高等院校信息技术规划教材

C++程序设计基础教程

孙　涛 主编
周李涌　王丽颖 副主编
张晓琳 主审

清华大学出版社

高等院校信息技术规划教材

C++程序设计基础教程

孙涛 主编
周李涌 王丽颖 副主编

清华大学出版社
北京

内容简介

本书系统地介绍了C++的语法规则和面向对象的程序设计方法，重点突出、逻辑清晰、简洁明了、深入浅出。书中精选了丰富的例题，对立体的语法规则、设计思路和输出结果做了详尽的解释和分析，所有例题均在Visual Studio C++ 6.0环境下运行通过。

本书适合作为高等学校计算机专业及相关专业的C++程序设计课程教材，也可作为编程人员的自学参考书。

图书在版编目(CIP)数据

C++程序设计基础教程/孙涛主编．—北京：清华大学出版社，2013
高等院校信息技术规划教材
ISBN 978-7-302-33681-5

Ⅰ．①C…　Ⅱ．①孙…　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204673号

责任编辑：张　玥
封面设计：傅瑞学
责任校对：白　蕾
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，010-62795954

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**：21.75　　**字　　数**：527千字
版　　次：2013年11月第1版　　**印　　次**：2013年11月第1次印刷
印　　数：1～2000
定　　价：34.50元

产品编号：055452-01

前　言

随着计算机技术的发展，种类繁多、特点各异的计算机语言在不同领域被广泛应用。C++作为一种高级语言，既具有面向对象的特征，又保留了传统的面向过程的程序设计方法的优点。它的可移植性、可靠性和高效率一直被广大程序开发人员所青睐。在C语言的基础上，C++增加了面向对象编程、数据抽象、类属编程等技术支持，通过继承和多态性，使程序具有很高的可重用性，使软件的开发和维护更方便。C++语言既可以用于开发系统软件，也普遍用于开发应用软件，同时也广泛地应用于科研和教学。

本书内容深入浅出，概念全面，突出重点，侧重基础。针对初学者和自学者的特点，力求用通俗易懂、简洁浅显的语言讲述复杂的概念，不但适合作为计算机专业程序设计基础课教材，也可作为电子信息、通信、自动化等相关专业的程序设计课程教材。

参与本书编写的作者都是长期工作在教学和科研第一线教师，有多年的C++程序设计课程的教学和编程经验。在编写过程中，作者总结了日常教学的经验，对一些常见问题进行了重点说明和讲解，力争使本书零距离贴近教学实际。

本书共12章，每章配备了丰富的实例讲解和精选习题，帮助读者更好地掌握本章内容，提高计算思维能力。本书中所有例题均在Microsoft Visual C++ 6.0的环境下调试通过，并配有运行截图。本书配套电子教案及书中所有章节的例题源程序代码均可从清华大学出版社网站(http://www.tup.com.cn)上的本书页面中下载。

本书第4章和第5章由孙涛编写，第8章和第9章由周李涌编写，第1章和第6章由王丽颖编写，第3章和第10章由杜鹏东编写，第7章由高鹭编写，第2章由徐喜志编写，第11章和第12章由郝斌编写。全书由张晓琳教授修改定稿。

本书获得2013年度内蒙古科技大学教材基金资助。

由于编者水平有限，疏漏之处在所难免，恳请读者批评指正。

编　者

2013年7月

目　录

第1章 绪 论

1.1 面向对象程序设计方法

C++是一种混合式面向对象语言，它既可以进行面向过程的结构化程序设计，也可以采用面向对象的程序设计方法。下面对这两种不同的程序设计方法进行介绍。

1.1.1 面向过程程序设计

C++可以看做是C语言的扩展和改进。C语言是一种面向过程的语言，C语言程序由模块构成，这些模块可以分别进行设计、编码并单独编译，一个模块就是一个过程。

面向过程程序设计通常采用自顶向下的设计方法，就是将一个复杂问题分解为若干易于处理的小问题，而每一个小问题都可以独立编写程序来解决，从而将整个系统划分成多个子模块，程序中的子模块在C语言中通常用函数实现。

面向过程程序设计方法有许多优点，如程序的各个模块可以独立编写，程序便于阅读、修改和维护。对复杂问题的处理，面向过程程序设计成功地提供了有力手段，易于为广大的程序设计人员接受和使用，成为当时程序设计的主要方法。

在面向过程程序设计中，由于把解决问题的过程作为程序设计的重点，程序中的数据与处理数据的过程(如函数)是分离的，当程序需要处理的数据量增大时，数据与处理这些数据的方法之间的分离使程序变得越来越难以理解。对某段程序进行修改或删除时，整个程序中所有与之相关的部分都要进行相应的修改，从而使程序的开发和维护变得更复杂。软件技术的发展要求程序的数据处理能力越来越强，面向过程的程序设计方法对于大规模软件系统的描述和实现也更加困难。因此，面向对象的程序设计方法应运而生。

