的概念。例如，每一个人都从父母身上继承了一些特性，如种族、血型、肤色等，这种身上的特性来自父母，也就是说，父母是我们所具有的属性的基础。面向对象程序设计提供继承的作用有两个：一是避免公用代码的重复开

发,减少代码和数据冗余,增加程序的可重用性;二是通过增强一致性减少模块间的接口和界面。例如,要设计新的电视机,有两种选择:一是从设计草图开始,另一种是对现有型号加以改进。也许现有的型号已经令人满意,但如果再增加一个功能,会更加完美。设计电视机的工程师是希望在原有型号的基础上增加一组电路做成。这就是继承和重用的思想。

3. 多态性

多态性也是面向对象系统的重要特性。简单地说,多态性就是一个接口,多种方法。即对于相同的操作产生不同的行为,这种性质称为多态性。例如,可以声明一个基类“学生”,该类中定义了一个计算学习成绩的操作,并定义该类的派生类“大学生”和“中学生”,这些类都继承了基类中的计算学习成绩的操作。对“中学生”类,计算学习成绩的操作表示语文、数学、英语等课程的计算,而对于“大学生”类,计算学习成绩的操作表示高等数学、计算机、普通物理等课程的计算。

1.3 C++程序的结构

1.3.1 简单的C++程序

【例 1.1】 简单的C++程序

```
#include<iostream.h>
void main()
{
    cout<<"welcome to C++ class!"<<endl;
    //显示输出字符串"welcome to C++ class!"
}
```

程序的输出结果是:

```
welcome to C++ class!
```

上述程序中#include<iostream.h>是一条预处理命令,这条命令在编译之前预处理阶段执行;void main()指定了程序的主函数,函数名是main,函数名后一定要有一对小括号,小括号是函数的标志。void表示函数没有返回值,在程序的最后不需要用return语句;花括号内的内容称为函数体;以//开始的部分称为“注释语句”,注释语句一般用来解释和说明程序语句的功能;cout<<"welcome to C++ class!"<<endl;是C++程序中常用的输出语句。

1.3.2 C++程序的结构

由【例 1.1】可知, 一个 C++ 程序通常由函数、语句、输入/输出和预处理命令这几个基本部分组成。

1. 函数

函数是 C++ 程序最基本的程序框架, 可以说 C++ 是由许许多多的“函数”所组成, 即使是最简单的程序至少也会包含一个函数——main 函数(主函数)。每个 C++ 的主要程序代码都必须放在 main 函数里, 其他函数都是通过主函数直接或间接调用来执行的。如果程序没有 main 函数, 即使有再多的方法, 程序也无法执行。main 函数就像 C++ 程序的心脏, 掌握着程序能否执行的命脉。

函数必须有自己的名称加以识别, 但是 main 函数却是不能更改名称的。因为编译器在编译程序时, 会从 main 函数的位置开始编译, 即 main 函数是整个程序的入口, 如果没有这个函数, 就无法完成编译工作。一般函数的结构如下:

类型修饰符 函数名(形式参数表){函数体}

其中类型修饰符表示函数的返回值类型, 函数体必须包括在左花括号“{”和右花括号“}”中, 左花括号表示函数的起始位置, 右花括号表示函数的结束位置; 系统将通过函数名调用该函数。例如, 主函数的结构如下:

```
void main()    //void 表示该函数无返回值, 形式参数表为空
{ //函数的起始位置
    ...
    程序语句;
    ...
} //函数的结束位置
```

