经典实例精解

Visual C++数据库开发经典实例精解

夏邦贵 刘凡馨 编著



机 械 工 业 出 版 社

需要全本请访问 www.ertongbook.com

本书主要介绍如何使用 Visual C++ 6.0 进行数据库管理系统开发，全书共分 3 篇：基础篇、应用篇和高级篇，各篇中的实例涉及到 Visual C++ 6.0 数据库管理系统在管理和开发领域中的应用。基础篇介绍了 Visual C++ 6.0 程序设计基础知识；应用篇包括 6 个小型实例：经典 HelloWord、ODBC 报表、ODBC 通讯簿、DAO 好友录、OLEDB 口令管理和 ADO 数据管理；高级篇包括一个比较大型的综合实例——考试管理系统。各篇实例的实用性极强，涵盖了 Visual C++ 6.0 数据库管理系统管理和开发的大部分操作知识点，主要包括 ODBC、DAO、OLEDB 和 ADO 等数据库访问技术。

本书定位于已有一定基础的 Visual C++ 6.0 初中级用户、数据库管理系统开发相关人员以及计算机爱好者学习使用，还可作为社会相关办学、大中专院校的辅助教材使用。

为了方便读者学习和操作，我们将书中实例所涉及到的源代码都收录到了本书的配套光盘中，供读者在学习过程中调用。

图书在版编目（CIP）数据

Visual C++ 6.0 数据库开发经典实例精解/夏邦贵，刘凡等编著.
-北京：机械工业出版社，2006.5
（经典实例精解）

ISBN 7-111-18874-8

Ⅰ.V… Ⅱ.①夏… ②刘… ⅢI.C 语言-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2006）第 031570 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：吴宏伟 责任编辑：李虎斌 版式设计：赵丽娜
三河市宏达印刷有限公司印刷
2006 年 5 月第 1 版第 1 次印刷
787mm×1092mm 1/16 • 20.75 印张 • 470 千字
0001-5000 册
定价：29.80 元（含 1CD）

凡购本图书，如有缺页、倒页、脱页，由本社发行部调换
本社购书热线电话：（010）68326294
封面无防伪标均为盗版

有言在先

为什么写这本书？

Visual C++ 6.0 是 Microsoft Visual Studio 6.0 中一个重要组件，它与 Visual Basic 6.0、Visual FoxPro 6.0 一起构成了 Microsoft 独具特色的程序开发工具集合。Visual Basic 6.0 比 Visual C++ 6.0 简单，更适用于初学者。Visual FoxPro 6.0 本身是一个数据库管理系统，并提供了程序开发功能，也比较简单。Visual C++ 6.0 则属于专业的程序开发工具，对具有一定程序设计基础，熟悉 C 或 C++的人更适合。

Visual C++ 6.0 在其强大的 Windows 应用程序开发功能之上，为程序员提供了功能完备、强大的数据库应用程序开发功能。Microsoft 总是力求在他的产品中为用户提供尽量完全的选择。Visual C++ 6.0 支持各种常用的数据库开发技术，包括 ODBC、DAO、OLEDB 和 ADO 等。

Visual C++ 6.0 力求将所有工作简单化，使用 MFC 应用程序向导，可以快速创建一个数据库应用程序。用 MFC 类向导，可以快速创建访问数据库的记录集类。当然，如果要开发功能更复杂的数据库应用程序，需要程序员熟练掌握 Visual C++ 6.0 集成开发环境，能够阅读和编写 C++代码，并掌握 MFC 中与 ODBC、DAO、OLEDB 和 ADO 等相关的类。

因为 Visual C++ 6.0 强大的应用程序和数据库应用程序开发功能，倍受用户青睐，市场上关于 Visual C++ 6.0 的图书也多如牛毛，其中大部分书要么讲开发，要么以简单实例来讲解，并未在操作前对实例进行设计分析，举例也不具代表性，这样造成的结果是使读者学后只会掌握 Visual C++ 6.0 基本的程序设计和数据库应用程序开发，或者只能照着别人的实例步骤操作而不能自己进行数据库管理系统开发，更谈不上举一反三了。许多初学者也容易只注重 Visual C++ 6.0 基本的数据库开发技术，并未有效地提高系统开发的技能，而通过实例来学习 Visual C++ 6.0 既可以了解相关的基础知识，又能快速提高 Visual C++ 6.0 数据库系统开发的技能，是学习 Visual C++ 6.0 的最有效方法。

