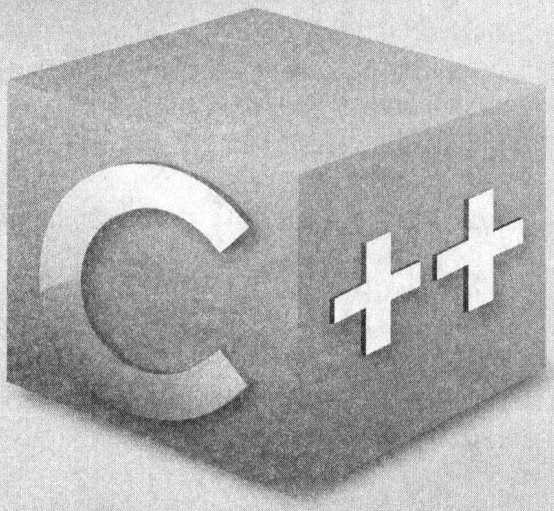
C++面向对象程序设计

李春葆　董尚燕　余云霞　编著

- **循序渐进：** 从C++基础出发，平滑过渡到面向对象程序设计方法
- **全面翔实：** 针对C++程序设计的各个知识点进行了全面、深入地剖析和提炼，构成了一个完备的知识体系
- **实例丰富：** 提供了200多个典型的程序设计示例，并给出了详尽透彻的分析过程

清华大学出版社



面向对象程序设计

李春葆 董尚燕 余云霞 编著

清华大学出版社
北 京

内 容 简 介

本书从 C++语言基础出发，平滑过渡到面向对象的程序设计方法，并针对 C++程序设计的各个知识点进行了全面、深入地剖析和提炼，为读者构建了一个完备的知识体系。

全书共分 2 部分：前 6 章讲解 C++程序设计的基础，结构化程序设计；后 6 章循序渐进地讲解面向对象程序设计的特征、概念与方法。书中强调学习过程的练习和实习训练，并设计数十个流程图来辅助阐释概念和过程，直观易懂。全书共设计了 200 多个典型的示例，并给出了详尽透彻的分析过程，便于巩固所学知识，提高程序设计能力。

本书内容翔实，实例丰富，可以作为高等院校计算机专业和非计算机专业学生学习C++语言和面向对象程序设计课程的教材和学习参考书。

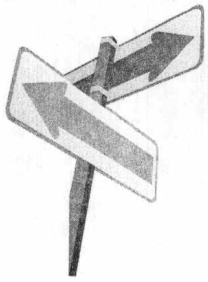
图书在版编目（CIP）数据

C++面向对象程序设计/李春葆，董尚燕，余云霞编著. —北京：清华大学出版社，2008
ISBN 978-7-302-16954-3

Ⅰ C…　Ⅱ.①李…②董…③余…　Ⅲ.C 语言－程序设计－高等学校－教材　Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2008）第 010864 号

责任编辑：夏非彼　张　楠
责任校对：贾淑媛
责任印制：孟凡玉
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　**邮　　编**：100084
c－service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：清华大学印刷厂
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：25.75　**字　数**：625 千字
版　　次：2008 年 2 月第 1 版　　**印　次**：2008 年 2 月第1 次印刷
印　　数：1～5000
定　　价：39.00 元



前言

C++是一种面向对象的程序设计语言，它提供了类、模板、函数重载和运算符重载设计等功能，充分支持抽象、继承和多态等面向对象程序设计的特征，十分方便大型软件的开发。学习C++语言，就是要掌握面向对象的思想和解决实际问题的方法。

本书分为12章。第1章是C++基础；第2章是类和对象（一），介绍类和对象的基本设计方法，包括类声明、对象定义、成员访问权限、构造函数和析构函数、静态成员等；第3章是类和对象（二），介绍类和对象的高级特征，包括常对象和常对象成员、类对象数组、子对象、嵌套类、局部类和this指针等；第4章是友元，介绍友元函数和友元类的设计方法；第5章是运算符重载，介绍各种C++运算符重载的设计方法；第6章是模板，介绍模板函数和类模板的设计方法；第7章是继承和派生，介绍派生类的设计、类层次图、继承中的构造函数和析构函数的调用次序、虚基类等；第8章是虚函数和多态性，介绍虚函数和抽象类的设计；第9章是C++流，介绍输入输出流和文件流的设计；第10章是异常处理和名字空间，介绍在C++中如何进行异常情况的处理，以及如何用名字空间进行标识符的分隔；第11章是C++标准模板库基础，介绍STL的容器、算法和迭代器的使用；第12章是面向对象软件设计，介绍面向对象软件开发的一般方法，并通过一个较大的实例说明在C++中如何进行软件开发。