函数的写法有多种, 详见后面的章节。

2. 语句

语句是函数的基本组成单位, C++ 中语句可以分为基本语句、复合语句、表达式语句和循环语句等多种。

基本语句以分号“;”作为结束标记, 复合语句则是使用花括号“{}”括起来的语句序列。

在 C++ 程序中, 字母的大小写是有区别的, Main 与 main 是不同的名称。

3. 输入/输出

C++提供了专门用于处理输入/输出的对象和操作符，并且可以使用格式控制符或调用输入/输出对象的成员函数来控制输入/输出的格式。

C++的输入流对象是 `cin`。`cin` 就是读取用户由键盘输入的数据，其语法如下：

```
cin>>变量1>>变量2>>...>>变量n;
```

输入流对象 `cin` 将会读取键盘的数据，直到用户按 `Enter` 键为止。“>>”是与之相联系的输入操作符。如果输入的数据包含两个以上的数据，则以空格键作为间隔符号。

C++的输出流对象是 `cout`，`cout` 就是将结果显示在屏幕上，其语法如下：

```
cout<<表达式1<<表达式2<<...表达式n;
```

`cout` 将会按“<<”的顺序，将表达式的值显示在屏幕上。“<<”是与之相联系的输出操作符。可以用 `endl` 实现换行。

例如，执行语句

```
cout<<35<<endl<<"hello world"<<endl;
```

的输出结果是：

```
35
```

```
hello world
```

4. 预处理命令

预处理命令不是 C++语言的一部分，它只是用来扩充 C++程序设计的环境。预处理命令有 3 种。

(1) `#include` 命令

`#include` 命令也称文件包含命令，其作用是将指定的文本文件引入到程序该点处，该文本文件包含了许多函数的定义。例如之前提到的 `cin` 或 `cout` 的定义，就在 `iostream.h` 头文件中，所以在使用输入/输出流对象 `cin` 或 `cout` 之前，必须先引入 `#include<iostream.h>` 或 `#include "iostream.h"`。

(2) `#define` 命令

`#define` 命令用来定义一个符号常量。例如：

```
#define PI 3.1415
```

(3) 条件编译命令

使用条件编译命令可以限定程序中的某些内容要在满足一定条件的情况下参与编译。因此，利用条件编译可以使同一个源程序在不同的编译条件下产生不同的目标代码。常用

的条件编译命令格式如下。

格式 1:

```
#ifdef 标识符
```

```
    程序段 1
```

```
[#else
```

```
    程序段 2]
```

```
#endif
```

功能是如果“标识符”已经定义过，则编译程序段 1，否则编译程序段 2；如果没有程序段 2，则`#else`可以省略。

格式 2:

```
#ifndef 标识符
```

```
    程序段 1
```

```
[#else
```

```
    程序段 2]
```

```
#endif
```

功能是如果“标识符”未被定义，则编译程序段 1，否则编译程序段 2；如果没有程序段 2，则`#else`可以省略。

5. 程序注释

程序的注释在程序的编写过程中是非常重要的。一个没有程序注释的程序就像一个没有使用说明书的软件。

C++的程序注释方式如下:

```
//注释的内容
```

或

```
/*注释的内容*/
```

注释以`//`开头，其后是注释文字，可一直延续到该行行尾。`/*注释的内容*/`是另一种方式，注释文字夹在`/*`和`*/`之间，这样的注释不但可以出现在行尾，也可以出现在一行中的其他位置，还可以跨越多行。

编译系统不会编译注释文字，因此注释文字可以是任意的。注释可以使程序更易于理解，不会有其他副作用，因此在编写程序时随时添加注释是一种良好的习惯。

1.4 Visual C++ 6.0 开发环境简介

1.4.1 Visual C++ 6.0 开发环境简介

Visual C++ 6.0 是美国 Microsoft 公司研制开发的,它是一个集C++程序的编辑、编译、调试、运行和在线帮助等功能及可视化软件开发功能为一体的软件开发工具,或称开发环境、开发系统等。本节对其作简单介绍,目的是让读者掌握编辑、编译和运行一个C++控制台应用程序(Console Application Program)的简要过程。

在 Visual C++ 6.0 集成开发环境中,要建立一个C++应用程序(以后简称C++程序),首先要建立一个工程(有书称为项目或 Project),就是在计算机的磁盘上建立一个表示工程的专用目录,然后把C++程序作为一个或若干个文件保存到这个目录中,再通过编译、连接和运行等步骤实现程序所具有的功能。

