嵌入式系统系列教材

高级程序设计技术

曾凡仔　杜四春　银红霞　徐署华　编著

人民邮电出版社
POSTS & TELECOM PRESS

嵌入式系统系列教材

高级程序设计技术

曾凡仔　杜四春　银红霞　徐署华　编著

人民邮电出版社
北京

图书在版编目（CIP）数据

高级程序设计技术 / 曾凡仔等编著. — 北京 : 人民邮电出版社, 2009.11
（嵌入式系统系列教材）
ISBN 978-7-115-21456-0

Ⅰ. ①高… Ⅱ. ①曾… Ⅲ. ①程序设计－教材 Ⅳ. ①TP311.1

中国版本图书馆CIP数据核字(2009)第164480号

内 容 提 要

本书全面系统地阐述了 C++语言的基本概念、语法和面向对象的编程方法；对 C++语言面向对象的基本特征：类和对象、继承性、派生类、多态性和虚函数等内容作了详尽的介绍；从软件开发的实际需要出发，按照面向对象的程序设计思想，详细地介绍了线性表、查找、排序等数据结构及其算法实现。本书例举了丰富的例题，每章后面备有形式多样的练习题。在内容安排上循序渐进、深入浅出，力求通俗易懂、突出重点、侧重应用。

本书不仅可作为高职高专院校和培训机构 C++语言程序设计的教材，也可作为自学 C++语言的指导用书和计算机工程技术人员的参考书。

嵌入式系统系列教材

高级程序设计技术

◆ 编　　著　曾凡仔　杜四春　银红霞　徐署华
责任编辑　王建军
执行编辑　李　静

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鸿佳印刷厂印刷

◆ 开本：787×1092　1/16
印张：21.25
字数：528 千字　　2009 年 11 月第 1 版
印数：1－3 000 册　　2009 年 11 月北京第 1 次印刷

ISBN 978-7-115-21456-0

定价：42.00 元

读者服务热线：(010)67119329　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

C++语言是目前使用最为广泛的一种高效程序设计语言，它既可以进行过程化的程序设计，也可以用于面向对象的程序设计。C++是从C语言发展演变而来的，是C语言的超集。它实现了类的封装、数据隐藏、继承及多态，使得其代码容易维护及高度可重用。

本书作为C++语言的入门教材，不仅详细地介绍了C++语言本身，还深入地阐述了面向对象的程序设计方法。本书的主要特点是语言流畅，简洁易懂，例题丰富，实用性强。这使读者不仅可以学会一门程序设计语言，还能初步掌握面向对象的程序设计方法。其中，丰富的例题使初学者可以在学习的同时就开始积累初步的编程经验，以尽快达到学以致用的目的。

本书共计8章。

第1章C++语言概述，主要介绍C++的发展历史，面向对象的程序设计概念，C++的词法与规则，C++程序的结构与实现；数据类型，常量、变量，运算符与表达式，流控制，数据的输入/输出；C++语句，顺序、分支和循环程序设计。

第2章类和数据抽象，主要介绍类、对象，对象的初始化，构造函数与析构函数，堆与拷贝构造函数，静态成员和静态成员函数，友元函数和友元类，this指针等。

第3章继承和派生，主要介绍基类和派生类，单继承、多继承和虚基类。

第4章多态性与虚函数，主要介绍模板的概念，函数模板和类模板，运算符重载，普通成员函数重载，构造函数重载，派生类指针，虚函数，纯虚函数和抽象类等。

第5章C++流，主要介绍I/O标准流类，键盘输入、屏幕输出，磁盘文件的输入和输出，字符串流等内容。

第6章线性表，主要介绍线性表的逻辑结构，线性表的顺序存储及运算实现，线性表的链式存储和运算实现及顺序表和链表的比较等内容。

第7章查找，主要介绍查找的基本概念，静态查找表，动态查找表，哈希表查找等内容。

第8章排序，主要介绍排序的基本概念，插入排序，交换排序，选择排序，二路归并排序，基数排序和外排序等内容。

附录中给出了Linux环境下C++编译系统提供的库函数和类库，以方便读者查阅。

本书中所有例题都在Linux环境下C++编译系统中运行通过，在其他版本的编译系统中一般也都可以运行。本书为高职高专院校的C++程序设计教材，建议教授课时为45课时，上机实践课时为45课时，课程设计课时为16课时。各院校可根据教学实际情况适当增删。