本书作者长期从事数据库管理系统开发，我们深知很多读者朋友都希望购买的图书能多举实例、多讲技巧，并与系统开发应用结合起来，便于他们在工作中应用自如，提高工作效率，取得更好的成果。于是我们编写了这本《Visual C++ 6.0 数据库开发经典实例精解》，希望本书能给广大 Visual C++ 6.0 用户带来意想不到的收获。

这本书适合您吗？

如果您只是一个 Visual C++ 6.0 初学者，你迫切需要认识它、掌握它，那么本书将成为您打 SQL Server 这扇大门的“开门钥匙”。因为本书为了满足初学者的需要，在第 0 篇



中系统而全面地介绍了关系数据库和 Visual C++ 6.0 的基础知识，再通过后面实例部分的学习，想信可以使初学者从入门到精通。

如果您是一个希望从事 Visual C++ 6.0 据库管理系统开发的非专业人员，对 Visual C++ 6.0 已有一定的了解，但对独立设计数据库管理系统时还感到茫然，那您得看看本书，本书不仅介绍了数据库管理系统开发，还介绍了如何进行设计。

如果您对 Visual C++ 6.0 的基本基础知识已经比较熟悉了，也能设计一些简单的程序，但想加深对 Visual C++ 6.0 的认识，或想在系统开发领域找一份工作，本书可以让你找到自信，让您真正领略到用 Visual C++ 6.0 进行数据库管理系统开发的技能，达到举一反三、融会贯通的效果。

Visual C++ 6.0 是一个典型的 Windows 应用程序开发工具，它侧重于程序开发，并不像 Visual FoxPro 一样提供数据库管理功能。所以，如果想成为真正的数据库管理系统开发人员，你至少还需要掌握一种数据库管理系统，如 Microsoft Access。

本书是以中文版 Visual C++ 6.0 进行编写的，全书详细介绍了共 7 个实例，各个实例所涉及到的知识点由浅入深，其目的是让一个 Visual C++ 6.0 初学者从入门到精通，再到实际的应用，让一个已有一定 Visual C++ 6.0 和数据库管理基础的用户掌握如何自己设计并开发出具有使用性的数据库管理系统。因此，无论您是一个初学者还是已有一定基础的读者，本书都无疑是一个很好的选择。另外，本书也可作为各类培训班、大中专院校上机教材和自学参考书。

本书看点

本书是我们在精心研究了市面上的实例类书籍之后，并汲取众多电脑培训学校教师的优秀经验及自学成才的设计大师的宝贵意见之后而编写的。与其他同类图书相比，本书有以下几大看点：

（1）选择的实例代表性强。本书选择的实例均是精挑细选，涉及当前广泛使用的数据库管理系统。

（2）详尽的分析。市面上某些实例类电脑图书中对于实例的处理一般都是直接讲解绘制该实例的操作步骤，注重一步一步教读者做实例，而缺乏必要的分析过程，使读者不能达到活学活用的目的。实际上，大多数实例的制作方法往往有好几种，对这些方法的详细分析可以使读者在学习时融汇贯通，举一反三。本书每个实例分析部分以“制作特点”、“设计分析”等小栏目向读者详细分析本例的制作特点、实现途径以及不同的方法等，使读者在动手制作之前即做到心中有数。

（3）兼顾相关知识的讲解。一般来说，每个实例均可涉及覆盖一定的知识点，而一些概念性的知识或与该软件无关但与实例操作过程密切相关的知识等在实例中不可能详细讲解，这就需要单独提出讲解。本书在“技术要领”小栏目中即向读者讲解本例涉及到的相关软件知识点，保证读者汲取知识的完整性及系统性。