要用C++语言编写出一个完整的程序,第一步需要上机建立相应工程;第二步输入和编辑该工程所含的一个或多个程序文件以及一些头文件,其中有一个程序文件必含有main 函数,称为程序主文件,简称主文件,它通常被首先输入和编辑;第三步对每个程序文件进行编译,生成各自的目标代码(即二进制代码)文件,通常首先将主文件编译并生成主目标文件(注意用户建立的头文件不需要编译);第四步使主文件与同一程序中的其他目标代码文件以及有关C++库函数文件相连接,生成一个可执行文件;第五步运行最后生成的文件,实现用户编程所要求的对数据的计算或处理功能。

当由一个程序文件编译生成一个目标文件时,目标文件的主文件名与程序文件的主文件名相同,而扩展名改为.obj,当把程序文件中所有的目标文件连接生成一个可执行文件时,该文件的主文件名采用工程名,扩展名为.exe。

用户在第一次上机使用 Visual C++ 6.0 集成开发环境时,若在计算机上还没有安装 Visual C++ 6.0,则应设法安装然后才能使用它。

若在 Windows 操作系统桌面上含有以 Microsoft Visual C++ 6.0 名字的图标,双击该图标,在屏幕上就打开集成开发环境操作界面窗口。否则,应单击屏幕左下角的“开始”按钮,选择“程序”子菜单的 Microsoft Visual C++6.0,即可运行该程序,进入了 Visual C++ 6.0 集成开发环境。

1.4.2 C++语言程序的编写和运行

1. 编辑 C++程序

启动 Visual C++ 6.0 后,选择“文件”(File)|“新建”命令,得到如图 1.1 所示的“新

建”对话框。

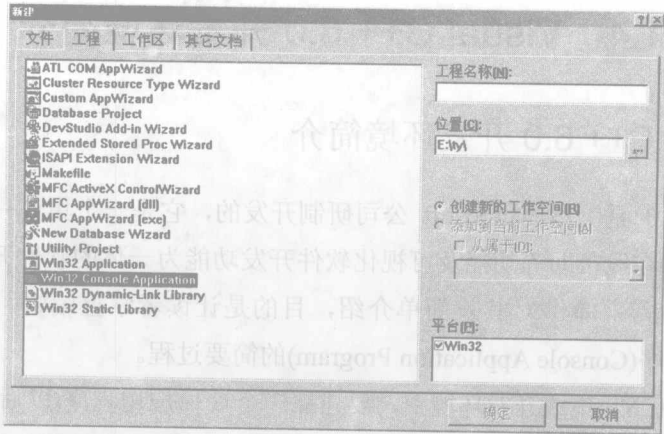


图 1.1 “新建”对话框

从打开的“新建”对话框中打开“工程”选项卡，从该选项卡中选择 Win32 Console Application 选项，准备建立应用程序的工程目录。在右边的文本“位置”文本框中输入目录或通过该文本框右边的按钮选择一个待建立工程目录的父目录，假定选定 E:\ty，再在其上面的“工程”文本框中输入一个标识符作为工程目录名，如“proj1”。工程名输入完成后单击“确定”按钮，在屏幕上显示如图 1.2 所示的对话框。

