



高等学校**应用型特色**规划教材

C++

程序设计



肖守柏 金欢 编著

赠送
电子教案



清华大学出版社

高等学校应用型特色规划教材

C++程序设计

肖守柏 金 欢 编著

清华大学出版社
北 京

内 容 简 介

C++是近年来国内外广泛学习和使用的现代计算机语言,但是C++牵涉的概念很多,语法比较复杂,内容十分广泛,使不少人感到学习难度较大,难以入门。本书做到准确定位,合理取舍内容,设计了读者易于学习的教材体系,并且以通俗易懂的语言讲解了许多复杂的概念,减少了初学者学习C++的困难。

全书分为三部分,共9章,依次介绍了C++面向对象程序设计的基本概念、C++对C的扩充、类和对象、特殊函数和成员、运算符重载、继承与派生、多态性与虚函数等。每个章节基本上都是从实例入手,然后系统地介绍本实例所涉及的知识,整本书从前至后注重应用性,由浅入深、逻辑性强,并且每章末尾均配有小结和练习。

本书可作为高等院校计算机专业本、专科学生的教材或参考书,也可供广大从事计算机软件工作的科技人员自学参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C++程序设计/肖守柏等编著. —北京:清华大学出版社,2014

高等学校应用型特色规划教材

ISBN 978-7-302-34044-7

I. ①C… II. ①肖… 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(2013)第234292号

责任编辑:张彦青 桑任松

封面设计:杨玉兰

责任校对:宋延清

责任印制:杨 艳

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>, 010-62791865

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:15.5 字 数:371千字

版 次:2014年1月第1版 印 次:2014年1月第1次印刷

印 数:1~3000

定 价:32.00元

产品编号:053279-01

前 言

面向对象程序设计作为一种主流的程序设计思想和方法,能够更好地对现实世界中的各种数据、概念、特征及相互联系进行真实的建模和抽象,使得程序设计与实体行为能够更加接近。此外,基于面向对象程序设计的思想和方法,能够更好地组织和管理大型程序项目,有利于继承和发扬程序设计领域中各种杰出的智慧及思想。

C++语言是当前最流行的一种高级程序设计语言,它完全兼容 C 语言,既支持结构化的程序设计方法,也支持面向对象的程序设计方法。与其他程序设计语言相比,C++语言在运行效率、语法及语义、组件及类库、代码与资源等方面都有着显著的优越性。因此,学好 C++,很容易在一个较高平台上架设强大、易用的应用软件。

案例教学是计算机课程教学最有效的方法之一。本书围绕教学内容组织案例,对学生的知识和能力训练具有较强的针对性。全书共分为 9 章,可看成三部分:第一部分(第 1、2 章),介绍了面向对象程序设计基础知识,重点讨论了面向对象程序设计的基本概念和特点,以及 C++对 C 的扩充;第二部分(第 3~5 章),介绍了基于对象的程序设计,重点讨论了类和对象、特殊函数和成员、运算符重载;第三部分(第 6~9 章),介绍了面向对象程序设计,重点讨论了继承与派生、多态性与虚函数、C++的 I/O 流、异常处理。此外,书中有两个附录,分别是“关键字”和“实验指导”。书中的所有程序都在 VC++ 6.0 环境下调试通过。

C++面向对象程序设计是实践性很强的课程,本书注重理论与实践相结合,每章都给出了不同层次、不同难度的思考题。通过习题与实训,使学生掌握所学知识,并能灵活运用所学知识解决实际问题。教师可根据学时数、专业和学生的实际情况适当取舍,部分内容可安排自学。

本书由肖守柏、金欢、万磊、刘燕编著。其中,肖守柏编写了第 2、5、7 章以及附录 B,金欢编写了第 1、4、6、8 章以及附录 A,万磊编写了第 3 章、刘燕编写了第 9 章。全书由肖守柏负责统稿。