（4）完整的操作过程。操作步骤的讲解详尽仔细，没有跳步和漏步，只要读者依照书中的步骤操作即可得到相应的实例效果。

（5）如何做到举一反三。“举一反三”部分是本书的一大亮点，为读者提供 3 个与

本实例密切相关的题目供读者练习，使本例涉及到的相关知识得到巩固。

（6）活泼可爱的卡通提醒。本书中一些需要特别说明的知识点、需要注意的问题和操作技巧等均由活泼可爱的鼠标卡通精灵随时为您提醒。

如何学习本书

本书内容编排尽量做到深入浅出、图文并茂，将程序设计语言和数据管理系统开发融为一体，力求以最简捷、最优化的方法开发出相应管理软件。本书各篇内容如下：

★ 第 0 篇：基础篇。系统地介绍了 Visual C++ 6.0 程序设计的基础知识，若你是一个 Visual C++的初学者，建议你从该篇学起。

★ 第 1 篇：应用篇。通过经典 HelloWord、ODBC 报表、ODBC 通讯簿、DAO 好友录、OLEDB 口令管理和 ADO 数据管理等小型实例设计，让读者掌握 Visual C++ 6.0 支持的各种数据库访问技术。

★ 第 2 篇：高级篇。通过创建考试管理系统这个比较大型实例设计，让读者掌握使用 Visual C++ 6.0 开发数据库管理系统的技能。

软件版本的处理

本书使用中文版 Visual C++ 6.0，在 Windows 2000 中完成所有程序的调试工作，并使用了以 Microsoft Access 2003 和 Microsoft SQL Server 2000 桌面引擎（MSDE）。在实例 6 中使用了 Microsoft SQL Server 数据库，其他实例中使用的都是 Microsoft Access 数据库。

本书配套光盘提供了实例中使用的数据库，Microsoft Access 数据库复制到硬盘即可使用，而 Microsoft SQL Server 数据库必须附加到服务器之后才可使用。

本书光盘内容

本书附送的光盘中收录了本书所有实例的源代码和数据库，以便读者在学习过程中使用。

光盘的内容说明如下：

★ \实例\　　收录本书所有实例的源代码和数据库。在“\实例\Sample*n*\工程名称\”文件夹中包含了该实例工程的所有源代码。读者可打开其中的工作区文件（*.dsw）来打开工程。在“\实例\Sample*n*\数据库\”文件夹中放了该实例使用的数据库。

★ \举一反三\　　收录本书各个实例中“举一反三”板块中各习题的源代码。

软件及素材的获取

Visual C++ 6.0 和 Microsoft Access 2003 软件是由 Microsoft 公司推出并全球销售的，一般用户可到当地电脑城通过相应的软件供应商或软件销售处购买。Microsoft SQL Server 2000 桌面引擎 Microsoft 提供了免费下载使用，网址为 http://www.microsoft.com/downloads/details.aspx?FamilyId=413744D1-A0BC-479F-BAFA-E4B278EB9147&displaylang=zh-cn。

需要全本请访问 www.ertongbook.com



学习本书能够达到的效果

无论您是一个 Visual C++ 6.0 初学者还是一个已有一定基础的 Visual C++ 6.0 用户，我们都希望你能从头到尾系统地学习一遍，只要认真地学完本书，我们相信您的收获将是无法比拟的，你的知识和技术又将得到新的升华！当然，万事并不能十全十美，本书也不是万能的，如果您想能为一个 Visual C++ 6.0 高手或数据库管理系统软件设计师，那么日后的努力是非常重要的，希望大家能努力追求！

与作者交流并获取帮助

本书由西华大学夏邦贵、刘凡馨编著，参加编写的人员有：李莹、张艳秋，另外，陈宁、曾祥久、杨茂勇、李跃、肖渝、丁如容、任军、李翠英、周桂林、张洋、余有洲、付小军等参与了本书资料的收集整理及部分章节的编写工作，在此一并致谢。由于编者水平有限，加之时间仓促，书中疏漏和不足之处在所难免，肯请各位读者及专家不吝赐教。

