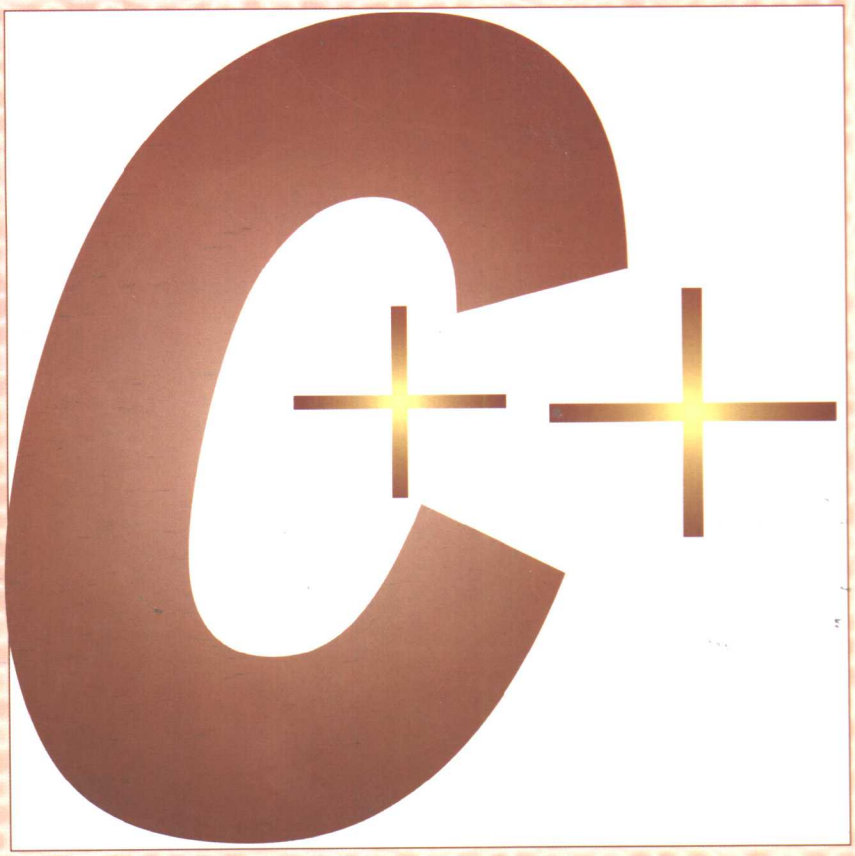
计 算 机 应 用 技 术 教 材

C++语言与应用基础

成 岩 周 露 杨嘉伟 编著



科 学 出 版 社

计算机应用技术教材

C++语言与应用基础

成 岩 周 露 杨嘉伟 编著

科学出版社

2002

内 容 简 介

本书以 C++语言与编程应用的紧密联系为编写宗旨，全书分为 C++语言和编程应用两大部分。C++语言部分重点介绍 C++语言的语法和面向对象语言的封装性、继承性、多态性等内容。编程应用部分以 VC 为开发工具，主要介绍如何借助 VC 的开发环境、开发工具和 MFC 类库，应用 C++语言编程实现 Windows 应用程序及其常见图形用户界面设计的核心问题，每章都相应给出可实践的编程范例。

本书的特色是把 C++语言与编程应用紧密联系起来，内容新颖，科学性和实用性强。

本书可作为本科生、研究生教材，亦可作为各类自学读者、软件开发人员的参考用书。

版权所有 翻印必究。

图书在版编目（CIP）数据

C++语言与应用基础/成岩，周露，杨嘉伟编著. —北京：科学出版社，2002

（计算机应用技术教材）

ISBN 7-03-010634-2

Ⅰ.C… Ⅱ.①成… ②周… ③杨… Ⅲ.C 语言—程序设计， Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2002）第 049474 号

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

http://www.sciencep.com

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2002 年 8 月第 一 版 开本：787×1092 1/16