本书适合作为本科、高等职业学校、高等专科学校、成人院校、民办高校计算机及相关专业的教材,也可供广大从事计算机软件工作的科技人员自学参考。

由于编者水平有限,书中难免存在不足之处,敬请广大同行和读者批评指正。

编 者

目 录

第一部分 面向对象程序设计基础知识

第 1 章 初识 C++	1	2.2.1 函数的相关知识	23
1.1 面向对象概述	1	2.2.2 内联函数	28
1.1.1 什么是面向对象程序设计	1	2.2.3 函数重载	29
1.1.2 面向对象程序设计的特点	3	2.2.4 函数模板	30
1.2 “This is a C++ program.” 实例	4	2.2.5 带默认参数的函数	31
1.2.1 C++程序的基本组成	8	2.3 “棋盘游戏”实例	33
1.2.2 C++程序的编写和实现	10	2.3.1 引用的基本概念	34
本章小结	11	2.3.2 引用作为函数参数	36
本章习题	11	2.3.3 引用作为函数返回值	37
第 2 章 C++对 C 的扩充	13	2.4 “日期格式的转换”实例	38
2.1 “书店库存清单管理”实例	13	2.4.1 字符串类型相关知识	40
2.1.1 C++程序的输入与输出	14	2.4.2 使用字符串类型对象	42
2.1.2 用 const 定义常变量	20	本章小结	44
2.2 “学生成绩管理”实例	21	本章习题	44

第二部分 基于对象的程序设计

第 3 章 类和对象	46	第 4 章 特殊函数和成员	81
3.1 “两线相交”实例	46	4.1 “网络登录计数器”实例	81
3.1.1 类的声明	49	4.1.1 不同对象间数据的共享	83
3.1.2 对象的使用	54	4.1.2 静态数据成员	84
3.1.3 this 指针	58	4.1.3 静态成员函数	86
3.2 “个人工资所得税的计算”实例	58	4.2 “家庭防盗警报系统”实例	90
3.2.1 构造函数	61	4.2.1 友元的基本概念	92
3.2.2 析构造函数	67	4.2.2 友元函数	93
3.2.3 类的组合	73	4.2.3 友元类	95
本章小结	78	4.3 关键字 const	96
本章习题	79	4.3.1 常对象	97



4.3.2 常对象成员.....	97	5.1.1 运算符重载的基本概念	113
4.3.3 使用 const 限定指针	100	5.1.2 运算符重载的方法	114
4.4 “单链表的类模板”实例.....	102	5.1.3 运算符重载的规则	115
4.4.1 类模板的定义.....	105	5.2 “重载加、减法运算符”实例.....	116
4.4.2 类模板的实例化.....	106	5.2.1 类成员函数重载运算符	118
4.4.3 类模板的模板参数表.....	107	5.2.2 友元函数重载运算符	122
本章小结.....	108	5.2.3 转换运算符重载	125
本章习题.....	109	本章小结.....	127
第5章 运算符重载.....	111	本章习题.....	128
5.1 “复数相加”实例.....	111		

第三部分 面向对象程序设计

第6章 继承与派生.....	129	7.2 “学生学费管理”实例.....	171
6.1 “半工半读学生信息管理”实例.....	129	7.2.1 虚函数	174
6.1.1 继承与派生的基本概念.....	133	7.2.2 虚析构函数	178
6.1.2 派生类成员的访问属性.....	135	7.3 “小型公司人员信息管理”实例.....	180
6.1.3 多层继承的访问属性.....	142	7.3.1 纯虚函数	184
6.1.4 赋值兼容.....	143	7.3.2 抽象类	186
6.2 “员工信息管理”实例.....	145	本章小结.....	188
6.2.1 简单派生类的构造函数.....	149	本章习题.....	188
6.2.2 有子对象的派生类的构造函数.....	150	第8章 C++的 I/O 流	190
6.2.3 多层派生时的构造函数.....	153	8.1 流的概念.....	190
6.2.4 派生类的析构函数.....	154	8.1.1 流类	191
6.3 “动物园信息管理”实例.....	155	8.1.2 I/O 流对象	192
6.3.1 多重继承.....	159	8.2 标准 I/O 流	192
6.3.2 二义性.....	161	8.2.1 输入成员函数	192
6.3.3 虚基类.....	163	8.2.2 输出成员函数	194
本章小结.....	168	8.3 “文件中建立平方根表”实例.....	195
本章习题.....	168	8.3.1 文件的基本概念	196
第7章 多态性与虚函数	170	8.3.2 文件的打开与关闭	198
7.1 多态性的基本概念.....	170	8.3.3 文本文件的输入/输出	200
7.1.1 多态的概念和类型.....	170	8.3.4 二进制文件的输入/输出	204
7.1.2 多态的实现方式.....	171	本章小结.....	206
		本章习题.....	206

