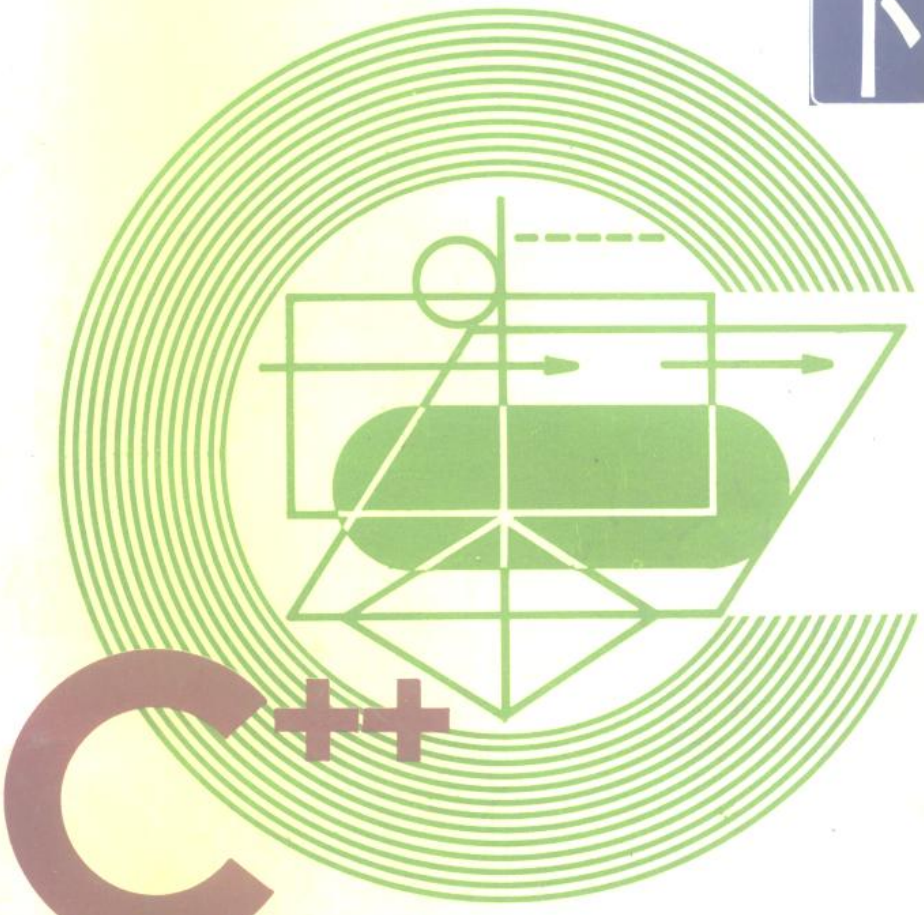
全国高校计算机基础教育研究会组编

计算机教育丛书

谭浩强　主编

下册



C++程序设计教程

谭浩强　刘炳文　编著

中国科学技术出版社

全国高校计算机基础教育研究会组编

计算机教育丛书

谭浩强　主编

C++程序设计教程

下　册

谭浩强　刘炳文　编著

中国科学技术出版社
·北京·

图书在版编目(CIP)数据

C++程序设计教程/谭浩强编. 北京:中国科学技术出版社,1995.7
(计算机教育丛书)
ISBN 7-5046-1956-6
Ⅰ.C…
Ⅱ.谭…
Ⅲ.C语言-程序设计-教材
Ⅳ.TP312C
中国版本图书馆数据据核字(95)07233号

中国科学技术出版社出版
北京海淀区白石桥路32号 邮政编码:100081
新华书店北京发行所发行 各地新华书店经售
天津市武清县长宏印刷厂印刷
*
开本:787×1092毫米 1/16 印张:21.75 字数:600千字
1995年8月第1版 1995年8月第1次印刷
印数:1—5000册 定价:23.00元

前　言

C++是C语言的超集，它保持了与80年代最流行的C语言的高度兼容性，继承了C语言的功能丰富、表达能力强、使用灵活方便、目标程序效率高、可移植性好等特性，同时融合了面向对象的能力，支持面向对象的程序设计。与其他面向对象和支持面向对象的程序设计语言(诸如Modula－2、Smalltalk、Ada等)相比，C++具有更完善的体系结构和更加强大的功能。从应用的角度来说，它可以充分地发挥硬件的潜力，在降低开发成本的同时，提供了强有力的软件开发工具。正是由于具有这些优良特性，C++可以广泛用于各种系统软件和应用软件的开发实践中。在"对象＋消息"的程序设计范式取代"数据结构＋算法"的程序设计范式的过程中，C++无疑有着举足轻重的作用。