本书也可作为大中专院校的程序设计课程教材和各类培训机构培训教材，还可供从事计算机应用的工程技术人员参考。

本书在编写过程中，编者参阅了许多C++的参考书和有关资料，并参阅了一些翻译的书籍，现谨向这些书的作者和译者表示衷心的感谢。

由于编者水平所限，书中难免有不妥或错误之处，欢迎广大读者指正。

编　者

2009年6月于岳麓山

目　录

第 1 章　C++语言概述

1.1　C++语言的起源与特点

C++语言是一门含 C 语言子集的高效程序设计语言。它比 C 语言更容易为人们学习和掌握，并且以其独特的语言机制在计算机科学领域得到了广泛的应用。它既可以进行过程化的程序设计，也可用于面向对象的程序设计。

1.1.1　从 C 到 C++

C++源于 C 语言，而 C 语言是在 B 语言的基础上发展起来的。1960 年，出现了一种面向问题的高级语言 ALGOL 60，它离硬件比较远，不宜用来编写系统软件。1963 年，英国剑桥大学推出了 CPL（combined programming language），后来简化为 BCPL。1970 年，美国贝尔（Bell）实验室的 K.Thompson 以 BCPL 为基础，设计了一种类似于 BCPL 的语言，取其第一字母 B，称为 B 语言。1972 年，美国贝尔实验室的 Dennis M.Ritchie 为克服 B 语言的诸多不足，在 B 语言的基础上重新设计了一种语言，取其第二字母 C，称为 C 语言。设计 C 语言的最初目的是编写操作系统。其简单、灵活的特点，很快就被用于编写各种不同类型的程序，从而成为世界上最流行的语言之一。但是，C 语言是一个面向过程的语言。随着软件开发技术的进步，程序员们最终发现，把数据和施加在数据上的操作结合起来，会使程序更易于理解，可读性更好，由此产生了面向对象的程序设计思想。

1980 年，贝尔实验室的 Bjarne Stroustrup 对 C 语言进行了扩充，推出了“带类的 C”，多次修改后起名为 C++。以后又经过不断的改进，发展成为今天的 C++。C++改进了 C 的不足之处，支持面向对象的程序设计，在改进的同时保持了 C 的简洁性和高效性。

目前，C++越来越受到重视并得到了广泛的应用，成为在商业软件开发中占统治地位的语言。

1.1.2　C++与 C 的关系

首先，C++语言是一个更好的 C 语言，C++语言根除了 C 语言中存在的问题，并保证与 C 语言相兼容。事实上，C++语言就是 C 语言的一个超集，绝大多数的 C 语言代码无须修改就可以直接在 C++程序中使用，这使得 C 程序员能够更容易地学习 C++语言。

其次，C++语言是支持面向对象的程序设计语言。为了保持与 C 的兼容，C++语言也支持面向过程的编程，因此，C++语言并不是一个纯粹的面向对象的设计语言。尽管如此，在使用 C++编程时，还是应注意采用面向对象的思维方法。C 程序员需要从全新的面向对象的角度来学习 C++语言，利用其新技术。

C++语言是在C语言的基础上发展起来的，但它更是一次变革。两者的本质差别在于C++语言支持面向对象的程序设计，C语言仅支持面向过程的程序设计。掌握好面向对象的程序设计思想是学好C++语言的关键。

学习C++语言必须有C语言基础吗？不是的。C++语言是一个完整的语言，读者完全可以直接学习C++语言。当然，读者如果具有C语言的基础，或许会更快地成为一个好的C++程序员。

1.1.3 C++面向对象的特性

1．C++支持数据封装

支持数据封装就是支持数据抽象。在C++中，类是支持数据封装的工具，对象则是数据封装的实现。面向过程的程序设计方法与面向对象的程序设计方法对待数据和函数关系是不同的，在面向对象的程序设计中，将数据和对该数据进行合法操作的函数封装在一起作为一个类的定义，数据将被隐藏在封装体中，该封装体通过操作接口与外界交换信息。对象被说明为具有一个给定类的变量，类似于C语言中的结构，在C语言中可以定义结构，但这种结构包含数据，而不包含函数。C++中的类是数据和函数的封装体。在C++中，结构可作为一种特殊的类，它虽然可以包含函数，但是它没有私有或保护的成员。

2．C++类中包含私有、公有和保护成员

C++类中可定义3种不同访控制权限的成员。一种是私有（private）成员，只有在类中说明的函数才能访问该类的私有成员，在该类外的函数不可以访问私有成员；另一种是公有（public）成员，类外面也可访问公有成员，成为该类的接口；还有一种是保护（protected）成员，这种成员只有该类的派生类可以访问，其余的在这个类外不能访问。