第 9 章 异常处理	208	9.2.4 异常处理的执行过程	213
9.1 异常概述	208	本章小结	216
9.1.1 异常处理的概念	208	本章习题	216
9.1.2 异常现象的举例	209	附录 A 关键字	217
9.2 异常处理机制	209	附录 B 实验指导	218
9.2.1 异常处理的任务	210	参考文献	237
9.2.2 异常处理基本思想	210		
9.2.3 异常处理的组成	211		

第一部分

面向对象程序设计基础知识

第 1 章 初识 C++

【教学目标与要求】

本章首先介绍面向对象程序设计的基本概念和特点，然后通过一个简单的 C++ 实例，介绍 Visual C++ 6.0 开发环境的使用，并对 C++ 程序的基本组成及开发过程进行介绍。通过本章的学习，读者应该掌握以下内容：

- 了解面向对象程序设计的基本概念及特点。
- 理解面向对象程序设计与面向过程程序设计的区别。
- 熟悉 C++ 程序的基本组成。
- 熟悉 Visual C++ 6.0 开发环境。
- 掌握 C++ 程序的开发过程。

【教学重点与难点】

面向对象程序设计的特点；Visual C++ 6.0 开发环境的使用；C++ 程序的基本组成。

1.1 面向对象概述

面向对象程序设计的基本思想是：首先将数据和对数据的操作方法集中存放在一个整体中，形成一个相互依存、不可分割的整体，这个整体即为对象。通过相同类型的对象，抽象出其共性而形成类。然后，类再通过外部接口与外界发生联系，对象与对象之间用消息进行通信。

由于只是通过外部接口进行联系，所以可以使程序模块间的关系变得更简单；而且程序模块间的相对独立性高，数据的安全也能得到很好的保护。面向对象引入了继承、多态等高级特性，使软件的可重用性、软件的可维护性都得到了更大的提高。

1.1.1 什么是面向对象程序设计

面向对象，即把数据及对数据的操作方法放在一起，作为一个相互依存的整体——对象。对同类对象抽象出其共性，形成类。类中的大多数数据，只能用本类的方法进行处理。类通过一个简单的外部接口与外界发生关系，对象与对象之间通过消息进行通信。程序流



程由用户在使用中决定。

面向过程，即自顶向下顺序执行，逐步求精，其程序结构是按功能划分为若干个基本模块，这些模块形成一个树状结构。各模块之间的关系尽可能简单，在功能上相对独立；每一模块内部均是由顺序、选择和循环三种基本结构组成的，其模块化实现的具体方法是使用子程序。程序流程在写程序时就已确定。

例如，教室里有 50 名学生排成一队，要从中找出 C 语言程序设计成绩较好的学生。若是面向对象，可以随意提问，让学习过 C 语言程序设计的留下，或分数在 80 分以上的留下，这样可在 50 个学生中找到符合条件的学生；若是面向过程，就必须一个一个地问：“你学习过 C 语言程序设计吗，你的考试成绩是多少”，中间一个学生都不能跳过。

面向对象会涉及许多新的概念，下面将介绍基本概念。

1. 对象

从一般意义的角度上讲，对象是现实世界中真实存在的事物，包括一切有形的和无形的事物。比如，一本书、一种思想等都是对象。对象是世界中一个独立的单位，都有自己的特征，包括静态特征和动态特征。对象的静态特征可以用某种数据来描述，而动态特征表现为其所表现的行为或具有的功能。比如，一台录像机是一个对象，它的静态特征是生产厂家、牌子、重量、体积、颜色、价格等，这种静态特征称为属性；其动态特征是它的行为，根据外界给它的信息进行录像、放像、快进、倒退、暂停、停止等操作，这种动态特征称为行为。

