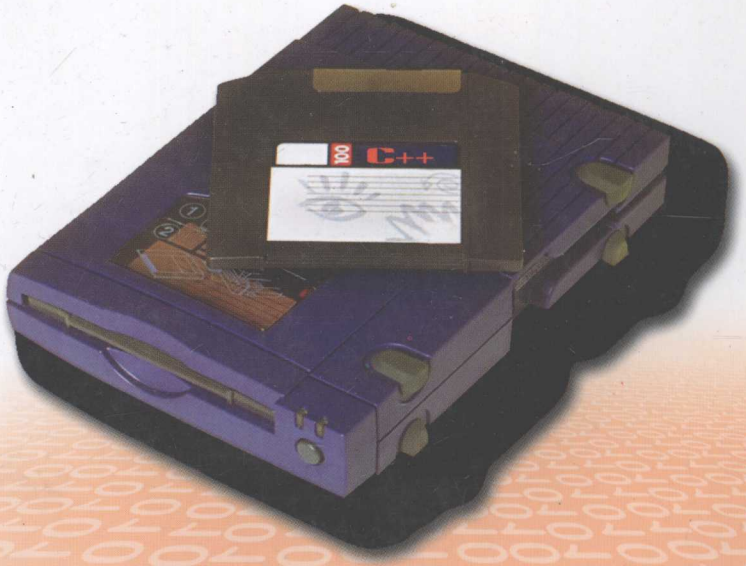
高等院校计算机系列教材

GAODENGYUANXIAO
JISUANJI
XILIEJIAOCAI

# C++与面向对象程序设计

总主编：陈火旺　主　编：傅　明　肖晓丽

湖南省计算机学会规划教材　中南大学出版社




++YMXDXCXSJ
++YUMIANXIANG
IIXIANGCHENGXUSHEJI

高等院校计算机系列教材

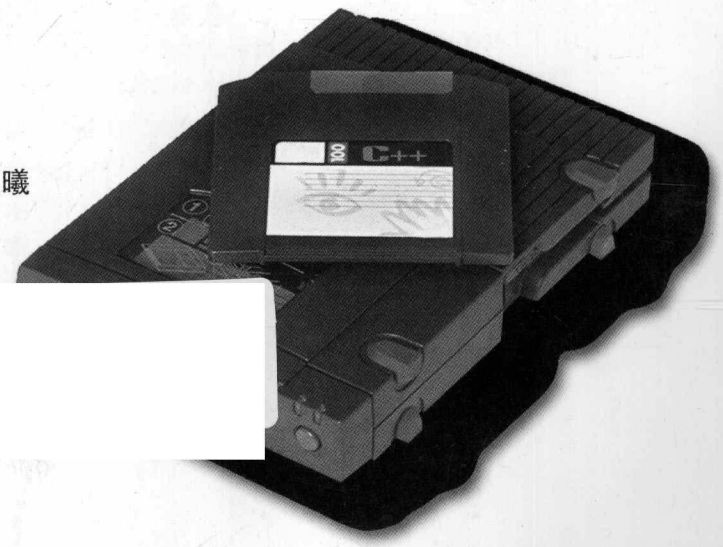


GAODENGYUANXIAO JISUANJI XILIEJIAOCAI

# C++与面向对象程序设计

总主编：陈火旺　湖南省计算机学会规划教材　中南大学出版社

主　编：傅　明　肖晓丽
副主编：王新祥　刘国英　满君丰
编　委：（按姓氏笔画排序）
刘翌南　刘　霞　陈超云　陈　曦
何建新　涂光平　夏卓群



**图书在版编目(CIP)数据**

C ++与面向对象程序设计/傅明主编. —长沙:中南大学出版社,2005.6
ISBN 7-81105-151-6

Ⅰ.C... Ⅱ.傅... Ⅲ.C语言-程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字(2005)第062685号

**C ++与面向对象程序设计**
**(含C ++与面向对象程序设计实验教程)**
傅 明 肖晓丽 主编

□责任编辑 谭晓萍
□责任印制 文桂武
□出版发行 中南大学出版社
社址:长沙市麓山南路 邮编:410083
发行科电话:0731-8876770 传真:0731-8710482
□印 装 中南大学湘雅印刷厂

□开 本 787×1092 1/16 □印张 18.75 □字数 460千字□插页:1
□版 次 2005年8月第1版 □2005年8月第1次印刷
□书 号 ISBN 7-81105-151-6/TP·016
□定 价 32.00元

