21世纪高等学校计算机科学与技术规划教材

C++

面向对象程序设计

主　编　董正言
副主编　张　聪

C++ MIANXIANG DUIXIANG CHENGXU SHEJI

北京邮电大学出版社
www.buptpress.com

21世纪高等学校计算机科学与技术规划教材

C++面向对象程序设计

主　编　董正言
副主编　张　聪

北京邮电大学出版社
·北京·

内 容 简 介

本书以面向对象程序设计的本质特征“抽象、封装、继承和多态”为主线，以C++语言为基础，全面地阐述了面向对象程序设计的基本原理和技术。全书共分10章，前3章用于介绍C++语言的基本编程技术，包括数据类型、常用运算符、结构化控制语句和函数。第4章～第7章为面向对象程序设计的核心内容，包括类和对象、类的继承和多态。第8章介绍C++语言的程序结构、编译预处理和命名空间等内容。第9章、第10章分别介绍C++语言的输入/输出操作和面向对象程序设计方法的异常处理机制。

本书语言简洁、流畅，内容通俗易懂、重点突出，对核心和重点内容都佐以大量例证。

本书既可以作为高等院校计算机科学与技术及相关专业“面向对象程序设计”课程的入门教材，也可作为广大C++语言爱好者和开发人员的参考书。

图书在版编目(CIP)数据

C++面向对象程序设计/董正言主编. --北京：北京邮电大学出版社，2010.1
ISBN 978-7-5635-2169-2

Ⅰ.①C… Ⅱ.①董… Ⅲ.①C语言—程序设计—高等学校—教材 Ⅳ.①TP312

中国版本图书馆CIP数据核字(2010)第006413号

书　　名 C++面向对象程序设计
主　　编 董正言
责任编辑 沙一飞
出版发行 北京邮电大学出版社
社　　址 北京市海淀区西土城路10号(100876)
电话传真 010-62282185(发行部)　010-62283578(传真)
电子信箱 ctrd@buptpress.com
经　　销 各地新华书店
印　　刷 北京忠信诚胶印厂
开　　本 787 mm×1 092 mm　1/16
印　　张 18.5
字　　数 403千字
版　　次 2010年1月第1版　2010年1月第1次印刷

ISBN 978-7-5635-2169-2　　定价：32.00元

如有质量问题请与发行部联系

前　言

面向对象程序设计技术是当前主流的程序设计技术。与传统的面向过程程序设计技术相比，面向对象程序设计技术具有明显的优势，这种优势主要体现在以下几个方面：

①传统的面向过程程序设计技术忽略了数据和操作之间的内在联系，程序中的数据和操作方法相分离。而面向对象程序设计技术将程序要处理的数据和处理方法封装在一起，构成一个统一体——对象。在编程过程中通过使用对象模型来建模现实世界中的事物，这样就使解空间模型的结构和问题空间模型的结构相一致。使用面向对象的方法解决问题的思路更加符合人类一贯的思维方法。

②面向对象的封装技术为程序提供了更好的数据安全性。

③面向对象的继承技术为程序提供了更好的可重用性。

④面向对象的多态技术使程序具有了更好的可扩展性。

⑤和传统的面向过程程序设计技术相比，面向对象的程序设计技术更适合开发大型的图形界面应用程序。

目前，常用的面向对象编程语言主要有 C++、Java、C# 等。其中，C++语言是在 C 语言的基础上发展演变起来的一种面向对象的程序设计语言。它既继承了 C 语言面向过程程序设计方法的特点，同时又支持面向对象的程序设计方法。由于其高效性、灵活性和适应性而广为应用，目前已经成为软件开发的最重要语言之一。

本书以 C++语言为基础，阐述了面向对象编程的基本原理。为了使读者能够更加透彻地理解面向对象编程的原理和方法，本书大量地使用了自编的例程。全部例程的源代码均使用 Visual C++ 6.0 编写，并编译通过。

本书在编写过程中力求采用读者容易理解的方法，深入浅出、通俗易懂，是一本面向高等院校计算机科学与技术及相关专业本、专科学生学习面向对象编程技术的入门教材，也可作为广大 C++语言爱好者和开发人员的参考书。