本书强调学习过程的练习和实验训练。各章附有大量的练习题和上机实验题。

本书内容的布局和讲述由浅入深、层次分明，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，用以说明有关概念的应用方法，力求提高读者的程序设计思维能力。所有例题都在Microsoft Visual C++ 6.0系统中运行通过。

本书可以作为高等院校计算机专业和非计算机专业学生学习C++语言和面向对象程序设计课程的教材。

本书的编写得到武汉大学计算机学院的大力支持，特别是黄传河教授作为教学院长对教材的编写工作给予了积极的鼓励和不遗余力的推动，在此编者向其表示深深地感谢。

由于作者水平有限，书中难免存在疏漏和错误之处，恳请专家和广大读者批评指正。在学习过程中，如遇到疑难问题，可以通过以下方式与我们联系：booksaga@126.com，也可以登录图格新知网站http://www.booksaga.com留言，我们将在第一时间给予答复！

编 者

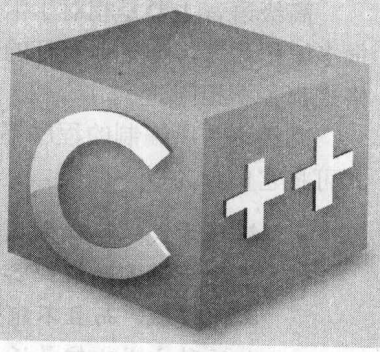
2007年10月

目录

Contents 目录

第 1 章

C++基础



C++语言本质上是带有面向对象程序设计扩展的C语言。它既支持面向对象程序设计，也支持面向过程程序设计，具有双重性。本章介绍C++的基础部分，主要是面向过程的内容，从下一章开始介绍面向对象程序设计方法。

1.1 C++概述

本节介绍计算机语言种类、程序设计方法、C++语言及其特点、C++程序的基本结构和C++程序的开发步骤等。

1.1.1 计算机语言种类

计算机语言的种类非常多，总的来说可以分成机器语言、汇编语言、高级语言三大类。

计算机每做的一次动作、一个步骤，都是按照计算机程序来执行的，程序是计算机要执行的指令的集合，而程序全部都是用某种计算机语言来编写的。所以人们欲控制计算机就一定要通过计算机语言向计算机发出命令。

计算机所能识别的语言只有机器语言，即由0和1构成的代码。但通常人们编程时，不采用机器语言，因为它非常难以记忆和识别。

目前通用的编程语言主要有两种形式：汇编语言和高级语言。

汇编语言的实质和机器语言是相同的，都是直接对硬件进行操作，只不过指令采用了英文缩写的标识符，更容易识别和记忆。它同样需要编程者将每一步具体的操作用命令的形式写出来。汇编语言的优点是用它所能完成的操作不是一般高级语言所能实现的，并且源程序经汇编生成的可执行文件不仅比较小，而且执行速度很快。但汇编源程序一般比较冗长、复杂、容易出错，使用汇编语言编程需要有更多的计算机专业知识。

高级语言是目前绝大多数编程者的选择。和汇编语言相比，它不但将许多相关的机器指令合成为单条指令，并且去掉了与具体操作有关但与完成工作无关的细节，例如使用堆栈、寄存器等，这样就大大简化了程序中的指令。同时，由于省略了很多细节，编程者也就不需要有太多的专业知识。

高级语言主要是相对于汇编语言而言，它并不是特指某一种具体的语言，而是包括了很多编程语言，如目前流行的 C/C++、Pascal、Fortran、Basic 等，这些语言的语法、命令格式都各不相同。