应当指出的是，C++是面向对象的语言，但不是纯面向对象的语言，而是混合型面向对象语言，即在过程语言中增加面向对象的结构。因此，用它编写非面向对象的程序，比现有的过程语言(包括C语言)更方便。原有的C语言程序，几乎不用作任何修改就可以在C++环境中编译、运行。

C++本身的优异特性，使得它不仅为计算机专业工作者所使用，而且已经引起广大计算机应用人员浓厚的兴趣，并开始用它编写简单的应用软件。但是，由于C++涉及到一些较复杂的概念，不少初学者感到难以掌握，希望能有一本通俗易懂、适合初学者使用的C++教材或参考书。本书就是为了满足广大计算机应用人员和初学者的迫切要求而编写的。

C++是在C语言的基础上扩展而成的。目前已面世的C++著述，大多把重点放在C语言的扩展部分，这对于已学习过C语言的读者，基本上可以满足需要，但对于从未接触过C语言或对C语言不太熟悉的读者来说，用它来学习C++可能是比较困难的，因为在这之前必须先学习C语言。有鉴于此，本书将把C++作为一种独立、完整的语言来介绍。对于C++和C语言中都有的内容(诸如输入输出、动态存储分配等)，则只介绍与C++有关的部分。

由于篇幅较大，本书分上、下两册出版。上册主要介绍C++的数据类型(包括基本数据类型、数组、结构体、共用体、枚举类型等)，标准输入输出，流程控制，模块化程序设计及指针。下册介绍与面向对象有关的内容，包括类、继承、多态；同时介绍了位运算、C++流和文件操作以及本书使用的C++编译系统——Borland C++3.1集成开发环境(IDE)；此外，还介绍了C++的图形处理功能。

对于已学习过C语言的读者，阅读本书时可以以下册为主；但上册中有一部分内容是C语言中所没有的，也必须认真学习。在目录中，这些章节标以星号(＊)。

本书是C++和Borland C++的基础读物，面向C++的初学者，它不要求读者具有专门的计算机专业知识的基础，只要学习过一种计算机高级语言，就能学习并掌握本书的基本内容。

C++是一种现代的程序设计语言，不依赖于具体的机器和操作系统。但是，如果只是"纸上谈兵"，不编写在具体机器和编译系统中运行的程序，要掌握C++的程序设计技术是不可能的。为此，本书将通过美国Borland国际公司的Borland C++3.1来介绍C++的程序设计方法。Borland C++3.1与Turbo C2.0完全兼容，实现了AT&T C++版本的全部功能，同时支持ANSI C和Microsoft windows应用程序的开发，并具有一定程度的功能扩展。Borland C++编译程序可编译C和C++源代码，具有较高的编译效率和很强的代码优化能力，因此已成为国际上最受欢迎的

面向对象程序设计软件。本书中所有的程序示例均已在IBM－PC系列机上用Borland C++3.1调试通过。但是，本书中介绍的C++知识和C++程序具有通用性，绝大多数程序不用作任何修改或只要进行少量修改就可以在其他C++编译系统中编译、运行。

C++所涉及的概念比较复杂，规则繁多，较难掌握。考虑到读者已有一种以上的计算机语言的基础，对算法和数据结构已有初步的了解，为了减少学习的难度，本书把重点放在语言的使用上，介绍如何用C++编写程序。所举例子主要是为了使读者了解如何正确使用C++，没有过多地讨论算法设计。此外，为了避免顾此失彼，本书没有从面向对象的程序设计入手介绍C++，而是首先把C++作为一种程序设计语言来介绍。在了解了C++的特点、基本概念和功能之后，再回过头来介绍C++面向对象的特性。笔者认为，这样可以使读者集中精力学好C++语言，实际上，掌握了C++的功能之后，再理解面向对象的概念就比较容易了。

C++是一种大型的、全新的程序设计语言，在我国的应用还处于开始阶段。希望本书能成为引玉之砖，激发广大读者对学习和使用C++的兴趣。

本书在编写过程中，参考并引用了谭浩强编著的《C程序设计》(清华大学出版社出版)和谭浩强、刘炳文、鲍泓编著的《Turbo C程序设计教程》(人民邮电出版社出版)。

由于作者水平有限，经验不足，加之时间仓促，缺点和错误在所难免。恳请专家和广大读者批评指正。

编者

1994.8.1

《计算机教育丛书》序

近10多年来，我国的计算机应用和教育事业在蓬勃地发展，愈来愈多的人认识到：没有计算机就没有现代化，计算机知识已经成为当代知识分子知识结构中不可缺少的一个重要组成部分。高等学校中几乎所有的专业都已开设了计算机课程。在中专、职业高中和中小学中也普遍进行着计算机教育。各个领域的在职干部，无论科技人员还是管理人员，都日益感觉到掌握计算机知识推进各项工作的迫切性。总之，一次新的计算机普及的高潮即将到来，我们对此应有充分的准备。