本书由董正言任主编，张聪担任副主编。另外，刘文涛、阮灵、方元、曹琳琅、冯锋等也参与了本书的部分撰写工作。

由于学识水平和时间的限制，疏漏和不妥之处在所难免，敬请批评指正。

编　者

目　录

第1章 序 论

1.1 编程语言的发展

计算机由硬件和软件组成。硬件是指组成计算机的各种零部件，如中央处理器(CPU)、主存储器(内存)、硬盘等；软件是指挥硬件运行的程序，如操作系统、各种硬件的驱动程序和其他应用程序等。程序设计语言是用来设计计算机软件的。自1946年第1台电子数字计算机诞生至今，计算机科学发展迅猛，程序设计语言也经历了漫长的发展过程。

第1代程序设计语言是机器语言。机器语言是计算机硬件可以识别的机器指令，由二进制编码组成。不难想象，采用机器语言编写程序的难度很大，只有极少数掌握计算机硬件结构的专业人员才能胜任。

不久后，出现了汇编语言。汇编语言就是用一些容易记忆的助记符代替机器语言中的二进制编码，使机器指令看上去更容易理解。常用的助记符有MOV、ADD、SUB等。计算机在运行由汇编语言编写的程序时，首先需要把程序翻译成计算机能够识别的机器语言，完成这个翻译功能的程序叫做汇编程序。因为不同类型的计算机具有不同的硬件结构和指令系统，所以，由汇编语言编写的程序不是平台独立的，也就是说，为一种计算机编写的汇编程序必须经过修改，才能在另一种类型的计算机上运行。

由于机器语言和汇编语言都能直接操作计算机硬件，所以被称为低级语言。

20世纪60年代末，出现了结构化高级编程语言。高级语言采用数量有限的、容易理解的执行语句编写程序，并允许程序员用具有一定含义的名字为数据命名。高级语言致力于解决问题，而不针对特定的硬件，屏蔽了计算机的硬件细节。采用高级语言编写的程序不需修改就可以在不同类型的计算机上运行，程序的可移植性高。计算机在运行高级语言编写的程序时，首先要用一个翻译程序把高级语言翻译成硬件可识别的机器语言，这个翻译程序叫做编译程序。常见的结构化高级语言有C、FORTRAN、Pascal、BASIC等。采用高级语言进行结构化程序设计的原则是：自顶向下，逐步求精。

进入20世纪90年代，随着图形界面操作系统(如微软的Windows操作系统)的普及，采用结构化程序设计方法设计大型的图形界面应用程序显得力不从心。这时，一种新型的程序设计方法——面向对象的程序设计方法逐步取代了结构化程序设计方法，成为大型程序设计的主流技术。同时，面向对象的程序设计语言也取代了结构化高级语言，成为主流的编程语言。在面向对象的程序设计方法中，把数据和操作它们的方法封装在一起，抽象出解域空间对象模型，用于描述实际的问题域空间对象，更直接地反映了客观世界中的事物及它们之间的关系；并且，利用继承和多态技术，极大提高了程序代码的重用性和程序设计效率。当前常见的面向对象程序设计语言有C++、Java等。

1.2 C++语言简介

C++语言是由C语言进化、发展而来的，C语言于20世纪70年代诞生于美国的贝尔实验室。C语言的特点是：使用简洁、灵活，数据类型和运算符丰富，具有结构化控制语句，既能像汇编语言一样直接访问存储器的物理地址、管理通信端口和磁盘驱动器，又具有结构化高级语言的特点——程序具备良好的可读性和可移植性。总之，C语言是一种使用简单灵活、功能强大的结构化程序设计语言。由于它同时具有低级语言和结构化高级语言的特点，也被一些人叫做中级语言。C语言从诞生时起，就以其使用简单灵活、功能强大的特点赢得了多数程序员的青睐，成为20世纪80年代占据统治地位的编程语言。C语言不支持面向对象的程序设计(OOP)。

C++语言诞生于20世纪80年代，和C语言相同，也是由美国的贝尔实验室开发的。C++由C发展演变而来，一方面C++继承了C语言使用灵活、功能强大的特性；同时C++也支持面向对象的程序设计，因此很多人把C++称为“带类的C”。其实，和C相比，C++不光是增加了对面向对象程序设计的支持，在其他很多方面，C++都对C进行了改进和优化。

为了提高C++程序的可移植性，美国国家标准局(ANSI)和国际标准化组织(ISO)于1990年建立了联合组织ANSI/ISO，负责制定C++标准。1998年发布了C++标准第1版(ISO/IEC14882：1998)，2003年发布了C++标准第2版(ISO/IEC14882：2003)。

