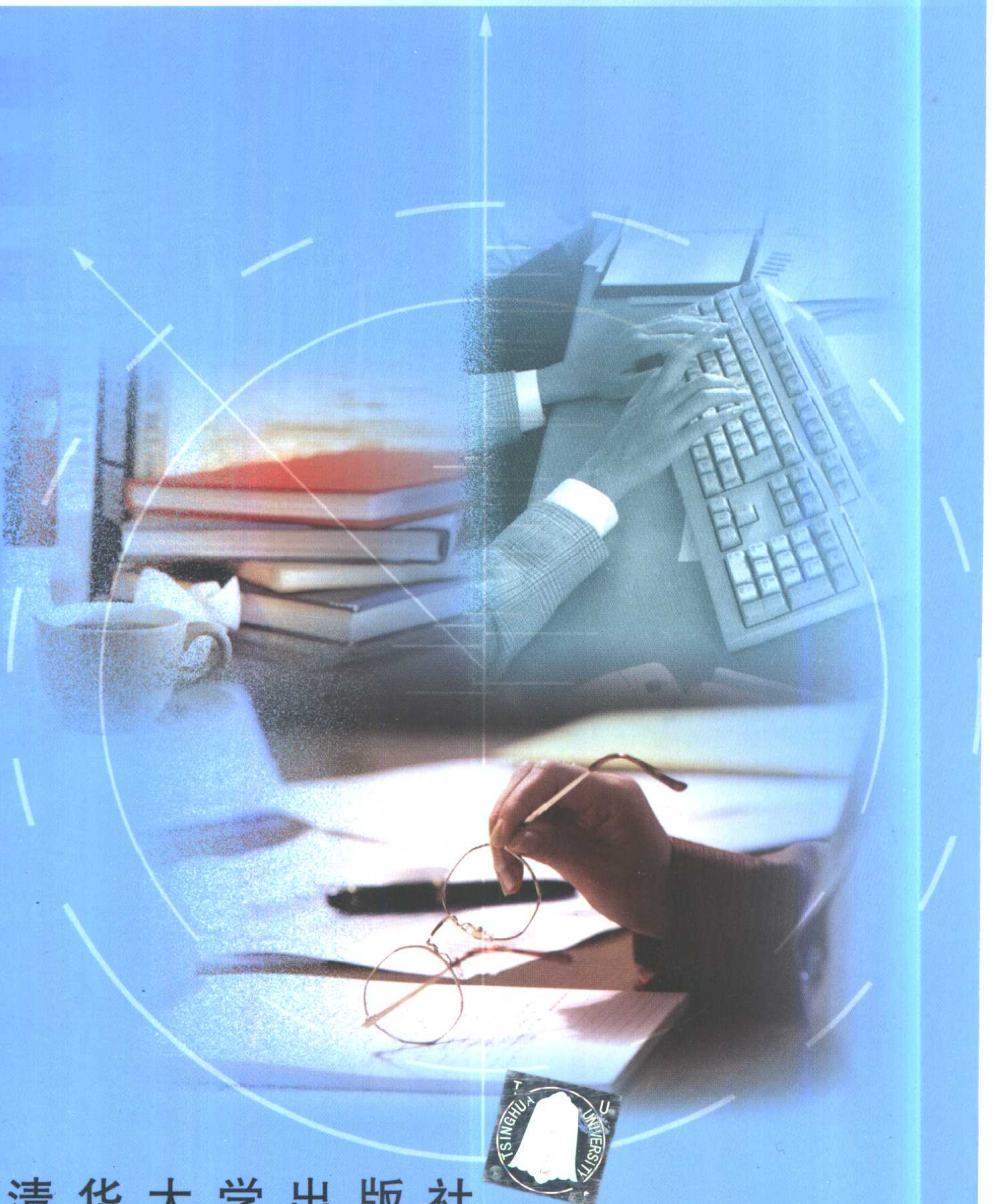
北京科海培训中心

编程之路系列教材

C++

程序设计导学



李春葆　编著

清华大学出版社
http://www.tup.tsinghua.edu.cn

北京科海培训中心

▶ 编程之路系列教材

C++语言程序设计导学

李春葆 编著

清华大学出版社

（京）新登字 158 号

内 容 简 介

本书全面讨论了 C++程序设计的有关概念，内容由浅入深逐步展开，力图使初学者易于理解。

书中精心设计了大量的例题，具体说明有关概念的应用方法。全书共 14 章，前 5 章为 C 语言基础，后 9 章是 C++语言的新增内容；最后是 8 个实习题，每个实习题分问题、要求、设计、程序和执行结果 5 个步骤。

本书强调学习过程的习题练习和上机训练。每章均有习题，所有习题都给出参考答案，便于读者参阅和模仿，以达到快速提高编程能力的目的。

本书可作为大专院校计算机专业和非计算机专业学生学习 C++语言的教材。

书　名：C++语言程序设计导学
作　者：李春葆
出版者：清华大学出版社（北京清华大学校内，邮编 100084）
印刷者：北京朝阳科普印刷厂
发行者：新华书店总店北京科技发行所
开　本：787×1092 1/16　　印张：21.625　　字数：526 千字
版　次：2002 年 1 月第 1 版　　2002 年 1 月第 1 次印刷
印　数：0001～5000
书　号：ISBN 7-302-05111-9/TP・2991
定　价：30.00 元

《编程之路系列教材》序

随着计算机技术在我国各个领域的广泛应用，以及计算机软件平台的不断提升，计算机编程不再仅限于计算机专业人员，越来越多的计算机爱好者通过专项培训或自学，已成为计算机编程的“行家里手”；特别是我国已经入世，国内IT企业及非IT企业对人才的需求将超过40万，其中一半是软件技术人才，这是传统学历教育远远满足不了的，它需要通过各种途径为这一行业的发展提供大批的IT技术人材。本丛书就是为此目的而编写的，它以计算机编程为核心，涵盖了从基础到专业应用的一些重要课目。本套丛书包括：