# 高等院校计算机系列教材编委会

# 总　序

21 世纪，人类社会已经步入信息时代，信息产业推动着全球经济的蓬勃发展，改变着人类的联系与交换方式，从某种意义上说，信息革命是人类历史上又一次深刻的社会变革。无疑，在以信息产业为基础的知识经济社会中，计算机科学与技术具有举足轻重的地位。有鉴于此，当今世界各国皆把培养高素质的创新型计算机科学与技术专业人才作为一项重要的战略任务来抓。早在 1984 年，邓小平同志就强调指出："计算机的普及要从娃娃抓起"，从此开启了中国信息革命的征程。经过 20 多年的努力，我国的计算机教育虽然取得了令人瞩目的成就，但离知识经济社会的要求还有很大的差距。据 2005 年信息产业部的数据显示，我国的信息化人才资源指数仅为 13.43，每年短缺信息化专业人才达 100 万之多。因此，快速培养和造就一大批高素质的计算机与信息人才，乃是我国高等教育所面临的一项严峻挑战。为此，我们必须改革和完善现有计算机与信息技术学科的教学计划和课程体系，优化课程结构，精炼教学内容，拓宽专业基础，强化实践环节，注重学生的知识、能力和综合素质的培养。

为了适应计算机科学与技术学科发展和教育的需要，湖南省计算机学会，参照《中国计算机科学与技术学科教程 2002》，组织了一批长期从事计算机科学与技术专业教学与科研的学者参与编撰了这套由中南大学出版社出版的《高等院校计算机系列教材》，希望在教材中及时反映学科前沿的研究成果与发展趋势，以高水平的科研促进教材建设，以优秀教材促进教学质量的提高。该系列教材具有如下特点：

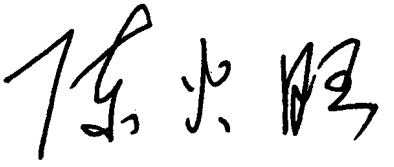
1. 教材参照《中国计算机科学与技术学科教程 2002》建议的教学大纲、知识领域、知识单元和知识点，结合作者多年教学与科研经验来编写，注重基本理论、基础知识的梳理、推演与挖掘，注意知识的更新，跟踪新技术、新成果的发展，并将之吸收到教材中来，力求开阔学生视野，逐步形成"基础课程精深，专业课程宽新"的格局，努力提高教材质量。

2. 注重理论联系实际，注意能力培养。力图通过案例教学、课堂讨论、课程实验设计与实习，训练学生掌握知识、运用知识分析并解决实际问题的能力以满足学生今后从事科研和就业的需要。

3. 在规范教材编写体例的同时，注重写作风格的灵活性：每册的每个章节包括教学目的、本章小结、思考题与练习题，每门教材都配有 PPT 电子教案，并做到层次分明、逻辑性强、概念清楚、图文并茂、表达准确、可读性强。

这套教材的编写吸纳了广大计算机科学与技术教育工作者多年的教学与科研成果，凝聚了作者们的辛勤劳动，也得到了湖南省各高等院校相关专业领导和专家的大力支持。我相信这套教材的出版，对我国计算机科学与技术专业本科教学质量的提高将有很好的促进作用。

由于编委和作者们水平与时间的限制，教材中难免还有不足之处，恳请广大读者批评指正。



**2005 年 7 月**

# 前 言

面向对象是一种重要的程序设计方法，采用这种方法的C++是当今世界上应用最广泛的程序语言之一。在所有编程语言中，C++可以说是最为复杂的。与C++有关的智力投入也是其他语言所不能比拟的。C++已经有20多年的历史了，特别是最近10年来得到了快速发展。

在1998年ISO(International Standards Organization)完成了C++的标准化后，C++便有了统一的标准，所有的编辑器都与标准兼容，这极大地提高了C++的可移植性。有了这个标准后，C++的特性也随之稳定了下来。