普及计算机应用首先遇到的问题是：缺乏足够的能够驾驭计算机的人才。当务之急是全面深入地开展计算机教育。这种教育应当是全方位多层次的，不同领域不同层次的人都需要在原有基础上学习和提高。我们着眼点首先是大多数。我们的目标是：把计算机从少数人手中解放出来，成为广大群众手中的有力武器。这个任务是十分繁重的，需要众多的有识之士共同投入，通力合作，经过长时间的努力才能实现。

为此我们愿贡献微薄之力，拟编辑一套《计算机教育丛书》促进计算机与普及。这套丛书的主要读者是计算机的初学者和初、中级应用人员。在选题上强调以应用为目的。面向应用。在写法上尽量做到通俗易懂，力求科学性、先进性与通俗性的统一。我们将根据计算机科学技术的发展和读者的要求不断扩充丛书的书目，使之符合社会需要。专家和读者能够给我们指出方向、提出要求、提供信息，参加写作。

本丛书的出版得到全国高等学校计算机教育研究会和中国科学技术出版社的大力支持，使丛书得以问世。期望本丛书能在专家和社会各界的关心爱护下逐步发展和壮大，为计算机教育作出贡献！

主编　谭浩强

1993年6月于北京

内容提要

C++程序设计教程分上下册。下册包括：类；继承与多态；

位运算；流；文件；Borland C++ 3.1 集成开发环境，面向对象程序设计简介；Borland C++图形程序设计等。

本书适合作大专院校教材及对程序设计有兴趣者。

计算机教育丛书编委会

责任编辑：朱桂兰
王　蕾
封面设计：赵一东

目　录

第八章　类

类(class)是C++的精华，是C++最强有力的特征，是进行封装和数据隐藏的工具。它把一组逻辑上相关的实体联系起来，并具备从外部显式地对这些实体进行访问的手段。因此，类是与程序有关的"资源"实体的集成，它提供了组织逻辑上有关的数据、函数及数据对象的技巧，从而为编写大的、复杂的程序提供所需要的资源。和函数一样，类也是C++中模块化程序设计的手段。但是，函数是将逻辑上有关的语句和数据集合在一起，主要用于执行；而类则是逻辑上有关的函数及其数据的集合，它主要不是用于执行，而是提供所需要的资源。类中的函数对用户是隐蔽的，而且受到保护，用户只能使用，不能修改。

第五章中介绍的数组是相同类型数据的集合，结构体允许把不同数据类型的相关数据组合到一个结构中。类是用户定义的一种数据类型，它不但含有不同类型的数据，而且含有实现不同操作的函数。

§8.1　结构体与类

8.1.1　结构体的扩充

第六章我们介绍了结构体。在结构体中，可以含有各种不同类型的数据。实际上，C++的结构体中不仅可以含有不同类型的数据，而且可以含有函数。例如：

```
struct angle
{
    double value;
    void set_value(double);
    double get_sine(void);
    double get_cosine(void);
    double get_tangent(void);
}
```

这是一个含有函数的结构体，该结构体用来计算一个角的正弦、余弦和正切值。这些函数叫做成员函数，为了访问这些成员函数，必须定义该结构体类型的变量，然后像访问结构体中的数据成员一样进行访问。例如：

```
angle deg; // 定义变量
deg.set_value(60.0); // 置初值
deg.get_sine(); //求正弦值
```

上面定义的结构体给出了成员函数的原型，没有定义函数的操作。成员函数的操作与普通函数没有任何区别，但为了能表示与结构体的关系，需要使用作用域运算符"::"。例如：

```
void angle::set_value(double a)
{
    value=a;
```

```
}
```