对象是描述世界事物的一个实体，是构成程序的一个基本单位。对象由一组属性(数据)和一组行为(函数)构成。属性用来描述对象的静态特征，行为用来描述对象的动态特征。

2. 类

抽象和分类是面向对象程序设计的两个原则。抽象是具体事物描述的一个概括，与具体是相对应的；而分类的依据原则是抽象。在对事物分类的过程中，忽略事物的非本质特征，只关注与当前对象有关的本质特征，从而提取事物的共性，把具有相同特性的事物划为一类，得出一个抽象的概念。比如，我们常用的名词“人”，就是一种抽象。因为世界上只有具体的人，如张三、李四、王五。把所有国籍为“中国”的人归纳为一类，称为“中国人”，这就是一种“抽象”。再把中国人、美国人、日本人等所有国家的人抽象为“人”。在实际生活中，我们只能看到一个一个具体的人，而看不到抽象的人。

面向对象中的类是具有相同属性和行为的一组对象的集合，它能为全部对象提供抽象的描述，包括属性和行为。

类和对象的关系是抽象与具体的关系，它们的关系就像模具与用模具所生产出的产品(铸件)的关系。一个属于某个类的对象称为该类的一个实例。

3. 封装

封装是面向对象程序设计方法的一个基本特点和重要原则。它是指将对象的属性和行为组合成一个独立的单元，并尽可能隐藏对象的内部细节。所以封装有两个特点，一是将对象的全部属性和行为组合在一起，形成一个不可分割的独立单元(类)；二是需要对这个独立的单元进行信息的隐藏，使外部无法轻易获得单元中的信息，从而实现信息的保护，

外部只有通过单元的外部接口来与其发生联系。

比如，录像机里有电路板和机械控制部件，但是外面是看不到的，从外面看，它只是一个“黑箱子”，在它的表面有几个按键，这就是录像机与外界的接口，我们不必了解录像机里面的结构和工作原理，只需知道按某一个键就能使录像机执行相应的操作即可。

4. 继承

继承是面向对象程序设计中能够提高程序的可重用性和开发效率的重要保障。一个类的对象拥有另一个类的全部属性和行为，则可以将这个类声明为从另一个类继承。

如果一个一般类具有更高抽象的特征，那么它的可被继承性就更高。如果相对于这个一般类，某个特殊类具备其所有的特征，此时就可以直接继承这个一般性而简化相应的开发任务。

比如，我们了解马的特征，现在要叙述“白马”的特征，显然不必从头介绍什么是马，而只要说明“白马是白色的马”即可。这就简化了人们对事物的认识和叙述，简化了工作程序。“白马”继承了“马”的基本特征，又增加了新的特征(颜色)，“马”是父类，或称为基类，“白马”是从“马”派生出来的，称为子类，或称为派生类。

5. 多态性

多态性是指对一般类中定义的属性或行为通过继承而产生相关的不同类，它们的对象对同一消息会做出不同的响应。

比如，定义一个一般类“图形”，它具有“绘图”行为，但这个“图形”类没有具体到所表示的图形是什么形状，所以其“绘图”行为不能确定需要绘制什么样的图形。当通过继承“图形”类来定义“正方形”、“圆”等类时，它们也获得了“绘图”行为。由于不同类绘制的图形不相同，需要在“正方形”、“圆”类中分别重新定义“绘图”行为，从而实现绘制不同的图形。这就是面向对象程序设计方法中的多态性。多态性是面向对象程序设计的一个重要特征，它能够增加程序的灵活性。

1.1.2 面向对象程序设计的特点

传统的面向过程程序设计是围绕功能进行的，用一个函数来实现一个功能。所有的数据都是公用的，一个函数可以使用任何一组数据，而一组数据又能被多个函数所使用，如图 1-1 所示。程序设计者必须考虑每一个细节，什么时候对什么数据进行操作。