C++数据类型和运算符丰富，具有结构化的控制语句，能直接访问和控制硬件，同时支持面向对象程序设计，是一种使用简单灵活、功能强大的面向对象的程序设计语言。

1.3 面向对象的程序设计方法

早期的计算机主要用于科学计算，随着计算机技术的发展，计算机的应用领域不断扩大，要解决问题的复杂程度不断提高，计算机程序的规模日益增大，大型程序的修改和维护工作越来越困难，程序的维护成本日益增高，甚至远远高于程序的开发成本。程序结构的混乱是造成这些现象的根本因素之一。

20世纪60年代产生的面向过程的结构化程序设计思想，为使用计算机开发和维护复杂程序提供了有利的手段。那么，什么是面向过程的程序设计方法呢？面向过程的结构化程序设计方法简单地说就是：采用自顶向下、逐步求精的思路，对复杂的问题进行层层分解，逐步将其按功能划分成相对简单的、规模较小的问题；应用程序由许多较小的模块组成，模块间的联系尽可能简单；每个模块完成某种独立的功能，其内部均由顺序、选择、循环3种基本结构组成。面向过程的模块化和结构化特点，极大提高了程序的开发效率，优化了程序结构，增强了程序的可维护性。20世纪70年代到20世纪80年代，这种方法在程序设计领域占据主导地位。

如上所述，面向过程的程序设计方法具有很多显著的优点，而其缺点也是显而易见的。我们知道，现实世界中的事物都具有两种属性，即静态属性和动态属性。例如，对于一个人

来说，姓名、身高、体重、视力等属性是静态属性，而走路、吃饭、睡觉等是动态属性；对于一辆汽车而言，品牌、价格、载重量、速度等是静态属性，而行驶、转弯、刹车等是动态属性。在面向过程的程序设计中，如果要对某个事物建模，通常使用一组数据来描述该事物的静态属性，而用一组函数来描述该事物的动态属性。而数据和函数是相互独立的，即程序中的数据和操作它们的方法分离，当需要修改程序中的某些数据的结构时，那么使用了该数据的所有的函数都要进行相应的修改。过程化程序设计的这种特点给程序的维护带来不便。进入20世纪90年代，Windows等图形界面操作系统取代了DOS等命令流操作系统成为主流操作系统，采用过程化的程序设计方法开发和维护大型的图形界面应用程序显得力不从心，面向对象的程序设计逐渐取代面向过程的程序设计，占据了主导地位。

面向对象的程序设计方法就是通过抽象，把数据和操作它们的方法一起封装在一个叫做类的抽象数据结构中，以描述现实世界中的某种事物。类的一个对象对应于现实世界中事物的一个具体个体；类中的数据用于描述事物的静态属性；类中的方法用于描述事物的动态属性。类中的大部分数据只能由该类的方法进行处理；类提供简单的接口和外部发生关系；对象与对象之间通过互发消息进行通信。与强调算法的过程化编程不同，面向对象程序设计强调的是数据。过程化程序设计总是试图使问题满足语言的过程性方法；而面向对象的程序设计是试图让语言来满足实际问题的要求。

从软件工程的角度来看，面向对象程序设计方法具有以下几个显著优点：

(1)更好的模块化

面向对象程序设计方法中的对象就是模块，它把数据和操作这些数据的方法封装在一起构成程序模块。

(2)更高级的抽象性

面向对象的程序设计方法不仅支持过程抽象，而且支持数据抽象。面向对象程序设计方法中的类就是一种抽象数据类型，它描述某一类对象共有的属性，它是定义对象的模板，而对象是类的具体实例。此外，某些面向对象的程序设计语言(如C++)还支持参数化抽象，即把类成员的数据类型参数化。更高级的抽象性使程序模块的可重用性更高。

(3)更好的信息隐藏性

在面向对象的程序设计方法中，允许为类的成员设置访问权限，这样就可以将数据隐藏在对象之中，类的用户只能通过类的公有接口来访问类的对象。

(4)低耦合、高内聚

在面向对象程序设计方法中，一个对象就是一个独立的单元，其内部各元素用于描述对象的本质属性，它们之间联系紧密、具有高度的内聚性。而不同类的对象之间或者同一个类的不同对象之间，只能通过外部接口发送消息来相互通信。如果一类对象的内部结构发生变化，只要它的外部接口保持不变，就不会影响其他的程序模块(其他的类和对象)，所以面向对象程序模块之间具有更低的耦合性。低耦合性和高内聚性符合软件工程的模块独立原理，是软件重用的基础和保证。