本书是根据《中国计算机科学与技术学科教程2002》与湖南省计算机学会组织的"高等院校计算机类系列教材"的编写协作会议精神对计算机专业教学的目标与定位、组成与分工，以及计算机专业教学的基本要求和计算机基础知识的结构而编写的。全书共分9章。第1章介绍面向对象及C++基础知识，其主要内容包括面向对象程序设计基础知识、C++基本程序结构。第2章介绍面向对象的程序设计，主要内容包括类与对象的定义、构造函数与析构函数。第3章介绍Windows应用程序，主要内容包括Windows应用程序的特点与消息驱动机制、利用MFC AppWizard创建Windows应用程序、MFC应用程序和文件、Windows消息，以及消息的发送和接受的基本过程和机制，并简要介绍了消息映射与消息处理函数。第4章介绍创建和使用对话框。主要内容包括对话框的基本原理、设计对话框资源、设计对话框类、运行对话框以及通用对话框。第5章介绍资源和资源编辑器，主要内容包括资源和资源编辑器基础知识、菜单的使用、自定义工具栏和状态栏。第6章介绍Windows标准控件，主要内容包括Windows控件概述、CStatic类控件的使用、CEdit类控件的使用以及CButton类控件的使用和CListBox类控件的使用。第7章介绍图形设备接口和CDC，主要内容包括图形设备接口(GDI)、绘图工具GDI对象CGdiObject、设备上下文(DC)、设备上下文类(CDC)、OnDraw()函数、WM-PAINT消息和获取设备上下文(DC)。这一章引导读者学习使用CGdiObject类和CDC类在视图中输出各种图形、文本的方法和技巧。第8章介绍文档和视图，主要内容包括文档/视图结构概述、文档读写。文档和视图类是MFC应用程序中最重要和最常用的两个类，因此深刻理解这两个类将会有助于应用程序的设计和实现。第9章介绍连接数据库，主要结合实例给大家介绍MFC ODBC数据库编程技术。

本书在介绍C++特性的同时，也与C语言有机地结合为一个整体，通过短小精悍的程序阐明了C++的基本概念，在深度和广度上已有较大的提升，因此教学理念、教学方法应做相应改革，建议将部分内容安排学生自学，培养学生的学习能力。

为便于教师使用本教材教学和学生学习，本书配有采用案例方式讲述并按零起点设计的辅助教材《C++与面向对象程序设计实验教程》和配套的电子教案。

本书由傅明、肖晓丽担任主编，由王新祥、刘国英、满君丰担任副主编，长沙理工大学涂光平、湖南工学院王新祥负责了本书的大部分编写工作。参加编写工作的还有：长沙理工大学刘翌南、陈曦、夏卓群以及湖南城市学院何建新、中南林学院陈超云、南华大学刘霞，在此一并致谢！

由于新教材的知识面较广，要将众多的知识很好地贯穿起来，难度较大，不足之处在所难免。为便于以后教材的修订，恳请专家、教师及读者多提宝贵意见。

编　者

2005年4月

# 目 录

# 第 1 章　面向对象及 C++ 基础知识

Visual C++ 应用程序是采用 C++ 编写的。C++ 语言在 C 语言的基础上进行了改进与扩充，是一种既面向对象又面向过程的混合程序设计语言。

本章介绍了面向对象程序设计方法的基本概念，并在 C 语言的基础上对比介绍了 C++ 的基本语法，使读者能够对 Visual C++ 应用程序开发中遇到的语法现象有初步的理解。

## 1.1　面向对象程序设计基础知识

### 1.1.1　面向过程与面向对象

现代计算机是面向算法的自动机，算法通过程序告诉计算机该如何完成工作。构成程序的符号系统是语言，它是描述算法的编程工具。计算机语言(程序设计语言)的发展过程就是其功能不断完善、描述问题的方法更加接近人类思维方式的过程。在计算机语言的发展过程中，新技术和新思想也不断出现。

高级语言起始于 20 世纪 50 年代末期，它屏蔽了机器的细节，提高了语言的抽象层次，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。60 年代中期，FORTRAN，COBOL，LISP 和 ALGOL 相继出现，这些语言又称为面向过程的语言。所谓“面向过程”，就是不必了解计算机的内部逻辑，而是把精力主要集中在解题算法的逻辑和过程的描述上，使用过程语言编写程序，通过程序把解决问题的执行步骤告诉计算机。

结构化程序设计思想产生于 20 世纪 60 年代，它的出现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 20 世纪 70 年代到 80 年代，结构化程序设计成为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。

虽然结构化程序设计方法具有很多优点，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。

面向对象方法的产生，是计算机科学发展的要求。20 世纪 80 年代，特别是 90 年代以来，软件的规模进一步扩大，对软件可靠性和代码可重用性的要求也进一步提高。就是在这样的背景下，面向对象的程序设计方法应运而生。