3．C++中通过发送消息来处理对象

C++中是通过向对象发送消息来处理对象的，每个对象根据所接收到的消息的性质来决定需要采取的行动，以响应这个消息。响应这些消息是一系列的方法，方法是在类定义中使用函数来定义的，使用一种类似于函数调用的机制把消息发送到一个对象上。

4．C++中允许友元破坏封装性

类中的私有成员一般是不允许该类外的任何函数访问的，但是友元可打破这条禁令，它可以访问该类的私有成员（包含数据成员和成员函数）。友元可以是在类外定义的函数，也可以是在类外定义的整个类，前者称为友元函数，后者称为友元类。友元打破了类的封装性，它是C++另一个面向对象的重要牲。

5．C++允许函数名和运算符重载

C++支持多态性，允许一个相同的函数名或运算符代表多个不同实现的函数，这被称为函数或运算符的重载，用户可以根据需要定义函数重载或运算符重载。

6．C++支持继承性

C++中可以允许单继承和多继承。一个类可以根据需要生成派生类。派生类继承了基类的所有方法，另外派生类自身还可以定义所需要的不包含在父类中的新方法。一个子类的每个对象包含从父类那里继承来的数据成员以及自己所特有的数据成员。

7．C++支持动态联编

C++中可以定义虚函数，通过定义虚函数来支持动态联编。

以上所讲的是C++对面向对象程序设计中的一些主要特征的支持。

1.2 C++语言的基本符号与词法

1.2.1 C++的字符集

1. C++的字符集

C++中含有以下字符。

- 数字：0，1，2，3，4，5，6，7，8，9。
- 小写字母：a，b，…，y，z。
- 大写字母：A，B，…，Y，Z。
- 运算符：+，−，*，/，%，<，<=，=，>=，>，!=，==，<<，>>，&，|，&&，||，^，～，()，[]，{}，->，•，!，?，:，，，；，”，# 。
- 特殊字符：连字符或下划线。
- 不可印出字符：空白格（包括空格、换行和制表符）。

2. 词与词法规则

（1）标识符

标识符是对实体进行定义的一种定义符，由字母或下划线（或连字符）开头、后面跟字母或数字或下划线（或空串）组成的字符序列，一般有效长度是 8 个字符（ANSI C 标准规定 31 个字符），用来标识用户定义的常量名、变量名、函数名、文件名、数组名、数据类型名和程序等。

（2）关键字

关键字是具有特定含义，作为专用定义符的单词，不允许另作它用。下面列举一些常用的关键字。

auto	break	case	char	class	const	continue	default
do	default	delete	double	else	enum	explicit	extern
float	for	friend	goto	if	inline	int	long
mutable	new	operator	private	protected	public	register	return
short	signed	sizeof	static	static_cast	struct	switch	this
typedef	union	unsigned	virtual	void	while		

（3）运算符和分隔符

运算符是 C++语言实现加、减等各种运算的符号。

C++语言的分隔符主要是空格、制表和换行符。

（4）字符串

字符串是由双引号括起来的字符。如“China”、“C++ Program”等。

（5）常量

C++语言中的常量包括实型常量（浮点常量）和整型常量（十进制常量、八进制常量、十六进制常量）、浮点常量、字符常量和字符串常量。

（6）注释

注释是用来帮助阅读、理解及维护程序的。在编译时，注释部分被忽略，不产生目标代码。C++语言提供两种注释方式。一种是与 C 语言兼容的多行注释，用/*和*/分界。另一种是

单行注释，以“//”开头的表明本行中“//”符号后的内容是注释。举例如下。

例 1-1：一个简单的 C++程序。

```
#include <iostream.h>
void main()
{
     cout<<"This is my first C++ program.\n";        //输出 This is my first
                                                     //C++ program.
     /*输出
     This is my first C++ program.*/
}
```

3. C++程序结构的组成

C++程序结构的基本组成包括以下几个部分。

（1）预处理命令。C++提供了 3 类预处理命令：宏定义命令、文件包含命令以及条件编译命令。

（2）输入和输出。C++程序中总是少不了输入和输出的语句，实现与程序内部的信息交流。特别是屏幕输出的功能，几乎每个程序都要用到，使用它把计算机的结果显示在屏幕上。