在这里，作用域运算符::通知编译系统，set_value 是结构体 angle 的成员函数。

对成员函数的调用与一般结构体成员的引用完全一样，即使用成员运算符“.”或“->”。例如：

```
deg.set_value(60.0);
```

或

```
deg->set_value(60.0);
```

根据上面的分析，编写完整的程序如下：

【例 8.1】

```
// Program example 8-1a
#include <iostream.h>
#include <math.h>
const double ANG_TO_RAD=0.0174532925;

struct angle
{
    double value;
    void set_value(double);
    double get_sine(void);
    double get_cosine(void);
    double get_tangent(void);
}deg;
void angle::set_value(double a)
{
    value=a;
}

double angle::get_sine(void)
{
    double temp;
    temp=sin(ANG_TO_RAD*value);
    return temp;
}

double angle::get_cosine(void)
{
    double temp;
    temp=cos(ANG_TO_RAD*value);
    return temp;
}
```

```
double angle::get_tangent(void)
{
    double temp;
    temp=tan(ANG_TO_RAD*value);
    return temp;
}

int main()
{
    deg.set_value(60.0);
    cout<<"The sine of the angle is: ";
    cout<<deg.get_sine()<<endl;
    cout<<"The cosine of the angle is: ";
    cout<<deg.get_cosine()<<endl;
    cout<<"The tangent of the angle is: ";
    cout<<deg.get_tangent()<<endl;
    return 0;
}
```

程序的执行结果如下：

```
The sine of the angle is: 0.866025
The cosine of the angle is: 0.5
The tangent of the angle is: 1.732051
```

程序开头定义了一个符号常量 ANG_TO_RAD，它表示的是 1 度的弧度值，即 $\pi/180$。三角函数的自变量以弧度为单位，当输入角度值时，可以用该常数转换为相应的弧度值。

在 main 函数中，用

```
deg.set_value(60);
```

把角度值 60 送给函数 set_value，函数接收该值后，把它分配给类中的成员变量 value，即对类变量初始化。此后，三个成员函数都可以访问变量 value，从而分别计算出 60 度角的正弦、余弦和正切值。

结构体中的变量 value 只能被成员函数访问，称为结构体的私有部分(private)；而成员函数可以在结构体的外部调用，称为公有部分(public)。因此，结构体 angle 可以写为：

```
struct angle
{
private:
    double value;
public:
    void set_value(double);
    double get_sine(void);
    double get_cosine(void);
    double get_tangent(void);
};
```

在缺省状态下，结构体的成员为公有部分，因此上面结构体中的 public 可以省略。

8.1.2 类的定义

前面介绍了 C++ 中结构体的扩充形式。类的定义与这种结构体的扩充形式十分类似，其一般格式如下：

```
class
{ [private:]
   私有成员数据及函数;
  protected:
    保护成员数据及函数;
  public:
    公有成员数据及函数;
};
```

从上面的定义可以看出，一个类含有三部分，即私有部分、保护部分和公有部分。缺省时，在类中定义的项都是私有的。私有部分(private)的数据和函数只能被该类的成员访问；保护部分(protected)的成员除可以被本类中的成员函数访问外，还可以被本类派生的类的成员函数访问(派生类将在下一章介绍)；公有部分(public)的成员可以被本类之外的函数访问。

实际上，在 C++ 中，类 class 和结构体 struct 是类似的，因为 C++ 的结构体与类一样，也可以含有数据以及对这些数据进行操作的函数。它们的主要区别是缺省值不同：在类中，缺省时成员是私有部分 private，而在结构体中，缺省时成员是公有部分 public。前面例子中的结构体 angle 可以用类改写如下：

```
class angle
{
 private:
     double value;
 public:
     void set_value(double);
     double get_sine(void);
     double get_cosine(void);
     double get_tangent(void);
};
```

如果使用类，则前面例子的程序可以改写如下：

```
// Program example 8-1b
#include <iostream.h>
#include <math.h>
const double ANG_TO_RAD=0.0174532925;

class angle
{
 double value;
```

```
public:
   void set_value(double);
   double get_sine(void);
   double get_cosine(void);
   double get_tangent(void);
}deg;

void angle::set_value(double a)
{
    value=a;
}

double angle::get_sine(void)
{
 double temp;
 temp=sin(ANG_TO_RAD*value);
 return temp;
}

double angle::get_cosine(void)
{
 double temp;
 temp=cos(ANG_TO_RAD*value);
 return temp;
}

double angle::get_tangent(void)
{
 double temp;
 temp=tan(ANG_TO_RAD*value);
 return temp;
}

int main()
{
    deg.set_value(60.0);
    cout<<"The sine of the angle is: ";
    cout<<deg.get_sine()<<endl;
    cout<<"The cosine of the angle is: ";
    cout<<deg.get_cosine()<<endl;
    cout<<"The tangent of the angle is: ";
```