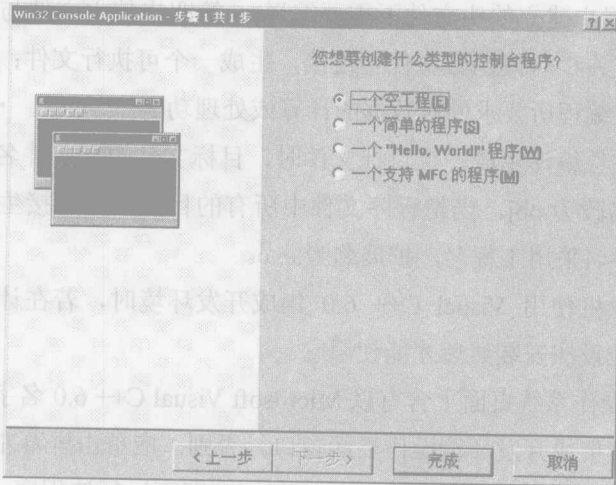


图 1.2 建立控制台应用程序的第一步

假定就是要建立一个初始为空的工程，用于保存用户程序，则单击“完成”按钮，打开建立程序的第二步对话框，单击对话框中的“确定”按钮后，立即回到 Visual C++集成开发环境界面中。通过以上步骤系统自动在 E:\ty 下建立了一个 proj1 子目录，用它作为待建程序的工程目录，在此目录下系统自动建立了该工程的工程文件(proj1.dsp)和工程工作

区文件(proj1.dsw)。

回到 Visual C++集成开发操作界面后,为了建立工程(即整个程序)中的每个程序文件和头文件,选择“文件”|“新建”命令则打开“新建”对话框,再在该对话框中切换到“文件”选项卡,如图 1.3 所示。

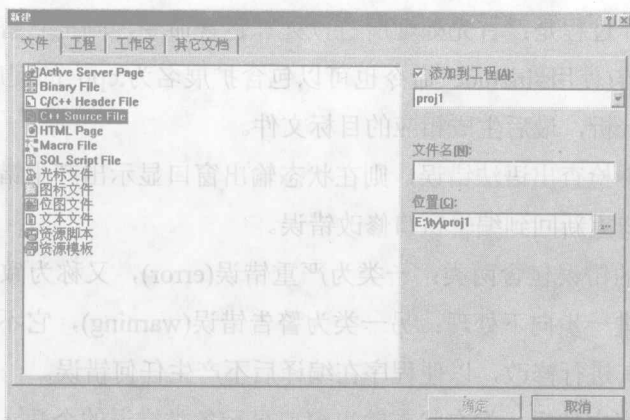


图 1.3 “文件”选项卡

在“文件”选项卡中若选择 C/C++ Header File 选项,则可新建一个 C++头文件;若选择 C++ Source File 选项则可新建一个 C++源程序文件,简称程序文件。假定选择 C++ Source File 选项则建立一个 C++程序文件,接着在右边的文件文本框中输入新建文件的文件名,默认扩展名为.cpp,假定输入的程序文件名为 exp8_1,最后单击右下角的“确定”按钮则关闭该对话框,回到 Visual C++集成开发环境界面。此时系统就在 E:\ty\proj1 目录下建立了一个空的 exp8_1.cpp 文件,待用户通过 Visual C++集成开发环境中的程序编辑窗口输入和编辑该程序文件的内容。编辑程序的操作界面如图 1.4 所示。

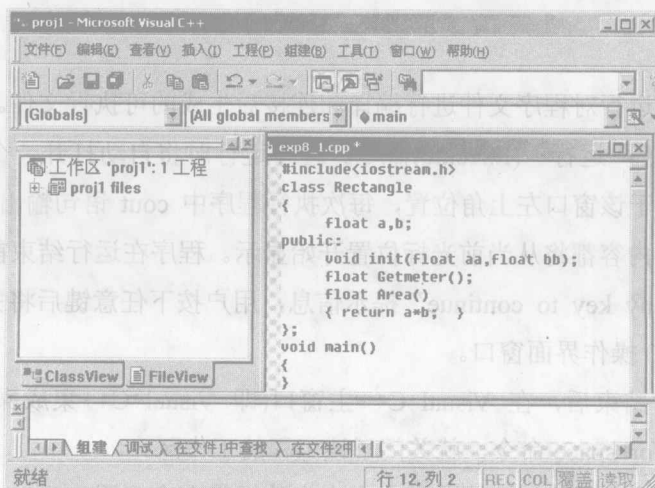


图 1.4 程序的编辑窗口