(5)更好的可重用性

继承性和多态性是面向对象程序设计方法的两个本质属性。通过继承可以从已有的类派生出新的类，新的类继承了原有类的全部属性，是对原有类的扩展。而多态性是指给不同的对象发送相同的消息，会产生不同的行为。

面向对象程序设计方法的封装性、高度抽象性、低耦合和高内聚性、继承性和多态性都使其具有更好的代码可重用性。

本书将紧密围绕抽象、封装、继承和多态等特性来阐述面向对象的程序设计原理。

1.4 一个简单的C++程序

使用面向对象的方法编写C++应用程序时，首先必须对程序的功能进行分析，从问题域的对象模型抽象出解域的对象模型，并将其封装成类；或从已有的类通过继承派生出新的类，作为解域的对象模型。对于比较简单的问题，如果使用标准C++类库中已有的类就可以解决问题，则不需要创建新的类，而对于规模较大和复杂的问题，则需要创建或派生多个新的类。

在创建类的过程中，不仅要设计合理的数据结构来描述对象的静态属性，还要设计正确的算法来描述对象的动态属性，以实现对象的功能。不同类的对象之间可以通过发送消息（调用成员函数）相互通信。在程序的主函数中，不同类的对象相互协作实现程序的功能。

现在我们通过一个简单的C++程序，学习程序的创建步骤。

例1.1 一个简单的C++程序。

程序代码如下：

```
#include<iostream>
using namespace std;
void main()
{
    cout<<"Hello!\n";
    cout<<"Welcome to C++!!!\n";
}
```

程序的第1行“#include<iostream>”是一条预编译指令，功能是在程序编译前，用头文件iostream中的内容取代该指令，嵌入到该指令所在的地方。头文件iostream是C++系统提供的库文件，其中声明了完成输入/输出操作的函数库和类库。因为本程序要向控制台输出信息，所以要包含iostream头文件。

程序的第2行中的using是一条编译指令，namespace是C++的关键字，std是命名空间的名称，命名空间中可以包含类库、函数库、对象等。这条语句的功能是使程序中可以使用包含在命名空间std中的所有元素。头文件iostream就包含在命名空间std中。

第3条语句是main函数的函数头。函数是C++程序中最小的、功能独立的单位。任何C++程序都必须有而且仅有一个名为main的函数。任何程序都从main函数的第1条语句开始执行，main函数是程序的入口点。main之前的void表示函数没有返回值。函数名后面的圆括号中可以放置函数的形式参数列表。本例中的main函数没有任何参数。

第4行～第7行是由一对花括号括住的函数体，函数体是若干条语句的集合，每条C++语句都要以一个分号(;)作为结束标志。本例的main函数中包含两条语句。cout是一个在

头文件 iostream 中预定义的输出流类的对象，代表计算机的标准输出设备(显示器)。cout 后面的<<是一个插入操作符，功能是把紧随其后的双引号中的字符串输出到显示器上。程序运行结果如图 1-1 所示。

图 1-1 程序运行结果

一个由高级语言编写的程序从开始编码到可以运行需要经过以下 3 步。

(1)编辑程序

可在普通的文本编辑器(如 Windows 记事本)或一些专业开发软件(如 Visual C++ 6.0 或 Tubor C++等)提供的编辑器中对程序进行编码。这种由高级语言编码的程序称为源程序。

(2)编译

使用编译程序对源程序进行编译。目的是将由高级语言编写的源程序翻译成计算机硬件可以识别的机器指令。

(3)链接

这一步是使用链接程序对源程序进行链接。目的是把程序中调用的所有库函数都链接到程序之中，和源程序一起创建一个目标程序。目标程序可以是一个可执行文件，也可以是一个其他格式的文件。

目前有很多软件支持 C++程序的开发，如 Visual C++、Borland C++、Watcom C++、Tubor C++等。本书中的程序都是采用 Visual C++ 6.0 编写的。本书最后的附录详细介绍了如何创建并执行一个 C++应用程序。