1. 《C++语言程序设计导学》
2. 《数据结构导学》
3. 《Visual Basic 6程序设计导学》
4. 《PowerBuilder 7.0程序设计导学》
5. 《Visual C++ 程序设计导学》
6. 《Delphi程序设计导学》
7. 《C++ Builder程序设计导学》

本丛书具有以下特点：

○ 力求通俗易懂

本丛书不仅面向计算机专业人员，更立足于计算机编程爱好者，因此，在文字叙述和内容的安排上尽量通俗易懂，力求讲出问题的来龙去脉，把编程的“过程”讲透。

○ 强化编程的概念

作为一个编程人员，必须深入领会编程的实质，这样才能做到举一反三，融会贯通，达到编制自己的应用程序的目的。所以本丛书不同于一般的软件系统使用手册，而是针对读者学习中可能遇到的问题诠释了编程思路和编程技巧，便于读者提升编程能力。

○ 编程思想与开发工具运用相结合

学习编程，不仅要在学习编程思想上有所突破，还应学会如何更好地运用编程的开发工具，只有两者的结合才是真正的理论联系实际，事半功倍的学习方法。本丛书精选了目前流行的软件开发工具（如Visual Basic，PowerBuilder，Visual C++，Delphi，C++ Builder），这些工具中提供了许多编程技巧和功能，对编程者具有实际的应用价值。

○ 内容表述与习题、实习训练并重

本丛书提供了大量的习题和实习题，而且给出了这些习题和实习题的参考答案，便于读者练习、仿效，达到快速掌握编程方法和技巧。

由于时间仓促，本书疏漏之处在所难免。但我们相信本丛书一定会成为计算机编程爱好者的良师益友。

前 言

面向对象的程序设计方法把数据和处理数据的过程当成一个整体，具有封装和数据隐藏、继承和重用以及多态性等优点，目前已经成为开发大型软件所采用的主要方法。C++语言是面向对象程序设计语言中应用最为广泛的一种。