当程序规模较大、数据很多、操作种类繁多时，程序设计者往往感到难以应付。就如工厂的厂长直接指挥每一个工人的工作一样，一会儿让某车间的某工人在 A 机器上用 X 材料生产轴承，一会儿又让另一个车间的某工人在 B 机器上用 Y 材料生产滚珠……，显然这是非常劳累的，而且往往会遗漏或搞错。

面向对象程序设计采取的是另外一种思路，它面对的是一个对象。实际上，每一组数据都有特定的用途，是某种操作的对象，即一组操作调用一组数据。

比如，假设 a、b、c 是三角形的三条边，只与计算三角形面积和输出三角形的操作有关，与其他操作无关。我们就把这 3 个数据和对三角形操作的代码放在一起，封装成一个对象，与外界相对分隔。正如一个家庭的人生活在一起，与外界相对独立一样。

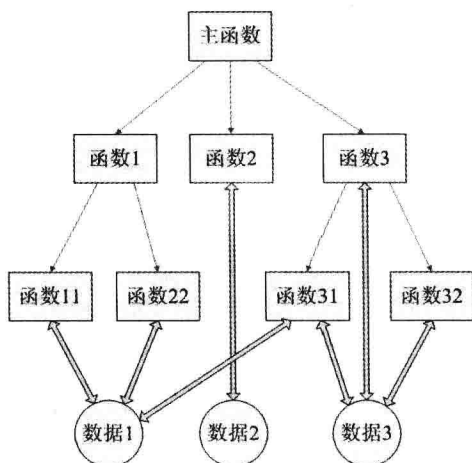


图 1-1 面向过程程序设计中的函数调用

把数据与有关操作封装成一个对象，正如工厂把材料、机器和工人承包给车间，厂长只要向不同的车间下达命令：“一车间生产 10 台发动机”，“二车间生产 100 个轮胎”，“三车间生产 15 个车身”，……；车间就会运作起来，调动工人，选择有关材料，在不同的机器上完成相关的操作，把材料变成产品，厂长可以不必过问车间内工作的细节。对厂长来说，车间就如同一个“黑箱”，只要给它一个命令或通知，就能按规定完成任务。

面向对象程序设计者的任务包括两个方面：一是设计所需的各种类和对象，即确定把哪些数据和操作封装在一起；二是考虑怎样向有关对象发送消息，以完成所需的任务。此时，如同有一个总调度，不断地向各个对象发出命令，让这些对象活动起来(或者说激活这些对象)，完成自己职责范围内的工作。当各个对象的操作都完成时，整体任务也就结束了。显然，对一个大型任务来说，面向对象程序设计方法是十分有效的，它能大大降低程序设计人员的工作难度，减少出错的机会。

1.2 “This is a C++ program.” 实例

1. 实例目的

- (1) 了解 Visual C++ 6.0 开发环境。
- (2) 掌握 Visual C++ 6.0 开发环境中各菜单的使用。
- (3) 掌握 C++ 语言的基本组成和程序执行过程。

2. 实例内容

输出一行字符：“This is a C++ program.”。

3. 实例步骤

- (1) 环境启动。选择“开始”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”命令，启动 Visual C++ 6.0 开发环境。
- (2) 建立新工程。从菜单栏中选择“文件”→“新建”命令，弹出“新建”对话框，

如图 1-2 所示。

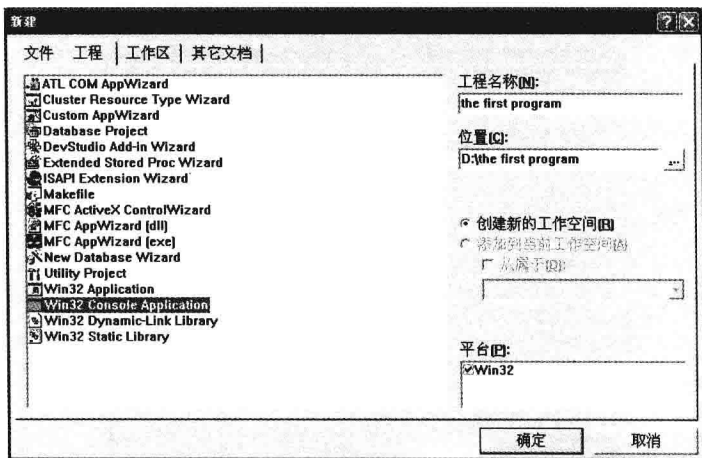


图 1-2 “新建”对话框

(3) 单击“工程”标签，进入“工程”选项卡。

(4) 选择工程类型和指定工程名。选择“Win32 Console Application”选项，然后在“工程名称”文本框中输入“the first program”。单击“位置”右侧的按钮，在弹出的窗口中进行路径选择，本实例选择 D 盘。