2002 年 8 月第一次印刷 印张：23 1/2

印数：1—5 000 字数：546 000

定价：30.00 元

（如有印装质量问题，我社负责调换〈新欣〉）

前　言

C++语言是目前面向对象编程的首选语言。目前市场上有关C++方面的书籍，按内容结构大概可分为如下四类：第一类是VC软件本身的体系结构来介绍；第二类是联系应用程序的各种界面和部件是如何用VC来实现；第三类是以程序实例来编写；第四类是就某一专门方面的编程要来编写，如图形、图像、数据库、多媒体、网络等。这些书适用于不同程度的读者学习VC的需求，但不适合于做C++语言的教材。这种书大多不介绍C++面向对象编程的思想和方法，学习者只能死跟随着书中例子的步骤操作，而且不知所以然。

本书从实际应用程序的设计出发，通过详细的实例对其概念进行描述。主要是将C++语言与C++编程相结合，如在讲解C++语法时，将C++语言的编程与在当今广泛应用的Windows的图形用户界面相结合，使学习者直接感受到C++面向对象编程的好处，了解C++是如何体现面向对象编程的，并掌握如何利用C++编程实现Windows应用程序将理论与实践相结合，使读者能够轻松地掌握其概念并用于实际工作中。

本书内容分为两部分。C++语言部分重点介绍类的封装性、继承性、多态性方面的内容、编程应用部分以VC为开发工具，主要介绍如何借助VC开发环境、开发工具和MFC类库，应用C++语言知识来编程实现Windows应用程序及其常见图形用户界面设计等核心问题，每章都相应给出一些可动手实践 的编程实例。

全书共12章，主要内容包括，面向对象程序设计概述、数据类型、运算符和控制结构、数组、函数和指针、类与封装性、派生类与继承性、运算符重载、虚函数与多态性、应用程序设计基础、窗口、菜单和绘图程序设计、文档/视图结构程序设计，为对话框与控件程序设计，面向对象的程序设计，多线程程序设计。

本书可作为本科生和研究生的教材，亦可作为各类自学者和软件开发人员的参考用书。

本书由成岩、周露和杨嘉伟编写。周露和杨嘉伟负责编写第1至6章，成岩负责编写第7至12章，最后由成岩统一修改、整理和定稿。由于作者水平有限，书中难免还存在缺点和不足，殷切希望广大读者批评指证。

作　者

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

此为试读，需要完整PDF请访问：www.ertongbook.com

第 1 章　面向对象程序设计概述

本章简要介绍编程语言的发展、面向对象程序设计语言的起源与特点和面向对象方法的由来及其基本概念，然后介绍面向对象的软件开发，最后介绍 C++程序在 VC 7.0 中的编译运行。

1.1　程序设计语言的发展

1.1.1　编程语言的发展

自从 1946 年 2 月世界上第一台数字电子计算机 ENIAC 诞生以来，在这短短的 50 多年间，计算机科学迅猛发展，计算机及其应用已渗透到社会的各个领域，有力地推动了整个信息化社会的发展，计算机已成为信息化社会中必不可少的工具。

计算机系统包括硬件系统和软件系统。计算机之所以有如此强大的功能，不仅因为它具有强大的硬件系统，而且也依赖于软件系统。软件包括了使计算机运行所需的各种程序及有关的文档资料。计算机的工作是用程序来控制的，离开了程序，计算机将一事无成。

机器语言是一种最原始的编程语言，这种语言是计算机可以直接识别的语言。这种语言使用 0 和 1 两种代码，编写出的程序难以理解和记忆，因为它远不同于人们的习惯思维方式。

汇编语言是一种使用助记符号来替代代码 0 和 1，是一种低级语言，它比机器语言稍有提高，符合人们的形象思维，它是低层次的抽像，对于汇编语言，计算机不能直接识别，需要编译后才可识别。这种语言虽然效率较高，但是由于难以记忆，使用较少。

高级语言的出现是计算机编程语言的一大进步。它屏蔽了机器的细节，提高了语言的抽像层次，程序中可以采用具有一定含义的数据命名和容易理解的执行语句。这使得在书写程序时可以联系到程序所描述的具体事物。

