信息与电子学科百本精品教材工程

新编计算机类本科规划教材

# C++语言程序设计

## (第2版)

吕凤翥 编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

新编计算机类本科规划教材

# C++语言程序设计

## （第2版）

吕凤翥　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

## 内 容 简 介

本书全面系统地讲述C语言和C++语言的基础知识、基本语法以及编程方法，详尽地讲述C++语言面向对象的重要特征：类和对象、继承性和派生类、多态性和虚函数等内容。本书配有丰富的例题，每章后面备有形式多样的练习题。

本书文字通俗易懂，内容由浅入深，讲解突出重点，全书偏重应用。本书适合作为高等院校理科学生学习C++语言的教材，同时也可以作为自学C++语言的指导和参考书。



**图书在版编目（CIP）数据**

C++语言程序设计/吕凤翥编著. —2版. —北京：电子工业出版社，2005.4
新编计算机类本科规划教材
ISBN 7-121-01032-1

Ⅰ.C… Ⅱ.吕… 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（2005）第021344号

责任编辑：冉 哲
印 刷：北京牛山世兴印刷厂
出版发行：电子工业出版社
北京市海淀区万寿路173信箱 邮编 100036
经 销：各地新华书店
开 本：787×1 092 1/16 印张：29.75 字数：754千字
印 次：2005年4月第1次印刷
印 数：5 000册 定价：33.80元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

# 再 版 前 言

本书作者长期从事C语言和C++语言程序设计课程的教学工作。本书是作者在总结多年来讲授C语言和C++语言的经验基础上，根据讲稿整理编写的。书中突出C++语言的重点，对其重点内容都进行了反复讲解；根据教学中学生所提出的难点，本书进行了详细讲解，并列举了实例；书中各章节中请读者回答的一些问题，多是教学中遇到的疑点。因此，突出重点、详解难点和提出疑点是本书的第一个特点。

本书的第二个特点是语言简明、概念准确、例题丰富。以通俗易懂的语言讲述C++语言的基础知识、基本规则和编程方法，以丰富的例题讲解操作方法和验证语法规则，读者可以模仿例题的程序去编写形式相似的程序和去解决内容相仿的问题。本书中例题较多，但重复性较小。每个例题都针对一种规则或一种操作，读者可以从每一个例题中学到一种方法。

本书的第三个特点是每章后边都备有较多的练习题，适合作为教材和自学参考书。每章后面的练习题内容全面，形式多样，有问答题、选择题、判断题、分析程序输出结果题和编程题等。通过这些题目，读者可以及时地检查和考核对本章内容学习和掌握的情况，老师可以从中选出一些题目留为作业题。

本书不仅较为全面地讲述C语言的主要内容，也较为系统地讲述C++语言的基本内容。通过对本书的学习，读者可以掌握C语言和C++语言的基础知识和基本规则及编程方法。本书第1章讲述面向对象的概念，揭示C语言和C++语言的关系，指明C++语言是一种使用较广的面向对象的编程语言，给出C++程序的实现方法。另外，还讲述了C++语言的词法规则。第2～7章讲述的大多是C语言的内容，同时也是C++语言的基本内容，C++程序也是建立在这些基本内容的基础上的。这些内容包括变量和常量、运算符和表达式、各种语句、函数和存储类、指针和引用、结构和联合等，在讲述过程中指出C语言与C++语言的不同。第8～12章较系统地讲述C++语言中面向对象的主要特征：封装性（类和对象）、继承性（基类和派生类）、多态性（重载和虚函数）、I/O流库及操作。这些都是C++的核心内容，从中体现C++语言面向对象的特点，这也是C++语言的重点内容。本书中的C语言部分，对于学过C语言的读者是一个很好的复习机会，从中可以搞清楚C语言和C++语言的区别；对于没有学过C语言的读者可以通过学习这部分内容，掌握C语言这个编程工具。C++语言是以C语言为基础的，掌握了C语言对学习C++语言是会有帮助的。