### 1.1.2 面向对象程序设计方法

什么是面向对象的方法呢？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。对同类型对象抽象出其共性，形成类。类中的大多数数据，只能用本类的方法进行处理。类通过一个简单的外部接口与外界发生关系，对象与对象之间通过消息进行通信。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。同时，也提高了程序的可重用性，使得软件的开发和维护都更为方便。

面向对象方法所强调的基本原则，就是直接面对客观存在的实物来进行软件开发，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方法、规则和技巧中回到客观世界，回到人们通常的思维方式中来。

### 1.1.3 面向对象的基本概念

现在，我们简单介绍一下面向对象方法中的几个基本概念。

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实物，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

把众多的实物归纳、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象，即忽略事物的非本质特征，只注意那些与当前目标相关的本质特征，从而找出事物的共性，把具有共同性质的事物划分为一类，得出一个抽象的概念。

面向对象方法中的“类”，是具有相同属性和服务的一组对象的集合。它为属于该类的全部对象提供了抽象的描述，其内部包括属性和行为两个主要部分。一个属于某类的对象称为该类的一个实例。

3. 封装

封装是面向对象方法的一个重要原则，就是把对象的属性和服务结合成一个独立的系统单位，并尽可能隐蔽对象的内部细节。这里有两个含义：第一个含义是把对象的全部属性和全部服务结合在一起，形成一个不可分割的独立单位。第二个含义也称为“信息隐蔽”，即尽可能隐蔽对象的内部细节，对外形成一个边界，只保留有限的对外接口使之与外部发生联系。

4. 继承

继承是面向对象技术能够提高软件开发效率的重要原因之一，其定义是：特殊类的对象拥有其一般类的全部属性与服务，称为特殊类对一般类的继承。

继承具有重要的实际意义，它简化了人们对事物的认识和描述。比如我们认识了轮船的特征以后，再考虑客轮时，因为知道客轮也是轮船，于是可以认为它理所当然地具有轮船的

全部一般特征，从而只需要把精力用于发现和描述客轮独有的那些特征。

继承对于软件复用有着重要意义，使特殊类继承一般类，本身就是软件复用。不仅如此，如果将开发好的类作为构件放到构件库中，在开发新系统时便可以直接使用或继承使用。

5. 多态性

多态性是指在一般类中定义的属性或行为，被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或行为在一般类及其各个特殊类中具有不同的语义。例如，我们可以定义一个一般类“几何图形”，它具有“绘图”行为，但这个行为并不具有具体含义，也就是说并不确定执行时到底画一个什么样的图（因为此时尚不知道“几何图形”到底是一个什么图形，“绘图”行为当然也就无从实现）。然后，再定义一些特殊类，如“椭圆”和“多边形”，他们都继承一般类“几何图形”，因此也就自动具有了“绘图”行为，使之分别实现画椭圆和多边形的功能。进而，还可以定义“矩形”类继承“多边形”类，在其中使“绘图”行为实现绘制矩形的功能。这就是面向对象方法中的多态性。

## 1.2　C++ 基本程序结构

### 1.2.1　新的风格

让我们从比较一个简单的 C 程序的例子入手，介绍 C++ 语言的风格。

1. 新的输入和输出风格

【例 1.1】　编写一个程序，输入一个人的姓名，然后输出“HELLO，＊＊＊!”

我们先用 C 语言来编写这个程序，源程序如下：

```
#include <stdio.h>
main()
{
  char *name;
  printf("%s", "Please input your name:");
  scanf("%s", name);
  printf("Hello, %s! \n", name);
}
```

C 语言的输入/输出通过函数来实现，使得 C 语言更加精练。但对于不同类型的数据又采用不同的控制字符，使人很难记住。

下面我们用 C++ 语言来编写一个功能完全与此相同的程序。

```
#include <iostream.h>
void main()
{
  char *name;
  cout << "Please input your name:";
  cin >> name;
```

```
    cout < <"Hello," < <name < <"!" < <endl;
}
```

通过比较上面的两个程序，我们可以看出，C++语言在输入/输出方面对C语言作了比较大的改进。

C++语言的输入/输出，是通过流来实现的。C++是自带输入和输出流的，并且可以根据数据的类型，自动使用合适的输出方式。

现在再来看我们的C++程序。在第一行，包含了一个头文件iostream.h。有了它，我们就可以使用C++风格的输入和输出。这与C语言的道理一样，只是C语言的头文件是stdio.h。