20 世纪 60 年代末开始出现的结构化编程语言进一步提高了语言的层次。结构化数据、结构化语句、数据抽像、过程抽像等概念，使程序更便于体现客观事物的结构和逻辑含义。这使得编程语言与人类的自然语言更接近。但是二者之间仍有不少差距。主要问题是程序中的数据和操作分离，不能够有效地组成与自然界中的具体事物紧密对应的程序成分。

目前应用比较广泛的几种高级语言有 FORTRAN、BASIC、PASCAL、C 等。当然本书介绍的 C++语言也是高级语言，但它与其他面向过程的高级语言有着根本的区别。

面向对象的编程语言与以往各种编程语言的根本不同点在于，它设计的出发点就是为了能更直接地描述客观世界中存在的事物（即对象）及其之间的关系。

开发了一个软件是为了解决某些问题，这些问题所涉及的业务范围称为该软件的问题域。面向对象的编程语言将客观事物看作具有属性和行为的对象，通过抽像找出同一类对象的共同属性（静态特征）和行为（动态特征），形成类。通过类的继承与多态可以很方便地实现代码重用，大大缩短了软件开发周期，并使得软件风格统一。因此，面向对象的编程语言使程序能够比较直接地反映问题域的本来面目，软件开发人员能够利用人类认识事物所采用的一般思维方法来进行软件开发。

面向对象的程序设计语言经历了一个很长的发展阶段。例如，LISP 家族的面向对象语言，Simula 67 语言，Smalltalk 语言，以及 CLU、Ada、Modula－2 等语言，或多或少地都引入了面向对象的概念，其中 Smalltalk 语言是第一个真正的面向对象的程序语言。

然而，应用最广的面向对象程序语言是在 C 语言基础上扩充出来的 C++语言。由于 C++对 C 兼容，而 C 语言又早已被广大程序员所熟知，所以，C++语言也就理所当然地成为应用最广的面向对象程序语言。

1.1.2　C++语言的起源与特点

C++语言是在 C 语言的基础上为支持面向对象的程序设计而研制的一个通用目的的程序设计语言，它的开发者是 AT&T 贝尔实验室的 Bjarne Stroustrup 博士。

C 语言是 1972 年由 Dennis Richie 在 AT&T 贝尔实验室设计的一个通用目的的程序设计语言，它的前身是 B 语言，而 B 语言又是在继承和发展了 BCPL 语言的基础上设计的。C 最初用作 UNIX 操作系统的记述语言，由于 UNIX 的成功和广泛使用，也使 C 成为一种普遍使用的程序设计语言，目前在各种机型和各种操作系统上都运行有 C 编译器。C 有广泛的应用基础，有众多的程序员在使用 C 语言，并且，有许多 C 语言的库代码和开发环境。

研制 C++的一个首要目标是使 C++首先是一个更好的 C，所以，C++要根除 C 中的问题。同时，在 C++中引入了类等机制来支持面向对象的程序设计。所研制的这个语言最初被称为“带类的 C”，1983 年取名为 C++，以后又经过不断完善和发展，成为目前的 C++。除了个别的例外情况，它将 C 作为它的子集，并引入了其他语言中的一些概念和机制。它确实达到了使 C++成为更好的 C 的目标，并且还保持了 C 的简洁、高效和接近汇编语言的特点。C++对 C 的类型系统进行了改进和扩充，因此，C++比 C 更安全，C++编译器能检查出更多的类型错误。研制 C++的另一个主要目标是支持面向对象的程序设计。面向对象的程序设计是一种更好的程序设计技术，它通过使程序员在程序中能够定义更多更强有力的类型，使 C++语言对更高级的抽像有更好的支持，这种支持使在计算机上解决问题的方式更加类似于人类的活动。

C++保持与 C 兼容，这就使许多 C 代码不经修改就可以为 C++所用，用 C 编写的众多的库函数和实用软件可以用于 C++中；更重要的是，C 程序员仅需学习 C++语言的新特征，就可以很快地用 C++编写程序。另外，C++编写的程序可读性更好，代码结构更为合理，可以更直接地在程序中映射问题空间的结构。