本书中所选用的C++程序都是在Visual C++ 5.0、Visual C++ 6.0和其他版本的C++编译系统下运行过的。

本书第一版自2001年出版以来，许多使用该书的读者提出了宝贵的意见，作者本人也使用该书作为教材，经过三年多的教学实践，感觉到原书中有些问题需要修改，有些内容需要删减和增添，部分章节内容需要调换。因此，本书第二次出版时，在保持原书章节基本不变的前提下，进行如下修改。

（1）改正了原书中出现的个别错误，在概念描述方面更加精练和准确。

（2）讲授顺序基本保持原状，只进行少量修改，修改后使得内容更加连贯。

（3）每章增添了上机指导和上机练习题。原来这部分内容放在另册中，合并后便于读者

上机实践。

本书适合作为高等院校学生的教材，也可作为教师和学生的参考书，以及广大电脑爱好者自学C++语言的指导书。

本书承蒙广大读者的关心和支持，许多读者为本书提出了宝贵的意见和建议，作者在此表示最衷心的感谢，并诚恳希望读者们继续关注本书，欢迎提供宝贵意见。

作　者

2005年2月

# 目　录

# 第 1 章　C++语言概述

**教学要点**

C++是一种面向对象的程序设计语言，它是在 C 语言基础上发展起来的。虽然它不是最早的面向对象的程序设计语言，但是它是目前使用较广泛的面向对象的程序设计语言。

本章主要讲述 C++语言编程特点，包括 C++语言与 C 语言的关系，C++语言的词法规则及 C++语言程序的实现。在讲述 C++语言之前，介绍一个与 C++语言有关的重要概念——面向对象，包括什么是面向对象，面向对象语言的特点等。通过对面向对象概念的学习和理解，可以进一步了解 C++语言的特点及其应用。

## 1.1　面向对象语言简介

本节主要讲述在计算机出现后，编程语言的发展历史。从这段历史中不难看出，面向对象语言的出现是人们对客观事物认识的不断发展的需要。因此，面向对象语言的出现是必然的。本节首先介绍面向对象这一概念。

### 1.1.1　面向对象的概念

什么是面向对象？简单地说，它和面向过程一样都是软件开发的一种方法。但是它与面向过程不同，面向对象是一种运用对象、类、继承、封装、包含、消息传递、多态性等概念来构造系统的软件开发方法。这里提出了一些新的概念，这些新的概念描述了面向对象的特点。

下面从解释这些概念中给出面向对象的特点，进而对面向对象这种方法有所了解和认识。

**1. 对象是软件系统的基本构成单位**

分析问题的出发点是对象。对象是对待解决问题（问题域）中的客观事物的抽象表示，它是面向对象程序的基本要素。

**2. 对象的属性和服务结合为一个独立的实体**

对象的属性是表示客观事物的静态特性，一般用数据来表达；对象的服务是描述客观事物的动态特性，即事物的行为，一般用函数或称方法来表达。对象的属性与服务结合为一个独立的实体，称为封装体，对外屏蔽其内部的部分特性。

**3. 类是对某些对象的抽象描述**

类是具有相同属性和服务的若干对象的集合。类为该类的所有对象提供一种统一的抽象描述。一个类中包含属性和服务两部分。实际上，类是一种类型，这种类型是自定义的，而

对象是某个类的一个实例。

#### 4. 派生类继承基类中的属性和服务

在不同的层次上运用抽象的原则，可以获得基类和它的派生类。派生类继承基类中的属性和服务，通过类间的继承关系可以简化系统的构造过程和文档，实现共享。例如，根据世界上存在的各种各样的汽车，可以将它们的共性抽象为一个汽车类，再将各种各样的轿车的共性抽象为一个轿车类。显然，轿车类应该是汽车基类的派生类，因为轿车具有汽车的共性，另外还具有它自身的特性。我们说，轿车类继承了汽车类中的属性和服务。