（3）函数。C++的程序由若干个文件组成，每个文件又由若干个函数组成，因此，可以认为 C++的程序就是函数串，即由若干个函数组成。函数与函数之间是相对的，并且是并行的，函数之间可以调用。在组成一个程序的若干个函数中，必须有一个 main()。

（4）语句。语句是组成程序的基本单元。函数由若干条语句组成。空函数是没有语句的。语句由单词组成，单词间用空格符分隔。C++程序中的语句是以分号结束的。语句除了有表达式语句和空语句外，还有复合语句、分支语句、循环语句和转向语句等若干类。

（5）变量。多数程序都需要说明和使用变量。广义讲，对象包含了变量，即将变量称为一种对象。狭义讲，将对象看作是类的实例，对象是指某个类的对象。

（6）其他。除了以上讲述的 5 个部分，还有其他组成部分。例如，符号常量和注释信息也是程序的一部分。C++中都尽量把常量定义为符号常量，在 C++的程序中出现的是符号常量，该符号常量代表某个确定的常量值。

4. 书写格式

C++语言程序的书写格式自由度高、灵活性强、随意性大，如一行内可写一条语句，也可写几条语句；一个语句也可分写在多行。不过应采用适当的格式书写，便于人们阅读和理解。为了增加程序的可读性和利于理解，编写程序时按如下要点书写：

（1）一般情况下每个语句占用一行；

（2）不同结构层次的语句，从不同的起始位置开始，即在同一结构层次中的语句，缩进同样的字数；

（3）表示结构层次的大括弧，写在该结构化语句第一个字母的下方，与结构化语句对齐，并占用一行；

（4）适当加些空格和空行。

1.2.2 数据类型概述

数据类型是指定义了一组数据以及定义在这一组数据的操作，它是程序中最基本的元

素。程序的基本功能是处理数据，数据以变量或常量的形式存储，每个变量或常量都有数据类型。确定了数据类型，才能确定变量的空间大小及其操作。C++的数据类型十分丰富，大体上可分为基本类型、空类型、构造类型、指针类型和类类型5种，如图1-1所示。