C++对 C 的兼容性对 C++的快速普及是功劳卓著的，C++极大地促进了对面向对象技术的广泛应用。在 20 世纪 80 年代出现了许多面向对象的语言和工具，但 C++是比较高效

的，并且有广泛的程序员使用基础。

但需要指出的是，C++对 C 的兼容使 C++不是一个纯正的面向对象的语言。因为 C 语言不是面向对象的语言，而是面向过程的语言，所以，C++也必须支持 C 的面向过程的程序设计。由于两种不同风格的程序设计技术融于同一个语言中，使初学 C++的程序员感到有些困难。C++的初学者不仅需要记忆一些本不需要记忆的东西，还需要掌握用面向对象的方法去构造程序。

所以，在学习 C++时，我们必须首先明确 C++是一门完整的语言，它提供特定的机制去支持特定的程序设计技术——面向对象的程序设计。保持与 C 的兼容是为了保护在 C 上已做的大量的投资，同时，使 C 程序员在转向使用新技术时，对于新语言的语法和语言中的一些机制的学习不致于感到太困难。在新的技术——面向对象的程序设计面前，C 程序员也是新手。从 C++面向对象的程序设计需要的角度去考察 C++中的 C 成份，我们会发现，C 中的大部分成份在 C++中被降到次要的地位，其原因在于，面向对象的程序设计从一个不同于面向过程的程序设计的角度去观察程序和构造程序。虽然与 C 兼容部分不是 C++的主要部分，但仍然不能超越它，像数据类型、算法的控制结构、函数等，不仅是面向过程程序设计的基本成分，也是面向对象编程的基础。

1.1.3　C++源程序的构成

C++是 C 的一个超集，它几乎保留了 C 的所有的特性，下面给出一个简单的 C++程序，以便读者对 C++程序的格式有一个初步的了解。

【例 1.1】　输出一行字符。

```
#include <stdio.h>
#include<iostream.h>
/*本程序的作用是输出一行字符*/
void main()
{
printf("This is a C++ program.\n");
cout<<"This is a C++ program.\n";//本行输出一行字符
}
```

程序运行结果：

```
This is a C++ program.
This is a C++ program.
```

分析：

（1）在 C++程序中一般习惯在主函数 main 前面加了一个类型声明符 void，表示 main 函数没有返回值。

（2）除了可以用/*……*/形式的注释行外，还允许使用以//开头的注释。从程序最后一行中可以看到：以//开头的注释可以不单独占一行，它出现在语句之后。编译系统将//以后到本行末尾的所有字符都作为注释。应注意：它是单行注释，不能跨行。C++的程序设计

人员多愿意用这种注释方式，它比较灵活方便。

（3）除了可以用 printf 函数输出信息外，还可以用 cout 进行输出。cout 要与运算符<<配合使用，程序中 cout 的作用是将<<运算符右侧的内容送到输出设备中输出（有关 cout 的使用，将在 2.3 节中详细介绍）。

（4）使用 cout 需要用到头文件 iostream.h，在程序的第一行用#include 命令将该头文件“包含”进来。

程序运行时输出：

```
This is a C++ program.
This is a C++ program.
```

可以看到程序中最后两个语句的作用相同，都是输出 This is a C++ program。

1.2　面向对象的方法

程序设计语言是编写程序的工具，因此程序设计语言的发展恰好反映了程序设计方法的演变过程。这里我们首先初步介绍一下面向对象方法的基本概念和基本思想，当您学习完本书之后，对面向对象的方法会有一个深入、完整的认识。

1.2.1　面向对象方法的由来

在面向对象的方法出现以前，我们都是采用面向过程的程序设计方法。早期计算机是用于数学计算的工具，例如，用于计算炮弹的飞行轨迹。为了完成计算，就必须设计出一个计算方法，或解决问题的过程。因此，软件设计的主要工作就是设计求解问题的过程。