第五行的"cout < <"，是告诉计算机，把后面的内容送到标准输出设备。同样，第六行的"cin > >"是告诉计算机，把标准输入设备接收到的数据，存入后面的变量。如果把"< <"和"> >"看作表示方向的符号，我们就会发现，数据确实是在按照一定的方向流动，这就是"流"这个名称的由来。

虽然C风格的输入输出在C++中也是允许的，但是，C++风格的输入/输出更方便。

2. *灵活的注释方式*

C++提供了一种新的注释方式：从"//"开始，直到行尾，都将被计算机当作注释。例如：

```
i++;          //i = i + 1
```

另一方面，C风格的多行注释在C++中也仍然可以使用。一般情况下，多行注释仍旧使用"/*…*/"，而短的注释则较多使用行注释方式"//"。

3. *告别宏定义*

在C语言中，宏定义是一个重要的内容。无参数的宏作为常量，而带参数的宏则可以提供比函数调用更高的效率。在C++中，由于const修饰符和内联函数的引入，无论是带参数的宏还是不带参数的宏，都失去了存在的必要。也就是说，他们可以"光荣退休"了。

首先我们来看const修饰符。用类型修饰符const声明的变量只能被读取。该变量必须声明时被定义(即初始化)，并且它的值在程序中不能被改变。如果在程序中企图改变这个变量的值，则是非法的，会出现编译错误。

**【例1.2】** 输入圆的半径，输出圆的面积和周长。

```
#include <iostream.h>
const float pi = 3.1416;
void main()
{
  float r, s, c;
  cout < <"r = ";
  cin > >r;
  s = pi * r * r;
  c = 2 * pi * r;
  cout < <"s = " < <s < <endl;
  cout < <"c = " < <c < <endl;
}
```

const修饰符的使用很简单。事实上，对基本数据类型的变量，一旦加上const修饰符，

编译器就将其视为一个常量，不再为它分配内存，并且每当在程序中遇到它时，都用在说明时所给出的初始值取代它。使用const可以使编译器对处理内容有更多的了解，从而允许对其进行类型检查，同时还能避免对常量的不必要的内存分配并可改善程序的可读性。

const还可以修饰指针变量，有以下三种情况：

(1)const int *p；这表明p是一个指针，它只能指向一个被const修饰的int类型的变量，即只能指向一个整型常量，例如：

```
const int *p;
const int var=555;
p=&var;
```

(2)char const *p；这表明指针p本身是常量，即p指向一个固定的char类型的地址，而p的内容却是可以修改的，例如：

```
char const *p[4]={"a","b","c","d"};
p[2]="m";
```

再例如：

```
char const *p="adcd";
p="mnop";
```

(3) const char *const p="abcd"；这里指针和它所指的内容都是常量，必须对指针进行初始化，且对任何一个进行修改都是错误的。

4. *使用函数原型和缺省参数*

在C++中，函数原型是一个很重要的概念。任何函数，如果缺少了函数原型，C++都将无法编译。函数原型使得C++能够提供更强的类型检查，将函数调用表达式中可能存在的问题发现在编译阶段。

函数原型标识一个函数的返回类型，同时也标识该函数参数的个数和类型。C++编译器从一个函数定义中抽取该函数的函数原型。程序员也可在程序中使用函数说明语句来说明一个函数的原型。函数说明语句一般形式为：

类型　函数名(参数类型说明列表)；

其中“列表”是用逗号隔开的一个类型说明，其个数和指定的类型必须和函数定义中的一致。例如：

```
int sum(int , int);
```

在函数说明中也可以给出参数名，例如：

```
int sum(int first, int second);
```

名字first和second对编译器没有意义，但如果取名恰当的话，这些名字可以起到隐含参数用途的作用，以帮助程序员正确掌握函数的使用方法。

C++语言的实现者为我们考虑得非常周到。一方面，他们总是在设法减少编码的复杂程度，另一方面他们又使得编译器能检查出更多的错误。在函数调用时，他们引进了一种新类型的参数：缺省参数。缺省参数就是不要求程序员设定该参数，而由编译器在需要时给该参数赋予预先设定的值。当程序员需要传递一个与预先设定不同的值时，必须显式地指明。缺省参数是在函数原型中说明的。例如：

```
int SaveName(char *name, char *last_name="");
```