如果您对本书某个问题有疑问，如果您在学习过程中遇到不能解决的难题，如果您对我们有什么意见及建议，请通过 E-mail：book@itebook.com 与我们联系，我们将有问必答，为您提供超值延伸服务。

电脑视界





第 0 篇　基础篇

第0篇

Visual C++ 6.0 基础

Visual C++是 Microsoft Viusal Studio 6.0 包含的组件，Visual 指“可视化”，使人容易联想到 Viusal Studio 6.0 的另一个组件 Visual Basic 6.0。Visual Basic 6.0 以其可视化的集成开发环境、简单易学特点，得到了非常多的程序设计开发人员的青睐。很多初学者都选择 Visual Basic 6.0 作为程序设计的入门。这让人们认为 Visual C++ 6.0 与 Visual Basic 6.0 一样，至少在刚开始学习 Visual C++ 的一段时间内会这么认为。Visual C++ 6.0 是在 C++基础上发展而来的，它采用面向对象的程序设计方法。Visual C++具有复杂的应用程序框架，它是为更加专业的程序开发人员使用的。使用 Visual C++提供的各种向导，可以快速、甚至不用编程就可创建 Windows 应用程序。但是，程序员必须能够阅读、理解和编写 C++代码，必须掌握 MFC 库的结构和 Windows 操作系统的工作原理。为了让读者能够学习本书后面的各个实例，本章讲解了 C++程序设计和类的基础知识。



- *Visual C++ 6.0 程序设计基础*
- *类*

0.1 编程风格

0.1.1 注释

注释用于在代码中添加说明性的文字，用于解释代码，便与阅读程序。

Visual C++有两种注释。第一种注释以//开头，直到本行结束。例如：

```
//这里一个单行注释
CEmailSet m_EmailSet;        //定义 CemailSet 类对象
```

第二种注释以/*开始，以*/结束，允许跨越多行。这种一般用于在文件开头添加说明文字。例如：

```
/************************************************
*EmailBookView.cpp ： CemailBookView 类的实现文件。
*用于创建 Email 通讯簿
************************************************/
```

0.1.2 命名约定

Visual C++允许在名称中同时使用大写和小写字母，并区分大小写。例如，abc、Abc和 ABC 是三个不同的名称。

决大多数 Visual C++程序员都对变量用小写，对常用用大写。Visual C++中的内部常量都使用大写名称。

0.1.3 代码格式

Visual C++以分号（;）作为语句结束符号，可以将一条语句分多行书写。例如，下面是一条语句：

```
pDC->TextOut(pText.x+100,
                pText.y,
                m_pEmails->m_Email);
```

下面是相同的语句，仅仅书写格式不同而已：

```
pDC->TextOut(pText.x+100,pText.y, m_pEmails->m_Email);
```

0.1.4 代码缩排

缩排可以使程序结构清晰，大多数程序员都会缩排程序代码。一般情况下，一个新的块或条件缩排一级。例如：

```
for (i=1;i<9;i++)
    for (j=i;j<10;j++)
    {
        if (a[i]>a[j])
        {
```

```
                k=a[i];
                a[i]=a[j];
                a[j]=k;
            }
        }
```

下面的代码是没有使用缩排，对 Visual C++来说仍是正确的代码，但前面的书写格式更具有可读性：

```
for (i=1;i<9;i++)
for (j=i;j<10;j++)
{
if (a[i]>a[j])
{
k=a[i];
a[i]=a[j];
a[j]=k;
}
}
```

0.1.5 程序基本结构

一个简单的 Viusal C++程序如下：

```
#include "stdafx.h"
int main(int argc, char* argv[])
{
    printf("这是 Visual C++中输出的信息！\n");
    return 0;
}
```

main 函数是 C++程序的执行入口，即从 main 函数开始执行。该程序在运行时，在 Windows 的命令窗口（控制台窗口）中显示“这是 Visual C++中输出的信息！”如图 0-1 所示。