1.1.2 面向对象程序设计

面向对象的程序设计方法继承了结构化、过程化、模块化等方法的所有积极成分，创造性地引入了“对象”的概念。对象是构建程序的基础，是由数据和对数据的操作(方法)封装而成的一个整体。封装使得算法和数据结构的关系成为相互依存的关系，而不再是过程化程序设计中算法对数据结构的单方依赖关系。

为了理解面向对象的程序设计方法，首先就需要理解类和对象。简单地说，类是对象的抽象，对象是类的实例。对象是实实在在存在的事物，比如苹果、橘子、菠萝，对这些客观事物的共同特性进行抽象，就得到了类。苹果、橘子、菠萝都是多汁且有甜味的植物果实，它们就被抽象为水果类。在程序设计中，还可以用数据类型和变量来描述类和对象的关系。在语句int a;中，int是整型数据类型，a是整型变量，用数据类型来定义变量。可以这样理解类和对象，类相当于数据类型，对象就相当于变量，用类去创建对象。

1.1.3 面向对象程序设计的特征

对象的封装性、继承性和多态性是面向对象方法最重要的三大特征，它们是相互联系的。封装性是面向对象方法的基础，继承性是面向对象方法的技术支持，多态性提供了面向对象方法设计的灵活性。

1. 封装性

封装性是面向对象方法的一个重要原则，是把对象的一组属性和一组操作组装在一起，形成一个独立的实体，对外界尽可能地隐藏对象的内部细节。

封装具有两层含义：第一是把对象的全部属性和操作组合在一起，形成一个不可分割的封装体；第二是对于不允许外界直接操作的数据以及外界没必要了解的数据封装起来，隐藏在对象的内部，外界通过对象提供的接口来访问这些数据，实现信息的隐蔽性，同时增强了数据的安全性。比如一台微型计算机的台式机，是一个实实在在存在的对象，这个对象有CPU、主板、内存、硬盘、电源等部件，它们通常封装在主机机箱内部，对这些部件的操作通过机箱上的接口来进行，按下电源按钮启动计算机，通过 USB 接口连接 U 盘，通过并行接口连接打印机等。

2. 继承

对象是客观世界的实体。共性与个性、一般与特殊是客观世界中的实体之间普遍存在的关系。面向对象的程序设计中如何体现这些实体间的关系呢？还是来看水果的例子。香蕉、菠萝、火龙果等属于水果，它们还有一个共同特点，就是生长在热带，所以它们是热带水果；苹果、梨、葡萄都是生长在温带，属于温带水果，图 1-1 表示了这种关系。

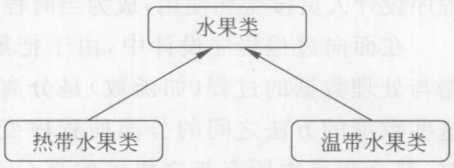

图 1-1　继承与派生的关系

继承是在类、子类以及对象之间自动共享数据和共享操作方法的一种机制。在图 1-1 中，称水果类为基类，称热带水果类和温带水果类为水果类的派生类。基类通常也称为父类，派生类称为子类。派生类可以继承基类的数据和操作，也可以新增自己的数据和操作。因此，基类抽象出了其所有派生类的共同特征，在设计派生类时，继承性允许程序设计人员通过新增自己的数据和操作来体现其个性特点，如果基类中的某些行为不适用于派生类，设计人员也可以重置这些行为。

在面向对象程序设计方法中，继承分为单一继承和多重继承；一个子类有两个或两个以上父类，这种继承称为多重继承，简称多继承。C++ 语言既支持单继承，又支持多继承。继承的功能体现出重复使用的特性。继承性又称可重用性。当要对设计好的系统的原有功能进行补充和修改时，可以不必对已有的类进行修改，而用继承的特性来设计子类，补充、修改和添加新的功能。继承的引入实现了代码的可重用性和可扩充性，提高了软件的开发效率。

3. 多态性

多态性是面向对象程序设计中的一个重要概念，它是指同一个消息被不同类型的对象接收时所产生的不同行为。消息主要是指对函数的调用，不同的行为是指调用了不同的函数成员。生活中也有很多多态性的例子。比如，家用电器是对某些数据进行操作的函数。电饭煲这个函数对水和大米的数据进行操作，结果返回的是粥；榨汁机对水果和水进行操

作，返回的是果汁；豆浆机对黄豆和水进行操作，返回的是豆浆。在上述操作中引入多态性，就是只有一台多功能料理机，这个料理机对不同的数据有不同的操作，如果向料理机中加入水果和水，料理机就进行制作果汁的操作，如果向料理机中加入黄豆和水，料理机就制作豆浆，这就是多态。简单地说，多态性的特点是“一个接口，多种操作”。