#### 5. 复杂对象可由若干简单对象构成

复杂的事物常可以化解为若干简单的事物。同样，一个复杂的对象可以化解为简单对象的集合。例如，描述一架飞机，这是一个较为复杂的对象，但可以将它看成由机翼、机身、发动机和尾翼等部件组成。一架飞机看作一个复杂类的对象，这个复杂类由若干简单类的对象组成。这称为类的包含关系。

#### 6. 对象与对象之间使用消息进行通信

信息是向对象发出的服务请求，信息的发送者是一个要求提供服务的对象，而信息的接收者是一个能够提供服务的对象，通过消息传送实现对象之间的通信。

#### 7. 多态性是面向对象程序设计的重要支柱

多态性是指向不同对象发送同一消息，根据对象的类的不同而完成不同的行为。多态性通过继承的机制构造对象类的结果，由函数和运算符的重载及虚函数实现类的多态性。

以上是对面向对象这一概念的理解。下面通过一个例子形象地说明封装的概念。例如，街头上的早点小吃店，在一间小屋里或小亭子里，四周封闭，通过一个小窗口对外卖货。屋或亭内有油饼、豆浆等食品，另外，还具有制作上述部分食品的服务，如炸油饼、制作豆浆等，以及收钱找钱等服务。可以将小吃店看作一个封装体，这个被封装的“实体”对外服务只通过一个“窗口”，当买早点的人向它发出“买什么早点”、“买多少”的消息后，并将钱付给它，它就将所要的食品递出。把具有这类特点的服务统称为早点服务类，而某家小吃店看成为早点服务类的一个对象。

综上所述，面向对象这种方法具有三大特性：封装性、继承性和多态性。特别是前两大特性是不可缺少的。

### 1.1.2 编程语言的发展

#### 1. 编程和编程语言

早期，人们认为编程工作包含认识事物和描述事物两项内容，认为编程就是对要解决的问题产生一个正确的认识，再用一种语言将它正确地描述出来。这样就把软件开发与编程看成是一回事了。从软件工程的角度上看，软件开发和编程是不同的。软件开发包含两项主要活动：一是人们对于要解决的问题的认识，二是对这种认识的描述。“认识”是指对要解决的问题进行周密的分析和全面的理解，并找出解决的方法；“描述”是指选用一种语言来描述对要解决问题的认识。可见，编程是在认识基础上进行的描述，编程时所选用的语言称为编程

语言。因此，我们认为编程只是软件开发中的一项内容，而不是全部内容。

开发人员对于要解决的问题的认识又称为对问题域的认识。问题域是指要解决的问题的集合，或者指要解决的问题所涉及的业务范围。人们对于问题域的认识往往是用自然语言来描述的，而计算机所能识别的却是某种编程语言。于是在自然语言和编程语言之间存在一个过渡，或称为“鸿沟”。这个“鸿沟”形成的原因很简单，就是因为机器不能识别人们描述客观事物所用的自然语言，而机器能够识别（直接或间接）的编程语言又不符合人们习惯的思维方式。于是就形成了二者之间的鸿沟。鸿沟的存在耗费软件开发人员的许多精力，同时也是许多错误的发源地。

#### 2. 编程语言的发展史

编程语言的发展是从低级到高级的，具体过程如下。

（1）机器语言

这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，因为它远离人们习惯的思维方式。

（2）汇编语言

这种语言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，符合人们的某些形象思维方式。它是低层次的抽象。计算机不能直接识别汇编语言，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。

（3）高级语言

这是一种采用命令或语句的语言，屏蔽了机器细节问题。它提高了语言的抽象层次，比汇编语言更加接近于人们的思维方式。这种语言人们容易理解和记忆，但它还与自然语言有较大差别。20 世纪 70 年代，结构化程序设计语言的出现给编程带来了方便，使得自然语言与编程语言的鸿沟进一步缩短。