(5) 单击“确定”按钮，弹出工程向导对话框，如图 1-3 所示。

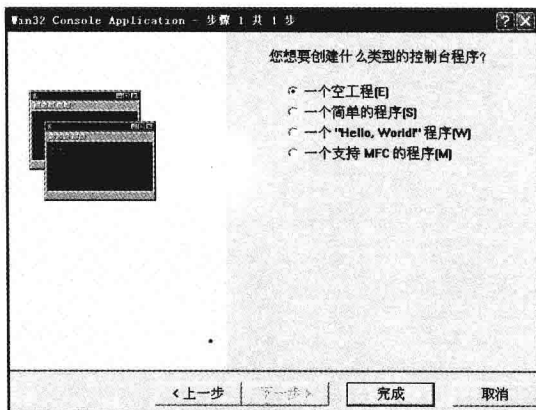


图 1-3 工程向导对话框

其中包括以下 4 个选项。

- 一个空工程：生成一个空的应用程序，工程中没有任何代码文件。
- 一个简单的程序：生成一个简单的应用程序，工程中只生成一个简单的程序框架，其中没有任何有实际意义的代码。
- 一个“Hello,World!”程序：生成一个可以向控制台输出“Hello,World!”的应用程序。
- 一个支持 MFC 的程序：生成一个支持 MFC 框架的应用程序。

(6) 完成工程建立。选择“一个空工程”单选按钮，单击“完成”按钮，出现“新建工程信息”对话框，如图 1-4 所示。



图 1-4 “新建工程信息”对话框

(7) 工程生成。单击“确定”按钮，进入 Microsoft Visual C++ 6.0 集成开发环境，如图 1-5 所示。

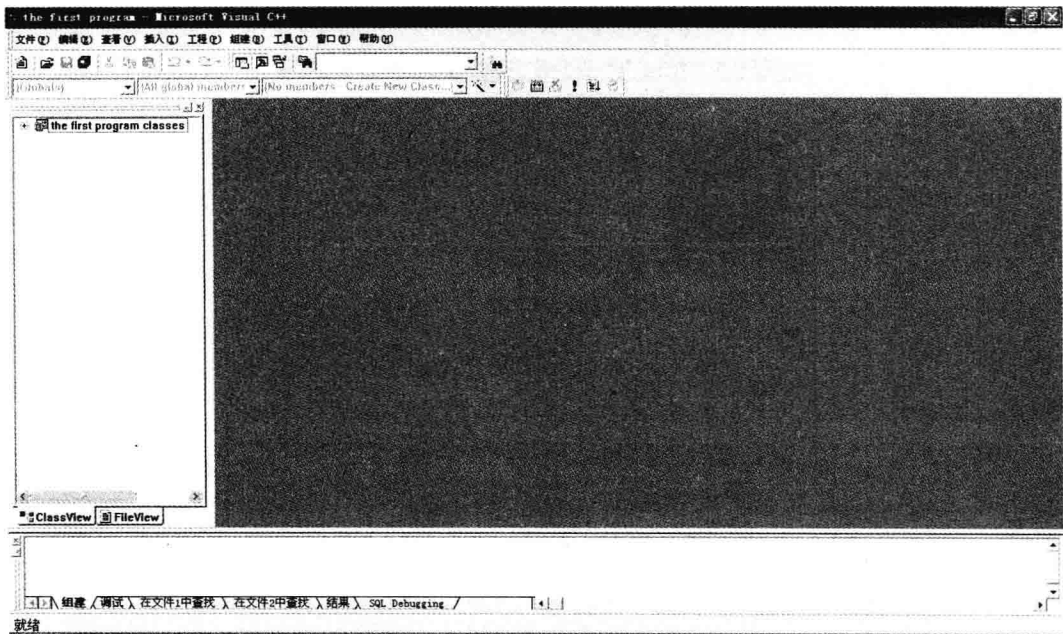


图 1-5 Microsoft Visual C++ 6.0 的操作界面

(8) 新建文件。选择“文件”→“新建”菜单命令，弹出如图 1-6 所示的“新建”对话框。单击“文件”选项卡，选择“C++ Source File”选项，然后在“文件名”文本框中输入“the first program”。