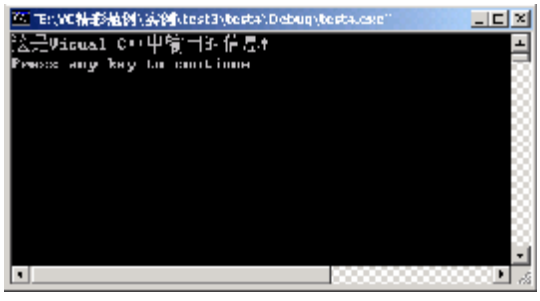


图0-1 Visual C++编程的控制台应用程序输出

这是一个在 Visual C++中创建的控制台应用程序。控制台应用程序在命令窗口中显示

输出和输入数据。

Windows 应用程序以 WinMain 函数作为入口，Visual C++应用程序向导在创建程序时，可自动根据需要编写 WinMain 函数。本书后面的各个实例都是使用 Visual C++应用程序。

0.2 变量与常量

0.2.1 数据类型

Visual C++提供两种基本数据类型：整数和浮点数

表 0-1 列出了整数和浮点数数据类型。

表0-1　　整数数据类型

类型	占用内存（字节）	说明
char	1	用单引号括起来的单个字符，如'A'
short	2	短整数
int	4	有符号整数
__int*n*	*N*	指定大小的整数，*n* 可以是 8、16、32 或 64
long	4	长整数
float	4	单精度浮点数
double	8	双精度浮点数
long double	8	与 double 相同

表中列出的类型类型默认为有符号数，可在类型名称前用 signed 关键字强调，如 signed int。若要声明无符号数据类型，可使用 unsigned 关键字 unsigned int。

0.2.2 变量声明

可在程序中改变变量的值，Visual C++要求变量必须在声明时候才能使用。

1. 声明格式

变量声明的一般格式如下：

类型名称 变量名称;

变量名称通常以字母开头，包含字母、数字、下划线等字母，不包含空格。Visual C++区别大小写，abc、Abc 和 ABC 是三个不同的变量。

例如：

```
int x;
```

Visual C++允许同时声明多个相同类型的变量，用逗号分隔各个变量。例如：

```
int x,y,z;
```

2. 初始化变量

可在声明时初始化变量，例如：

```
int x=100;
```


需要全本请访问 www.ertongbook.com

3. 引用变量

Visual C++允许将变量声明为引用类型，例如：

```
int x=100;
int& y=x;
```

&声明 y 是 x 的一个引用，即 y 和 x 表示同一个变量。在代码中使用 y 和 x 都是相同的。

0.2.3 常量

1. const 常量（符号常量）

关键字 const 用于定义符号常量，例如：

```
const float PI=3.1415926;
```

PI 即代表 3.1415926，不允许在程序中修改其值。

2. 八进制和十六进制常量

整数默认为十进制，如 12、-34。Visual C++允许使用八进制和十六进制。以零开头的整数为八进制，以 0x 开头为十六进制。例如：

```
x=012          //012 以零开头，为八进制数，x 的值为 10
y=0x12         //0x12 为十六进制数，y 的值为 18
```

0.3 字符与字符串

0.3.1 字符

类型 char 表示用单引号括起来的单个字符。字符变量的声明格式如下：

```
char  变量;
```

'a'、'A'、'!'、'1'、'3'等都是字符常量。可将字符常量赋值给字符变量，例如：

```
char c;
c='a';
```

有一些特殊的字符，如制表符 Tab、回车符、换行符等。这些特殊字符不能直接表示为字符常量，需使用反斜杠字符\（也叫转义字符）加上对应的字符表示。例如，\t 表示制表符 Tab、\r 表示回车符、\n 表示换行符。

表 0-2 列出了常用的特殊字符。

表0-2 特殊字符

字符	名称	作用
\a	响铃符	让扬声器发出“嘟”的一声
\b	退格符	光标左移一个字符
\f	换页符	光标移动到新的一页的第一行开始位置
\n	换行符	光标移动到下一行开始位置

（续表）

\r	回车符	光标移动到当前行开始处
\t	制表符	光标移动到下一个制表符位置（8 个字符宽度）
\'	单引号	字符'
\"	双引号	字符"
\\	反斜杠	字符\
\nnn	八进制字符	nnn 为字符 ASCII 码的八进制
\x*hhh*	十六进制字符	hhh 为字符 ASCII 码的十六进制