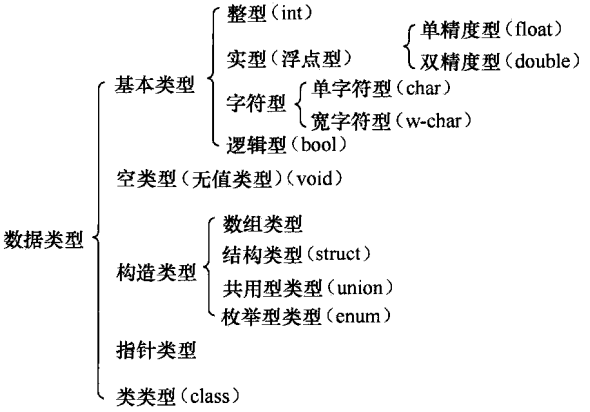

图1-1 C++数据类型

1．基本数据类型

基本数据类型有4种：整型（int）、浮点型（float）、字符型（char）和逻辑型（bool）。

整型数据在计算机内部一般采用定点表示法，用于存储整型量，如123、-7等。存储整数的位数依机器的不同而不同。

浮点型数据和整型数据不同的地方是浮点数采用浮点表示法，也就是说，浮点数的小数点的位置不同，给出的精度也不相同。

字符型数据表示单个字符，一个字符用一个字节存储。

逻辑型，也称布尔型，表示表达式的真和假。

2．空类型（void）

空类型（void）用于显示说明一个函数不返回任何值；还可以说明指向void类型的指针，说明以后，这个指针就可指向各种不同类型的数据对象。

3．构造类型

构造类型又称为组合类型，它是由基本类型按照某种规则组合而成的。

数组：是由具有相同数据类型的元素组成的集合。

结构体：是由不同的数据类型构成的一种混合的数据结构，构成结构体的成员的数据类型一般不同，并且在内存中分别占据不同的存储单元。

共用体：是类似于结构体的一种构造类型，与结构体不同的是构成共同体的数据成员共用同一段内存单元。

枚举：是将变量的值一一列举出来，变量的值只限于列举出来的值的范围内。

4．指针类型

指针类型变量用于存储另一变量的地址，不能用来存放基本类型的数据。它在内存中占据一个存储单元。

5．类类型

类是体现面向对象程序设计的最基本特征，也是体现C++与C最大的不同处。类是一个

数据类型，它定义的是一种对象类型，由数据和方法组成，描述属于该类型的所有对象的性质。

1.2.3 常量

常量是指在程序运行过程中其值不能改变的量。它分为字面常量和符号常量（又称标识符常量）。如 7、3.14、'a' 和 "abcdefg" 等都是字面常量，即字面本身就是它的值。符号常量是一个标识符，对应着一个存储空间，该空间中保存的数据就是该符号常量的值，这个数据是在定义符号常量时赋予的，以后是不能改变的。如 C++保留字中的 true 和 false 就是系统预先定义的两个符号常量，它们的值分别为数值 0 和 1。C++支持 5 种类型的常量：浮点型、整型、字符型、布尔型和枚举型。常量在程序中一般以自身的存在形式体现其值。常量具有类型属性，类型决定了各种常量在内存中占据存储空间的大小。

1．整型常量

整型常量表示通常意义上的整数，可以用十进制、八进制或十六进制表示。

（1）十进制常量

一般占一个机器字长，是一个带正负号的常数（默认情况下为正数），如+3、−7 等。

十进制整数由正号（+）或负号（−）开始、接着为首位非 0 的若干个十进制数字组成。若前缀为正号则为正数，若前缀为负号则为负数，若无符号则认为是正数。如 38、−25、+120、74 286 等都是符合书写规定的十进制整数。

当一个十进制整数大于或等于−2 147 483 648 即$-2^{31}-1$，同时小于等于 2 147 483 647 即$2^{31}-1$ 时，则被系统看作是 int 型常量；当在 2 147 483 648~4 294 967 295 即$2^{31}\sim 2^{32}-1$ 范围内时，则被看作是 unsigned int 型常量；当超过上述两个范围时，则无法用 C++整数类型表示，只有用实数（即带小数点的数）表示才能被有效地存储和处理。

（2）八进制常量

八进制整数由首位数字为 0 的、后接若干个八进制数字（借用十进制数字中的 0～7）组成。八进制整数不带符号位，隐含为正数。如 0、012、0377、04056 等都是八进制整数，对应的十进制整数依次为 0、10、255 和 2 094。

当一个八进制整数大于等于 0 同时小于或等于 017777777777 时，则称为 int 型常量；当大于或等于 020000000000 同时小于或等于 037777777777 时，则称为 unsigned int 型常量；超过上述两个范围的八进制整数则不使用，因为没有相对应的 C++整数类型。

（3）十六进制常量

以 0x 或 0X 开头，其后由若干个 0～9 的数字及 A～F（或小写 a～f）的字母组成，如 0x173、0x3af。同八进制整数一样，十六进制整数也均为正数。如 0x0、0X25、0x1ff、0x30CA 等都是十六进制整数，对应的十进制整数依次为 0、37、511 和 4 298。

当一个十六进制整数大于或等于 0 同时小于或等于 0x7FFFFFFF 时，则称为 int 型常量；当大于等于 0x80000000 同时小于或等于 0xFFFFFFFF 时，则称为 unsigned int 型常量；超过上述两个范围的十六进制整数没有相对应的 C++整数类型，所以不能使用它们。

（4）在整数末尾使用 u 和 l 字母

对于任何一种进制的整数，若后缀有字母 u（大、小写等效），则硬性规定它为一个无符号整型（unsigned int）数；若后缀有字母 l（大、小写等效），则硬性规定它为一个长整型（long int）数。在一个整数的末尾，可以同时使用 u 和 l，并且对排列无要求。如 25U、0327UL、

0x3ffbL、648LU 等都是整数，其类型依次为 unsigned int、unsigned long int、long int 和 unsigned long int。以下所列数是合法的：

```
375u          //无符号整数
12345UL       //无符号长整数
54321L        //长整数
13579ul       //无符号长整数
```

2．浮点数常量

浮点数也称为实型数，只能以十进制形式表示。共有两种表示形式：小数表示法和指数表示法。

（1）小数表示法

采用这种表示形式时，实型常量分为整数部分和小数部分。其中的一部分可在实际使用时省略，如 10.2、.2、2. 等。整数和小数部分不能同时省略。

（2）指数表示法

浮点表示的实数简称浮点数，它由一个十进制整数或定点数后接一个字母 e（大、小均可）和一个 1 至 3 位的十进制整数组成，字母 e 之前的部分称为该浮点数的尾数，之后的部分称为该浮点数的指数，该浮点数的值就是它的尾数乘以 10 的指数幂。如 3.23E5、+3.25e-8、2E4、0.376E-15、1e-6、-6.04E+12、0.43E0、96.e24 等都是符合规定的浮点数，它们对应的数值分别为 $3.23*10^5$、$3.25*10^{-8}$、2^4、$0.376*10^{-15}$、10^{-6}、$-6.04*10^{12}$、0.43、$96*10^{24}$ 等。