高级语言所编制的程序不能直接被计算机识别，必须经过转换才能被执行，按转换方式可将它们分为以下两类。

- 解释方式：解释方式类似于日常生活中的“同声翻译”，一边将应用程序源代码由相应语言的解释器“翻译”成目标代码（机器语言程序），一边执行，因此效率比较低，而且不能生成可独立执行的可执行文件，应用程序不能脱离其解释器，但这种方式比较灵活，可以动态地调整、修改应用程序。
- 编译方式：编译方式是指在应用源程序执行之前，就将程序源代码“翻译”成目标代码（机器语言程序），因此其目标程序可以脱离其语言环境独立执行，使用比较方便、效率较高。但应用程序一旦需要修改，必须先修改源代码，再重新编译生成新的目标文件才能执行，只有目标文件而没有源代码，修改起来很不方便。现在大多数的编程语言都是编译型的，例如 Visual C++、Delphi 等。

1.1.2 程序设计方法

计算机在一组指令控制下处理数据，这组指令称为计算机程序，这些计算机程序指导计算机按顺序进行计算机程序指定的一组操作。因此，用计算机求解问题就需要编写程序。

程序设计是指设计、编写和调试程序的方法与过程。由于程序是软件的本体，因此软件的质量主要通过程序的质量体现，因此，研究一种切实可行的程序设计方法至关重要。

1. 结构化程序设计

所谓面向过程是指从功能的角度分析问题，将待解决的问题空间分解成若干个功能模块，每个功能模块描述一个操作的具体过程。结构化程序设计方法就是面向过程的一个典型代表。

结构化程序设计方法的核心包括以下方面。

- 自顶向下、逐步求精的开发方法：将编写程序看成是一个逐步演化的过程。所谓自顶向下是指将分析问题的过程划分成若干个层次，每一个新的层次都是对上一个层次的细化，即步步深入，逐层细分。
- 模块化的组织方式：将整个系统分解成若干个模块，每个模块实现特定的功能，最终的系统将由这些模块组装而成。模块之间通过接口传递信息，模块划分应尽可能达到高内聚，低耦合。
- 结构化的语句结构：只使用顺序、选择和循环三种控制语言进行程序设计。

结构化程序设计的特点如下。

- 程序设计：程序是由一个个的函数组成的，函数之间通过调用而相互作用。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。
- 程序内容：由函数和函数调用构成。

结构化设计的弱点如下：

- 抽象级别较低。
- 封装性较差。
- 可重用性较低。

2. 面向对象程序设计

面向对象程序设计的方法是指用面向对象的方法指导程序设计的整个过程，所谓面向对象是指以对象为中心，分析、设计及构造应用程序的机制。

面向对象程序设计方法的核心包括以下几方面。

- 抽象：指从事物中舍弃个别的、非本质的属性，抽取出共同的、本质的属性的过程，它是形成概念的必要手段。抽象包括过程抽象和数据抽象。
- 封装：在面向对象的程序设计中，封装是指将对象的属性和行为分别用数据结构和方法描述，并将它们绑定在一起形成一个可供访问的基本逻辑单元。
- 对象：对象是用来描述现实世界中实体的部件，是面向对象软件系统在运行时刻的基本单位。为了区分属于同一个类的不同对象，每个对象都有一个惟一的标识。
- 类：类是一组具有相同属性特征的对象的抽象描述，是面向对象程序设计的又一个核心概念。类是对象抽象的结果。有了类，对象就是类的具体化，是类的实例。类可以有子类，同样也可以有父类，从而构成类的层次结构。类之间主要存在三种关系：关联、聚合和泛化。
- 消息：消息是一个对象要求另一个对象实施某项操作的请求。在一条消息中，需要包含消息的接收者和要求接收者执行哪项操作的请求，而并没有说明应该怎样做，具体的操作过程由接收者自行决定。消息传递是对象之间相互联系的惟一途径。发送者发送消息，接收者通过调用相应的方法响应消息，这个过程被不断地重复，使得应用程序在有效控制下进行计算，最终得到相应的结果。可以说，消息是驱动面向对象程序运行的源泉。
- 继承：继承是类之间的一种常见关系。这种关系为共享数据和操作提供了一种良好的机制。通过继承，一个类的定义可以基于另外一个已经存在的类。继承是面向对象程序设计方法的一个重要标志，利用继承机制可以大大提高程序的可重用性和可扩充性。

面向对象程序设计的特点如下。

- 程序设计：程序是由一个个的对象组成的，对象之间通过消息而相互作用。程序设计的主要技巧在设计哪些类以及类之间的关系。