（4）面向对象语言

面向对象语言是比面向过程语言更高级的一种高级语言。

面向对象语言的出现改变了编程者的思维方式，使设计程序的出发点由着眼于问题域中的过程转向着眼于问题域中的对象及其相互关系。这种转变更加符合人们对客观事物的认识，因此，面向对象语言更接近于自然语言。面向对象语言是人们对于客观事物更高层次的抽象。

从编程语言发展的历史来看，编程语言由低级向高级发展，使得自然语言与编程语言之间的鸿沟越来越窄，这就意味着软件开发人员耗费的精力越来越少，软件产品的质量越来越高。

面向对象语言的出现是人们期待填平“鸿沟”的必然结果，面向对象的程序设计方法是软件开发的新的里程碑。

### 1.1.3 面向对象语言的特点

使用面向对象的方法进行软件开发，需要用面向对象的语言进行编程。

面向对象的语言应该具有面向对象方法的特点。具体地讲，面向对象语言直接描述问题域中的对象及其相互关系。它从客观世界中所存在的事物出发，比较符合人们的思维方式。面向对象语言应具有如下特征。

① 客观世界是由一些具体事物组成的，每个事物都具有自身的静态特性和动态特性，这

些被抽象出来的特性将分别由一组属性和一组服务来描述。这种事物便是面向对象语言中的对象。

② 客观世界的事物之间有共性，也有个性。按其共性可分为若干类。类是具有共性的若干事物的集合，它是面向对象语言中相对独立的程序单位，是对某些事物的统一抽象。类是面向对象语言中的一种类型，具有类类型的事物称为对象。

③ 在客观世界中，为了简化对事物的认识和描述，往往采用继承机制。当某个特殊类的对象拥有某个一般类的全部属性与服务时，称为特殊类继承了一般类。面向对象语言中的类具有继承的关系。

④ 在客观世界中，对复杂事物的处理往往是将它化为若干简单的事物。这就是说，在一个描述复杂事物的类中，可以包含若干描述简单事物的类的对象，称为嵌套关系。即一个类的成员可以是其他类的对象，即为子对象。

⑤ 客观世界的事物是一个独立的实体，在面向对象的方法中采用封装机制，屏蔽其内部的细节，只表现出它的外部特性或行为。事物与事物之间存在着一定的联系，通过消息来表示事物之间的联系。一个对象可以通过向另一个对象发送消息来获得其服务。面向对象的语言应具有实现上述机制的功能。

总之，面向对象语言是用来实现面向对象程序设计的一种高级语言。它包含面向对象方法中所要实现的功能。

## 1.2 C 语言与 C++语言的关系

20 世纪 80 年代初，美国 AT&T 贝尔实验室设计并实现了 C++语言。C++语言对 C 语言有很大的改进，是 C 语言的扩展。它保持了 C 语言的简洁性和高效性，又克服了 C 语言中的一些不足，特别是引进了面向对象语言的要素，使得 C++语言成为一种面向对象的程序设计语言。

### 1.2.1 C++语言对 C 语言的改进

C++语言保留了 C 语言短小精简的风格，并对 C 语言的不足进行了改进。C++语言对 C 语言的改进表现如下。

① C++语言中增加了一些运算符，使其功能有所提高。例如，::，new，delete，.*，->* 等。

② C 语言是一种弱类型语言，类型转换不够严格；而 C++语言规定类型多采用强制转换，取消了对函数的默认类型，还规定函数必须用原型说明，改进了类型系统，提高了安全性。

③ 引进了引用的概念。使用引用作为函数参数，克服了使用指针带来的不便。

④ 允许函数重载，允许设置默认参数，还引进了内联函数的概念。这些措施减少了冗余性，提高了编程的灵活性和运行程序的效率。

⑤ 对变量的说明更加灵活，不受 C 语言中某些规定的限制。例如，在 C 语言程序的函数体或分程序内，必须说明语句在前，执行语句在后；在 C++程序中，可以根据需要随时定义变量。