对于一个浮点数，若将它尾数中的小数点调整到最左边第一个非零数字的后面，则称它为规格化（或标准化）浮点数。如 21.6E8 和-0.074E5 是非规格化的，若将它们分别调整为 2.16E9 和-7.4E3 则都是规格化的浮点数。

（3）实数类型的确定

对于一个定点数或浮点数，C++自动按一个双精度数来存储，它占用 8 个字节的存储空间。若在一个定点数或浮点数之后加上字母 f（大、小写均可），则自动按一个单精度数来存储，它占用 4 个字节的存储空间。如 3.24 和 3.24f，虽然数值相同，但分别代表一个双精度数和一个单精度数。同理，-2.78E5 为一个双精度数，-2.78E5F 为一个单精度数。

3．字符常量与字符串常量

（1）字符常量

C++中的字符常量通常是用单引号括起的一个字符。在内存中，字符数据以 ASCII 码存储，如字符‘a’的 ASCII 码为 97。字符常量包括两类：一类是可显字符，如字母、数字和一些符号 ‘@’、‘+’ 等；另一类是不可显字符常量，如 ASCII 码为 13 的字符表示回车。

（2）转义字符

转义字符是特殊的字符常量，表示时一般以转义字符“\”开始，后跟不同的字符表示不同的特殊字符，表 1-1 列出了常用的特殊字符。

（3）字符串常量

字符串常量是由一对双引号括起来的零个或多个字符序列。

字符串可以写在多行，不过在这种情况下必须用反斜线“\”表示下一行字符是这一行字符的延续。

字符串常量实际上是一个字符数组，组成数组的字符除显示给出的外，还包括字符结尾处标识字符串结束的符号‘\0’，所以字符串“abc”实际上包含 4 个字符：‘a’、‘b’、‘c’

和‘\0’。需要注意的是‘a’和“a”的区别，‘a’是一个字符常量，在内存中占一个字节的存储单元；而“a”是一个字符串常量，在内存中占两个字节，除了存储‘a’以外，还要存储字符串结尾符‘\0’。

表 1-1　　常用的特殊字符

名　称	符　号
空字符（null）	\0
换行（newline）	\n
换页（formfeed）	\f
回车（carriage return）	\r
退格（backspace）	\b
响铃（bell）	\a
水平制表（horizontal tab）	\t
垂直制表（vertical tab）	\v
反斜线（backslash）	\\
问号（question mark）	\?
单引号（single quote）	\’
双引号（double quote）	\”

4．逻辑常量

逻辑常量是逻辑类型中的值，VC++用保留字 bool 表示逻辑类型。该类型只含两个值，即整数 0 和 1，用 0 表示逻辑假，用 1 表示逻辑真。在 VC++中还定义了这两个逻辑值所对应的符号常量 false 和 true。false 的值为 0，表示逻辑假；true 的值为 1，表示逻辑真。

由于逻辑值是整数 0 和 1，所以它能够像其他整数一样出现在表达式里，参与各种整数运算。

5．枚举常量

枚举常量是枚举类型中的值，即枚举值。枚举类型是一种用户定义的类型，只有用户在程序中定义它后才能被使用。用户通常利用枚举类型定义程序中需要使用的一组相关的符号常量。枚举类型的定义格式为：

```
enum <枚举类型名> {<枚举表>};
```

它是一条枚举类型定义语句，该语句以 enum 保留字开始，接着为枚举类型名，它是用户命名的一个标识符，以后就直接使用它表示该类型。枚举类型名后为该类型的定义体，它由一对花括号和其中的枚举表所组成。枚举表为一组用逗号分开的由用户命名的符号常量，每个符号常量又称为枚举常量或枚举值。如：

```
(1) enum color{red, yellow, blue};
(2) enum day{Sun, Mon, Tues, Wed, Thur, Fri, Sat};
```

第一条语句定义了一个枚举类型 color，用来表示颜色，它包含 3 个枚举值 red、yellow 和 blue，分别代表红色、黄色和蓝色。

第二条语句定义了一个枚举类型 day，用来表示日期，它包含 7 个枚举值，分别表示星期日、星期一至星期六。

一种枚举类型被定义后，可以像整型等预定义类型一样使用在允许出现数据类型的任何