0.3.2 字符串

字符串指用双引号括起来的多个任意字符。"123"、"abc"、"a"、"1"等都是字符串。表 0-2 列出的特殊字符只能出现在字符串中。

1. MFC 字符串

Visual C++提供了 MFC 定义了一个 CString 类用于处理字符串。字符串变量声明格式如下：

```
CString name;
```

在 Visual C++必须使用 MFC，并在工程文件开头包含 afx.h 头文件，才能使用 CString 类。CString 类提供了大量的成员函数用于处理字符串，如 GetLength 函数用于获得字符串长度，使用格式如下：

```
CString name;
name="Mike";
x=name.GetLength();
```

字符串变量允许以数组的形式访问其中的字符。例如，name[0]表示 name 中的第 1 个字符，name[1]表示 name 中的第 2 个字符。

关于 CString 类，读者可参考 MSDN 中的 Visual C++文档。

2. c 风格字符串

c 风格字符串是一个字符数组。特殊字符'\0'表示字符串结束。例如：

```
char name[4];
name[0]='M';
name[1]='i';
name[2]='k';
name[3]='\0';
```

name 是一个包含 4 个元素的字符数组。必须最后一个数组元素中保存'\0'。不允许直接将字符串赋值给字符数组：

```
name="Mik";        //错误
```

0.4 运算符和表达式

0.4.1 算术运算符和算术表达式

Visual C++有五个简单的运算符，用于数值计算，如表 0-2 所示。

表0-3 算术运算符

运算符	说明
*	乘法
/	除法
+	加法
-	减法
%	取模（整除后取余数）

乘（*）、除（/）和取模（%）的优先级高于加（+）和减（-），可用括号改变优先级。例如：

(10+20)*30

除法可分整数除法和浮点除法。若两个操作数都是整数，则执行整数除法，结果也为整数。在整数除法中，结果的小数部分被截断（即丢弃小数部分）。例如，5/2 的结果为 2。

若除法中有一个操作数为浮点数（带小数部分），则执行浮点除法，结果仍为浮点数。例如，5.0/2 的结果为 2.5。

只包含算术运算符的表达式称为算术表达式。

0.4.2 关系运算符

关系运算符用于比较两个操作数的大小关系。表 0-3 列出了 Visual C++的关系运算符。

表0-4 关系运算符

运算符	说明
<	小于
<=	小于等于
>	大于
>=	大于等于
==	相等
!=	不等

关系运算的结果 int 类型，关系成立的结果为 1，不成立表示 0。例如，2>3 结果为 0，2<3 结果为 1。

使用关系运算符连接的多个常量或算术表达式称为关系运算符。例如：

x+2>3

0.4.3 逻辑运算符

逻辑运算符对逻辑值进行运算，0 表示逻辑假，1 表示逻辑真，非 0 数也视为逻辑真。表 0-4 列出了 Visual C++的逻辑运算符。

表0-5 逻辑运算符

运算符	说明
!	逻辑非，!1 为 0，!0 为 1
&&	逻辑与，两个操作数只要有一个为 0，则结果为 0
||	逻辑或，两个操作数只要有一个为 0，则结果为 1

0.4.4 位运算符

位运算符用于对数值执行二进制的位操作。

1. 按位与&

对两个操作数按二进制位执行与操作，若对应位都为 1，则结果中该位为 1。若对应位有一个为 0，则结果中该位为 0。例如：

```
int x=9,y=7,z;        //9 的二进制为 0000 1001,6 的二进制为 0000 0111
z=x & y;              //z 为 1,即 0000 0001
```

2. 按位或|

对两个操作数按二进制位执行或操作，若对应位都为 0，则结果中该位为 0。若对应位有一个为 1，则结果中该位为 1。例如：

```
int x=9,y=3,z;        //9 的二进制为 0000 1001,6 的二进制为 0000 0011
z=x | y;              //z 为 11,即 0000 1011
```