面向对象方法的多态性可以分为四类：重载多态、强制多态、参数多态和包含多态。

重载多态是指两个以上的函数具有相同的函数名，同名函数却具有不同的功能，如第3章第7节中介绍的函数重载和第8章第4节中介绍的运算符重载就属于重载多态。典型的例子就是运算符，如可以使用同样的加号＋实现整数之间、浮点数之间、双精度浮点数之间以及它们相互的加法运算，即同样的消息“加”，被不同的对象“变量”接收，不同类型的变量就会采用不同的方式进行加法运算。这就是系统提供的运算符重载方式，利用同样的运算符重载概念，可以实现自定义类的对象之间的运算符操作。

强制多态是指对一个对象或变量的类型加以变换，以符合一个函数或某个操作的要求。这种类型变换可以有显式或隐式两种。C++语言中强制类型转换就是显式类型变换的例子。

参数多态是为了使代码具有通用性。功能、结构、实现相同的代码不受参数类型的限制，可以自动适应各种数据类型的变化，这就是参数多态，也称为类型参数化多态，如第9章第1节和第2节介绍的函数模板和类模板就是参数多态的典型例子。

包含多态就是指通过继承，父类和子类的对象共享许多相似的特性，而由于多态性，子类的对象可以有独自的行为方式。本书重点介绍虚函数，作为包含多态的例子。

由于多态是相同的消息被不同的对象接收时表现出不同的操作，因此多态实现时必须要确定对于这个对象到底要进行哪个操作，这个将对象和对对象的操作对应起来的过程就叫绑定(Binding)或联编。

在程序编译链接阶段完成的联编，也就是在编译的过程中确定了消息的操作对象，称为静态绑定。静态绑定的多态称为编译时的多态。重载多态、强制多态和参数多态都属于编译时的多态。如果这个绑定是在程序运行阶段完成的，称为动态绑定。动态绑定的多态称为运行时的多态，包含多态就属于运行时的多态。

1.2 一个简单的C++程序

下面以一个具有输入和输出功能的简单的C++程序为例，介绍C++程序的构成。

【例1-1】 一个简单的C++程序。

```
1. /*1-1.cpp*/
2. #include <iostream.h>
3. void main()
4. {
5.     int stu_number;
6.     cout<<"please input your number:";
7.     cin>>stu_number;
8.     cout<<"hello!"<<stu_number<<endl;
9. }
```

程序的首行是以/*符号开始的注释行。

程序的第2行以#符号开始，这是编译预处理行，程序中用到的输入和输出对象cin和cout是在文件iostream.h中定义的。如果要使用，就要在程序中加入#include <iostream.h>语句，它会指示C++的编译器，编译时将程序iostream.h的代码嵌入到程序中该指令所在的位置，这个过程称为编译预处理，iostream.h称为头文件。

第3、4、9行void main(){}是程序的主函数，一个C++的程序可以由很多函数构成，main函数是C++程序中第一个被执行的函数，指示程序的入口位置。一个C++的程序有且只有一个main函数。

第5行定义了一个整型(int类型)的变量stu_number。

cout是C++中用做输出的对象。C语言中用函数printf()和scanf()进行输出和输入，C++中的输入和输出用对象来操作，cout代表一个标准输出流设备，将要输出的常量、变量用<<操作符送到这个输出设备上输出。简单地说，cout就是将用双引号引起来的字符串常量原样输出，将<<后面的变量或表达式的值输出。endl表示输出回车换行符。同理，cin是用做输入的对象，cin将键盘输入的值送到>>后面的变量所对应的存储单元中，作为变量的值。

接下来将运行这个程序。运行程序之前，先介绍C++的开发环境。

1.3 C++程序开发平台

众所周知，计算机只能识别机器语言，所有的程序在计算机运行之前，都必须由编译器翻译成机器语言。C++的源程序，也称为源文件，需要经过编译，产生对应的目标程序。由于目标程序地址的浮动性，因而它还需要经过链接生成可执行程序，最后运行可执行程序，才能得到运行结果。

Visual C++ 6.0是由Microsoft公司提供的Windows环境下的集成开发环境。集成开发环境将C++源程序的创建、编辑以及其后的编译、链接、调试、运行全部都集成在一起，为编程人员提供了方便的程序开发环境。