C++是于 1986 年由 AT&T 贝尔实验室开发的。开发这一语言的目的在于通过数据封装减少程序变量的副作用，从而降低程序的复杂性并提高程序的可靠性。C++是 C 语言的直接扩展，C++的多继承机制能更好地描述对象的属性和行为。

本书全面讨论了 C++程序设计的有关概念，内容由浅入深地逐步展开，力图使初学者容易理解，而不是死记概念。书中精心设计了大量的例题，具体说明有关概念的应用方法。所有例题都在 Microsoft Visual C++ 6.0 系统中运行通过。

本书共分 14 章，前 5 章与 C 语言有所重复，后 9 章是 C++的新增内容，熟悉 C 语言的读者可跳过前 5 章。各章的内容布局如下：第 1 章介绍 C++的特点，阐述了面向对象的有关概念；第 2 章介绍 C++的数据类型；第 3 章介绍三种控制语句；第 4 章介绍函数和预处理；第 5 章介绍数组和指针；第 6 章介绍类和对象；第 7 章介绍友元函数和友元类；第 8 章介绍运算符重载；第 9 章介绍引用；第 10 章介绍函数模板和类模板；第 11 章介绍派生和继承；第 12 章介绍多态性和虚函数；第 13 章介绍 C++流和文件；第 14 章介绍异常处理。最后实习题部分给出了 8 个实习题，每个实习题包括问题、要求、设计、程序和执行结果等 5 部分。

本书强调学习过程的习题练习和实习训练。每章均有习题，所有习题都给出参考答案，便于读者参阅和模仿，以达到快速提高编程能力的目的。

本书可以作为大专院校计算机专业和非计算机专业学生学习 C++语言的教材。

本书的编写得到武汉大学计算机学院陈珉教授的大力支持，北京科海培训中心夏非彼老师也提出了许多宝贵意见，在此编者向他们表示深深的感谢。

由于编者水平所限，书中难免存在不当之处，诚恳地希望广大读者批评指正。

编 者

2001 年 10 月

目 录

第 1 章　C++概述

C++是一种广泛应用的程序设计语言，它在 C 语言的基础上扩展了面向对象的程序设计特点。最主要的是增加了类功能，使它成为面向对象的程序设计语言，从而提高了开发软件的效率。

1.1　C++的发展历史

C++源于 C 语言。1972 年至 1973 年期间，D.M.Ritchie 首创了一种新的程序设计语言，取名为 C 语言。设计 C 语言的最初目的是编写操作系统。由于其简单、灵活的特点，C 语言很快就被用于编写各种不同类型的程序，从而成为世界上最流行的语言之一。

但是，C 语言是一个面向过程的语言。随着软件开发技术的进步，程序员们最终发现，把数据和施加在其上的操作结合起来，会得到更易于理解的程序，由此产生了面向对象的程序设计思想。于是，20 世纪 80 年代初，美国 AT&T 贝尔实验室的 Bjarne Stroustrup 设计并实现了 C 语言的扩充、改进版本，最初的成果称为“带类的 C”，1993 年正式取名为 C++。C++改进了 C 的不足之处，支持面向对象的程序设计，在改进的同时保持了 C 的简洁性和高效性。

目前，C++越来越受到重视并已得到了广泛的应用，许多软件公司为 C++设计编译系统，提供不同应用级别的类库和越来越方便的开发环境，如 Microsoft 公司的 Visual C++6.0 和 Borland 公司的 Borland C++5.02 等。利用 C++设计并实现应用系统变得日益简单和快捷了。

1.2　程序设计语言和程序设计方法

1.2.1　程序和程序设计语言

程序是计算机处理对象和计算规则的描述。程序设计语言是用来描述计算机事务处理过程、便于计算机执行的规范化语言。语言的基础是一组记号和规则，根据规则由记号构成记号串的总体就是语言。