3. 按位异或^

对两个操作数按二进制位执行异或操作，若对应位一个为 0，一个为 1，则结果中该位为 1，否则为 0。例如：

```
int x=9,y=7,z;        //9 的二进制为 0000 1001,6 的二进制为 0000 0111
z=x ^ y;              //z 为 10,即 0000 1010
```

4. 向左移位<<

将操作数向左移动指定位数。格式为：

操作数 << 移动位数

例如：

```
z=1<<2;               //z 为 4,即 0000 0100
```

5. 向右移位>>

将操作数向右移动指定位数。格式为：

操作数 >> 移动位数

例如：

```
z=8>>2;               //z 为 2,即 0000 0010
```

0.4.5 赋值运算符

赋值运算符用于将表达式的值存储到指定的变量中，格式为：

变量名 赋值运算符 表达式;

在赋值语句末尾必须加分号。

1. 简单赋值=

将右侧表达式的值赋给左边的变量。例如：

```
x=123
y=x+100
```

2. 乘法赋值*=

将变量乘以表达式值的结果赋给变量。例如：

```
x*=123          //等同 x=x * 123
```

3. 除法赋值/=

将变量除以表达式值的结果赋给变量。例如：

```
x/=10           //等同 x=x / 10
```

4. 取模赋值%=

将变量乘以表达式值的余数赋给变量。例如：

```
x%=10           //等同 x=x % 10
```

5. 加法赋值+=

将变量加上表达式值的结果赋给变量。例如：

```
x+=123          //等同 x=x + 123
```

6. 减法赋值-=

将变量减去表达式值的结果赋给变量。例如：

```
x-=123          //等同 x=x - 123
```

7. 左移位<<=

将变量向左移动表达式值的位数，再赋给变量。例如：

```
x<<=3           //等同 x=x << 3
```

8. 右移位>>=

将变量向右移动表达式值的位数，再赋给变量。例如：

```
x>>=3           //等同 x=x >> 3
```

9. 按位赋值&=

将变量和表达式值按位与，再赋给变量。例如：

```
x&=3            //等同 x=x & 3
```

10. 按位或赋值|=

将变量和表达式值按位或，再赋给变量。例如：

```
x|=3        //等同 x=x | 3
```

11. 按位异或赋值^=

将变量和表达式值按位异或，再赋给变量。例如：

x^=3　　//等同 x=x ^ 3

12. 自加++

对变量加 1，例如：

++x

x++

等同

x=x+1

13. 自减--

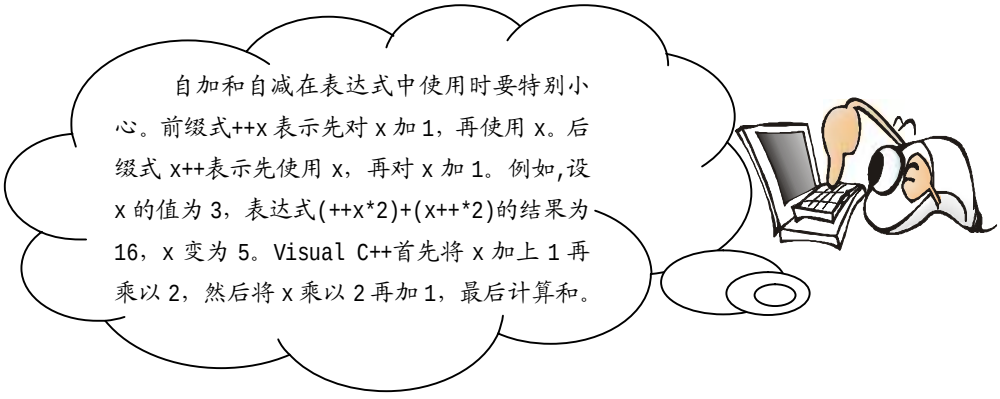
对变量减 1，例如：

--x

x--

等同

x=x-1



0.5 输出与输出

0.5.1 输出到控制台

1. cout 输出

用于向标准输出设备（显示器）输出数据，格式如下：

cout << 表达式

例如：

cout << 123

cout << x+123

输出结果为：

123223

同时输出多个表达式的格式如下：

需要全本请访问 www.ertongbook.com