通过上面的例子，我们看到，即使是创建一个简单的程序，一般也要分为分析、编码、编译、链接等多个步骤。如果程序的运行结果存在错误，还需要进行调试和修改。

1.5 小结

程序设计语言是编写程序的工具，可分为机器语言、汇编语言、面向过程的高级语言和面向对象的高级语言。C++语言是由 C 语言发展、进化而来的面向对象的程序设计语言。

面向对象的编程方法在当今的程序设计领域占据主导地位。面向对象编程方法的基本特征是：抽象、封装、继承和多态。

创建一个程序，至少需要经过编码、编译、链接等步骤。

习 题

1. 面向对象程序设计方法有哪些基本特征？
2. 从着手开始编写一个程序，到形成一个可执行文件，通常需要经过哪几个步骤？
3. C++程序的入口点是什么？

第2章　数据处理和控制语句

计算机程序最基本的功能是处理数据。目前计算机应用已经遍及社会生活的各个领域，人们使用计算机来处理各式各样的问题，计算机对这些问题的处理归根结底都要转化为对数据进行处理。作为一种功能强大的编程语言，C++提供了丰富的数据类型和运算符。本章主要介绍C++基本数据类型、复合数据类型、运算符和控制语句。

2.1　基本概念

本小结先给出一个程序实例，然后介绍编程的相关基本概念。

2.1.1　程序实例

例2.1　运行下面的程序。

程序代码如下：

```
#include<iostream>
using namespace std;
int j=1000;                                    //j是一个全局变量
void main()
{
    cout<<"您好！现在我们开始学习第2章"<<endl;
    int i;                                     //i是一个整型变量
    i=10;
    const float f=10.25;                       /*f是一个实型常量*/
    cout<<"j是一个全局变量\nj="<<j<<endl;
    cout<<"i="<<i<<endl;
    cout<<"i是整型变量,i的值可以改变!\n";
    i=100;
    cout<<"i="<<i<<endl;
    cout<<"f是实型常量,f的值不能改变\n";
    cout<<"f="<<f<<endl;
}
```

程序的运行结果如图 2-1 所示。

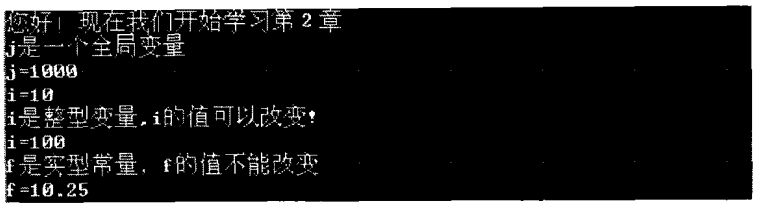

图 2-1　程序运行结果

2.1.2　C++字符集

字符集就是在程序设计中除字符数据外可以使用的全部字符的集合。C++语言的字符集由以下字符构成：

- 大小写英文字母：A～Z，a～z。
- 数字字符：0～9。
- 特殊字符：空格　！　＃　％　＾　＆　＊　＿　＋　－　＝　～　＜　＞　／
  \　｜　＂　．　，　()　[]　{}

例如，在例 2.1 的程序中，包含如下 3 条输出语句：

```
cout<<"您好！现在我们开始学习第 2 章"<<endl;
cout<<"i 是整型变量，i 的值可以改变！\n";
cout<<"f 是实型常量，f 的值不能改变\n";
```

以上 3 条语句中双引号引住的部分是由多个字符数据常量组成的字符串常量，其中可以包含 C++字符集以外的汉字字符，而程序的其他部分都由字符集中的字符组成。

2.1.3　C++关键字

关键字是 C++系统预定义的一些具有特别意义的单词。例如，在例 2.1 的程序中，int、float、const、using、namespace、void 都是 C++关键字。其中，int 代表整数类型，float 代表浮点数类型。关键字在程序中不能代表其他含义。下面按字母顺序列出 C++中的关键字：

```
auto  bool  break  case  catch  char  class  const  const_cast  continue
default  delete  do  double  dynamic_cast  else  enum  explicit  extern
false  float  for  friend  goto  if  inline  int  long  mutable  namespace
new  operator  private  protected  public  register  reinterpret_cast  return
short  signed  sizeof  static  static_cast  struct  switch  template  this
throw  true  try  typedef  typeid  typename  union  unsigned  using
virtual  void  volatile  while
```