我们知道，人类自然语言（如汉语）是人们交流和表达思想的工具。那么，人与计算机如何“交流”呢？为此，就产生了计算机语言，其功能是人用计算机语言编写一系列的动作，计算机能够“理解”这些动作，按照指定的动作去执行。正是这种相同点，所以计

算机语言和自然语言都叫作“语言”。

自然语言由于其历史性和文化性，除了其语法外，还包含复杂的语义和语境，所以，人们也能理解很多不完全符合语法的语句。但计算机语言是人发明的，它主要是用语法来表达人的思想，因而在编写程序时要严格遵守语法规则。

如同人类有很多自然语言一样，计算机语言也有很多种。按照计算机历史的发展有如下几类：

- **机器语言** 它是面向机器的，是特定计算机系统所固有的语言。用机器语言进行程序设计，需要对机器结构有较多的了解。用机器语言编写的程序可读性很差，程序难以修改和维护。
- **汇编语言** 为了提高程序设计效率，人们考虑用有助记忆的符号来表示机器指令中操作码和运算数，例如用 ADD 表示加法，SUB 表示减法等。相对机器语言来看，用汇编语言编写程序的难度有所降低，程序的可读性，有所提高，但仍与人类的思维相差甚远。
- **高级语言** 汇编语言和计算机的机器语言十分接近，它的书写格式在很大程度上取决于特定计算机的机器指令，这对于人们抽象思维和交流十分不便。高级语言指的是像 Fortran、C、Pascal 和 Basic 等与具体机器无关的语言，这样程序设计者不需要了解机器的内部结构，只要按照计算机语言的语法编写程序即可。

1.2.2 结构化程序设计

出现高级语言之后，如何用它来编写较大的程序呢？人们把程序看成是处理数据的一系列过程。过程或函数定义为一个接一个顺序执行的一组指令。数据与程序分开存储，程序设计的主要技巧在于追踪哪些函数和调用哪些函数，哪些数据发生了变化。为解决其中可能存在的问题，结构化程序设计应运而生。

结构化程序设计的主要思想是功能分解并逐步求精，也就是说，当我们要设计某个目标系统时，先从代表目标系统整体功能的单个处理着手，自顶向下不断地把复杂的处理分解为子处理，这样一层一层地分解下去，直到仅剩下若干个容易处理的子处理为止。当所分解出的子处理已经十分简单，其功能显而易见时，就停止这种分解过程，对每个这样的子处理用程序加以实现。

结构化程序设计仍然存在诸多问题：生产率低下，软件代码重用程度低，软件仍然很难维护等等。针对结构化程序设计的缺点，人们提出了面向对象的程序设计方法。

1.2.3 面向对象的程序设计

面向对象程序设计的本质是把数据和处理数据的过程当成一个整体即对象。

一般认为，面向对象程序语言至少包含下面一些概念：

- **对象** 对象是人们要进行研究的任何实际存在的事物，它具有状态（用数据来描述）和操作（用来改变对象的状态）。面向对象语言把状态和操作封装于对象体

之中，并提供一种访问机制，使对象的“私有数据”仅能由这个对象的操作来执行。用户只能通过向允许公开的操作提出要求（消息），才能查询和修改对象的状态。这样，对象状态的具体表示和操作的具体实现都是隐蔽的。

- **类**　把众多事物归纳、划分成一些类，把具有共性的事物划分为一类，得出一个抽象的概念，是人类认识世界经常采用的思维方法。类是面向对象语言必需提供的用户定义的数据类型，它将具有相同状态、操作和访问机制的多个对象抽象成为一个对象类。在定义了类以后，属于这种类的一个对象叫作类实例或类对象。一个类的定义应包括类名、类的说明和类的实现。
- **继承**　继承是面向对象语言的另一个必备要素。类与类之间可以组成继承层次，一个类的定义（称为子类）可以定义在另一个已定义类（称为父类）的基础上。子类可以继承父类中的属性和操作，也可以定义自己的属性和操作，从而使内部表示上有差异的对象可以共享与它们结构有共同部分的有关操作，达到代码重用的目的。