随着计算机硬件系统的高速发展，计算机的性能越来越强，用途也更加广泛，不再仅限于数学计算。由于所处理的问题日益复杂，程序也就越来越复杂和庞大。20 世纪 60 年代产生的结构化程序设计思想，为使用面向过程的方法解决复杂问题提供了有力的手段。因而，在 70 年代到 80 年代，结构化程序设计方法成了所有软件开发设计领域及每个程序员都采用的方法。结构化程序设计的思路是：自顶向下、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；其模块化实现的具体方法是使用子程序。结构化程序设计由于采用了模块分解与功能抽像，自顶向下、分而治之的方法，从而有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子任务，便于开发和维护。

虽然结构化程序设计方法具有很多的优点，但它仍是一种面向过程的程序设计方法。它把数据和处理数据的过程分离为相互独立的实体，当数据结构改变时，所有相关的处理过程都要进行相应的修改，每一种相对于老问题的新方法都要带来额外的开销，程序的可重用性差。另外，由于图形用户界面的应用，使得软件使用起来越来越方便，但开发却越来越困难。一个好的软件，应该随时响应用户的任何操作，而不是请用户按照既定的步骤

循规蹈矩地使用。例如，我们都熟悉文字处理程序的使用，一个好的文字处理程序使用起来非常方便，几乎可以随心所欲，软件说明书中决不会规定任何固定的操作顺序，因此对这种软件的功能很难用过程来描述和实现，如果仍使用面向过程的方法，其开发和维护都将很困难。

那么，面向对象的方法是什么呢？首先，它将数据及对数据的操作方法放在一起，作为一个相互依存、不可分离的整体——对象。对同类型对象抽像出其共性，形成类。类中的大多数数据，只能用本类的方法进行通讯。这样，程序模块间的关系更为简单，程序模块的独立性、数据的安全性就有了良好的保障。另外，通过后续章节中将介绍的继承与多态性，还可以大大提高程序的可重用性，使得软件的开发和维护都更为方便。

面向对象的方法有如此的优点，然而对于初学程序设计的人来说，是否容易理解、容易掌握呢？回答是肯定的。面向对象方法的出现，实际上是程序设计方法发展的一个返朴归真过程。软件开发从本质上讲，就是对软件所要处理的问题域进行正确的认识，并把这种认识正确地描述出来。面向对象方法所强调的基本原则，就是直接面对客观存在的事物来进行软件开发，将人们在日常生活中习惯的思维方式和表达方式应用在软件开发中，使软件开发从过分专业化的规则和技巧中回到客观世界，回到人们通常的思维方式。

1.2.2　面向对象的基本概念

现在，我们简单介绍面向对象方法中的几个基本概念。当然我们不能期望通过几句话的简单介绍就能让您完全理解这些概念，在本书的后续章节中，我们会不断帮助读者加深对这些概念的理解，以达到熟练运用。

1. 对象

从一般意义上讲，对象是现实世界中一个实际存在的事物，它可从是有形的（比如一辆汽车），也可以是无形的（比如一项计划）。对象是构成世界的一个独立单位，它具有自己的静态特征（可以用某种数据来描述）和动态特征（对象所表现的行为或具有的功能）。

面向对象方法中的对象，是系统中用来描述客观事物的一个实体，它是用来构成系统的一个基本单位。对象由一组属性和一组行为构成。属性是用来描述对象静态特征的数据项，行为是用来描述对象动态特征的操作序列。

2. 类

把众多的事物归纳、划分成一些类，是人类在认识客观世界时经常采用的思维方法。分类所依据的原则是抽像，即忽略事物的非本质特征，只注意那些与当前目标有关的本质特征，从而找出事物的共性，把具有共同性质的事物划分为一类，得出一个抽像的概念。例如，石头、树木、汽车、房屋等都是人们在长期的生产和生活实践中抽像的概念。

面向对象方法中的“类”，是具有相同属性和行为的一组对象的集合。它为属于该类的全部对象提供了抽像的描述，其内部包括属性和行为两个主要部分。类与对象的关系犹如模具与铸件之间的关系，一个属于某类的对象称为该类的一个实例。

此为试读，需要完整PDF请访问：www.ertongbook.com