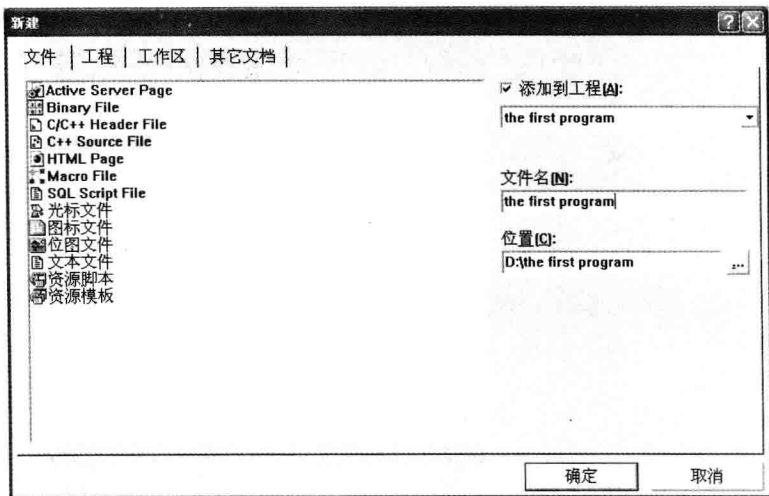


图 1-6 “新建”对话框

(9) 编辑程序。单击“确定”按钮，进入程序编辑窗口，在窗口中输入程序，如图 1-7 所示。



图 1-7 程序编辑窗口

(10) 编译和连接程序。从菜单栏中选择“组建”→“组建”命令，编译器会完成对程序的预处理及编译和连接，最后生成可执行文件。编译和连接正常结束后，将在工程目录的 Debug 文件夹下新增 the first program.obj 和 the first program.exe 两个文件。

(11) 执行程序。从菜单栏中选择“组建”→“执行”命令，编译后的程序会被执行，程序运行结果如图 1-8 所示。



图 1-8 程序运行结果

1.2.1 C++程序的基本组成

一个 C++ 程序主要由预处理命令、输入输出、函数、语句和变量组成。除此之外，还有一些诸如常量和注释等，也是程序的一部分。

【例 1.1】 输入两个任意整数，输出二者之差的绝对值。代码如下：

```
#include <iostream>           //包含头文件 iostream
using namespace std;         //使用命名空间 std
int main()                   //main 是主函数，程序从主函数开始执行
{                             //花括号之间是主函数 main 需要完成的任务
    int a, b, dif;           //声明 a、b 和 dif 是 3 个整型(int)的变量
    cout << "请输入被减数" << endl; //显示“请输入被减数”，endl 是换行的意思
    cin >> a;                 //从键盘对变量 a 输入整数
    cout << "请输入减数" << endl; //显示“请输入减数”，endl 是换行的意思
    cin >> b;                 //从键盘对变量 b 输入整数
    dif = a - b;             //求出 a-b，通过赋值运算符“=”赋给变量 dif
    cout << "二数之差的绝对值是："; //屏幕显示“二数之差的绝对值是：”
    if(dif < 0) cout << -dif << endl; //若表达式 dif<0 为真，则输出-dif 值，换行
    else cout << dif << endl;    //否则，输出 dif，换行
    return 0;               //程序正常执行，返回 0
}
```