面向对象程序设计的主要优点是：

- **与人类习惯的思维方式一致**　结构化程序设计是面向过程的，以算法为核心，把数据和过程作为相互独立的部分。面向对象程序设计以对象为中心，对象是一个统一体，它是由描述内部状态表示静态属性的数据以及可以对这些数据施加的操作封装在一起所构成的。面向对象设计方法是对问题领域进行自然分解，确定需要使用的对象和类，建立适当的类等级，在对象之间传递消息实现必要的联系，从而按照人们习惯的思维方式建立起问题域的模型，模拟客观世界。
- **可重用性好**　面向对象的软件技术在利用可重用的软件成分构造新的软件系统时有很大的灵活性。有两种方法可以重复使用一个对象类：一种方法是创建该类的实例，从而直接使用它；另一种方法是从它派生出一个满足当前需要的新类。继承性机制使得子类不仅可以重用其父类的数据结构和程序代码，而且可以在父类代码的基础上方便地修改和扩充，这种修改并不影响对原有类的使用。人们可以像使用集成电路（IC）构造计算机硬件那样，比较方便地重用对象类来构造软件系统。
- **可维护性好**　类是理想的模块机制，它的独立性好，修改一个类通常很少会牵扯到其他类。如果仅修改一个类的内部实现部分(私有数据成员或成员函数的算法)，而不修改该类的对外接口，则可以完全不影响软件的其他部分。面向对象软件技术特有的继承机制，使得对软件的修改和扩充比较容易实现，通常只要从已有类派生出一些新类，无须修改软件原有成分。面向对象软件技术的多态性机制，使得扩充软件功能时对原有代码所需作的修改进一步减少，需要增加的新代码也比较少。所以，面向对象方法设计的程序具有很好的可维护性。

正因为面向对象程序设计有众多的优点，所以，今天程序设计方法逐步由结构化程序设计发展为面向对象程序设计。

1.3 C++语言的特点

C++语言的主要特点在于它支持面向对象程序设计。下面说明它和面向对象有关的一些特征。

1. 类和数据封装

C++中的类是面向对象程序设计的基本支撑，类是数据抽象及信息隐藏的工具，对象是类的具体化。抽象对象的值和相关的操作被封装在一个类定义中。对象可被说明为给定类的变量，对象之间可以通过发送和接受消息相联系，接受消息的对象通过调用类的方法来实现相应的操作。

2. 构造函数和析构函数

类可以包含一组构造函数和一个析构函数。构造函数是在每次创建一个特定对象时由C++自动执行的类成员函数，析构函数是在特定的类的对象被撤消时自动执行的类成员函数。类的构造函数保证了在类的对象生成时可以自动进行初始化，类的析构函数保证对类的对象正常清除。

3. 访问限制符和信息隐蔽性

类的成员有：公有、私有和保护成员，它们有不同的访问限制。声明为私有成员只能由类自己的函数访问；保护成员可以由该类及其派生类的成员函数访问；公有成员构成类的界面，允许所有函数访问。通过将成员设置为具有不同的访问权限，实现了信息隐蔽。

4. 对象和消息

对象是面向对象程序设计的基本单元，通过向对象发送消息来实现对象的操作，对象根据消息的内容调用相应的方法。C++中的消息传递采用类似函数调用的机制。

5. 友元

友元是 C++面向对象的另一个重要特征。通常类的私有成员禁止该类外面的函数和类直接访问，而友元机制允许有选择地突破这种限制。只要在类定义中声明非成员函数或其他类为该类的友员函数或友元类，则这些友元就可以直接访问类的私有部分和保护部分。

6. 运算符重载和函数重载

C++允许为已有的函数和运算符重新赋予新的含义，使它们可以用于用户所希望操作的对象。运算符重载和函数重载使得我们可以以更自然的表现方式实现对象的操作，提高程序的可读性。

7. 继承和派生类

在程序中定义类时，会出现许多两个或多个类享有相似特征的情况。这时，不必在每