1.3.1 界面介绍

启动Visual C++ 6.0，进入Visual C++ 6.0的集成开发环境，界面如图1-2所示。

(1) 标题栏：显示当前项目的名称和当前编辑文档的名称。

(2) 菜单栏：用户通过选取各个菜单项执行常用操作。

(3) 工具栏：工具栏中的工具按钮可以完成常用操作命令，它实现的功能与菜单相同，但比菜单操作快捷。

(4) 项目工作区窗口：列出当前应用程序中所有类、资源和项目源文件。

(5) 文件编辑窗口：用户可以编辑源程序代码，同时显示各种程序的源代码文件。

(6) 输出窗口：它显示编译、链接和调试的相关信息。如果进入程序调试状态，主窗口中还将出现一些调试窗口。

(7) 状态栏：状态栏用于显示文本信息，包括对菜单、工具栏的解释提示以及键盘按键的状态等。

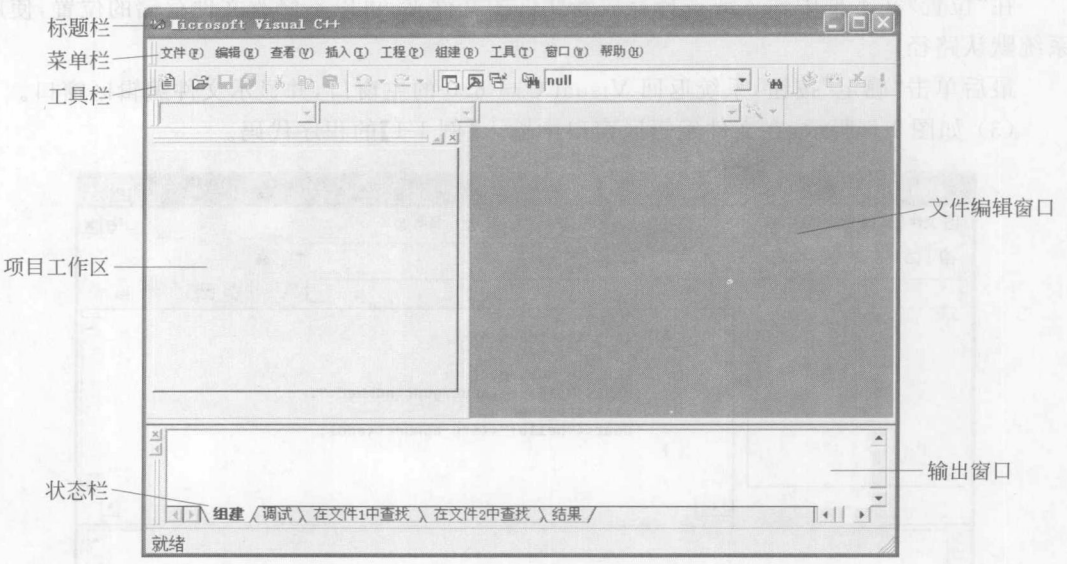

图 1-2　C++ 6.0 的集成开发环境

1.3.2　编辑 C++ 源程序

(1) 进入 Visual C++ 6.0 的集成开发环境后，从菜单栏中选择“文件”选项，选择下拉菜单的“新建”选项，如图 1-3 所示。

(2) 如图 1-4 所示，在弹出的“新建”对话框中单击“文件”标签，在“文件”选项卡中单击 C++ Source File 选项。

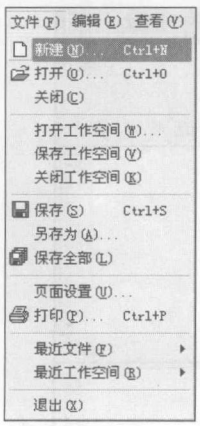

图 1-3　文件菜单

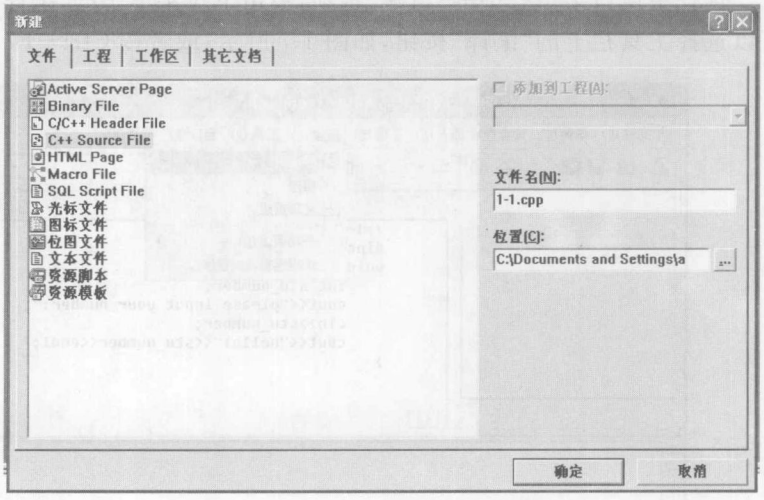

图 1-4　“新建”对话框

在“文件名”文本框中输入文件名 1-1.cpp，C++ 文件的扩展名为 cpp，输入时可以省略扩展名，只输入文件名 1-1。如果文件名不确定，这个步骤中也可以不输入文件名，在编译程序时会要求用户输入文件名。