程序运行结果如图 1-9 所示。

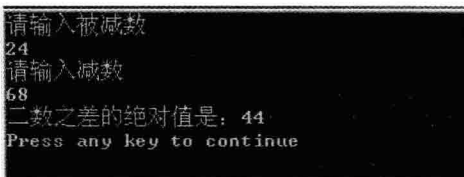


图 1-9 例 1.1 的运行结果

下面通过这个例子，对 C++ 的基本组成进行简单介绍。

1. 注释

在程序代码的右边，以“//”开头的一行字是程序的注释。这种注释的写法只限在一行之内有效，如果一个注释超过了一行，则注释必须改为以“/*”开头，并以“*/”结尾。

注释的作用是对程序做些解释，以便让人们能看懂程序，即提高程序的可读性。比如我们把上面例子的程序代码结合注释来看，理解起来就比较容易。但是，注释不参与程序的运行。

2. 包含头文件和使用命名空间 std

按标准 C++ 的新要求，程序的头两行用了“#include <iostream>”和“using namespace std”，意思是包含头文件 iostream 和使用命名空间 std。这两行是为程序中所要用的输入和输出提供必要支持的，其道理暂且先不去深究。但要注意，每逢程序设计，开头总少不了要写上这两行。

3. 主函数

int main()表示主函数，main 是函数名，int 表示函数返回一个整数。程序总是从主函数 main 开始执行。下面花括弧所括起来的内容是函数所需要完成的任务，称为函数体。若程序正常执行完毕，最后向操作系统返回数值 0(即 return 0)，否则返回数值-1(即 return -1)。

4. 标识符

标识符是用来表示程序中实体名字的有效字符序列。比如，定义变量名、函数名、数组名等，程序中的变量 a、b 和 dif 都称为标识符。标识符由字母、数字和下划线组成，且只能由字母或下划线开头。注意，大小写字母有区别。比如_B2b 和_b2b 都是合法的标识符，但由于第二个字符有大小写之分，所以二者不可视为同一标识符。

5. 关键字

在 C++ 语言中，已经具有特定含义的词叫作关键字。比如 int、if、else 等都是关键字，不能作为标识符来使用。表 1-1 列举了常用的关键字。

表 1-1 常用的关键字

bool	break	case	catch
char	class	const	continue
default	delete	do	double
else	enum	extern	false
float	for	friend	goto
if	inline	int	long
namespace	new	operator	private

续表

protected	public	return	short
signed	sizeof	static	struct
switch	template	this	throw
true	try	typedef	typename
union	unsigned	using	virtual
void	while		

6. 语句

程序中语句用分号“;”结尾，分号是语句的组成部分。在一行中可写多条语句，每条语句用分号隔开，而一条语句可写多行。书写代码时，缩进和分行只是为求醒目，便于看懂而已。

7. endl

endl 的含义是 end of line，表示结束一行。在头文件 `iostream` 中定义了控制符 `endl` 代表回车换行操作，作用与“`\n`”相同。

1.2.2 C++程序的编写和实现

一个程序从编写到最后得到运行结果，要经过编辑、编译、连接和运行 4 个步骤。

1. 编辑

所谓程序，就是一组计算机系统能识别和执行的指令，每一条指令使计算机执行特定的操作。用高级语言编写的程序称为“源程序”(Source Program)，C++的源程序是以 `.cpp` 作为后缀名的(`cpp` 是 `c plus plus` 的缩写)。

2. 编译

从根本上说，计算机只能识别和执行由 0 和 1 组成的二进制的指令，而不能识别和执行用高级语言编写的指令。为了使计算机能执行高级语言源程序，必须先用一种称为“编译器(Compiler)”的软件(也称编译程序或编译系统)，把源程序翻译成二进制形式的“目标程序(Object Program)”。

编译是以源程序文件为单位分别进行编译。目标程序一般以 `.obj` 作为后缀名(object 的缩写)，编译的作用是对源程序进行词法检查和语法检查，编译时对文件中的全部内容进行检查，编译结束后会显示出所有的编译出错信息。编译系统给出的出错信息一般分为两种：一种是错误(Error)；另一种是警告(Warning)，即一些不影响运行的轻微错误。如果检查出 Error 类的错误，就不能生成目标程序，必须改正后，重新编译。

3. 连接

在改正所有的错误并全部通过编译后，得到一个或多个目标文件。此时，用编译系统