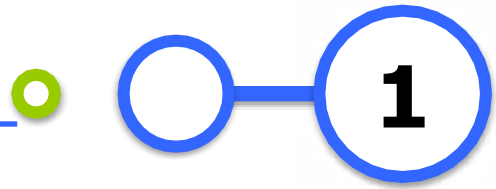




第4章 选择构造程序设计



本章要点

- 条件体现式
- if语句
- switch语句
- 选择构造程序设计





4.1.1 关系运算符及其优先顺序

1. 关系运算符

所谓“关系运算”实际上就是“比较运算”，即将两个数据进行比较，鉴定两个数据是否符合给定的关系。

关系运算符均为二目运算符，共有下列6种：

- > 不小于
- < 不不小于
- >= 不小于或等于
- <= 不不小于或等于
- == 等于
- != 不等于1



4.1.1 关系运算符及其优先顺序

2. 关系运算符的优先级

- (1) 在关系运算符中，前4个优先级相同，后2个相同；且前4个高于后2个；结合方向均为自左至右。
- (2) 关系运算符的优先级低于算术运算符，但高于赋值运算符。



关系体现式

1. 关系体现式的概念

所谓关系体现式是指用关系运算符将两个体现式连接起来进行关系运算的式子。

例如，下面的关系体现式都是正当的：

$a+b>c-6$; $(a=3)<=(b=5)$; $'a'>='b'$; $(a>b)==(b>c)$;

2. 关系体现式的值——逻辑值（“真”或“假”）

因为C语言没有逻辑型数据，所以，用整数“1”表达“逻辑真”，用整数“0”表达“逻辑假”。



逻辑运算符及其优先顺序

1. 逻辑运算符

C语言提供3种逻辑运算符:

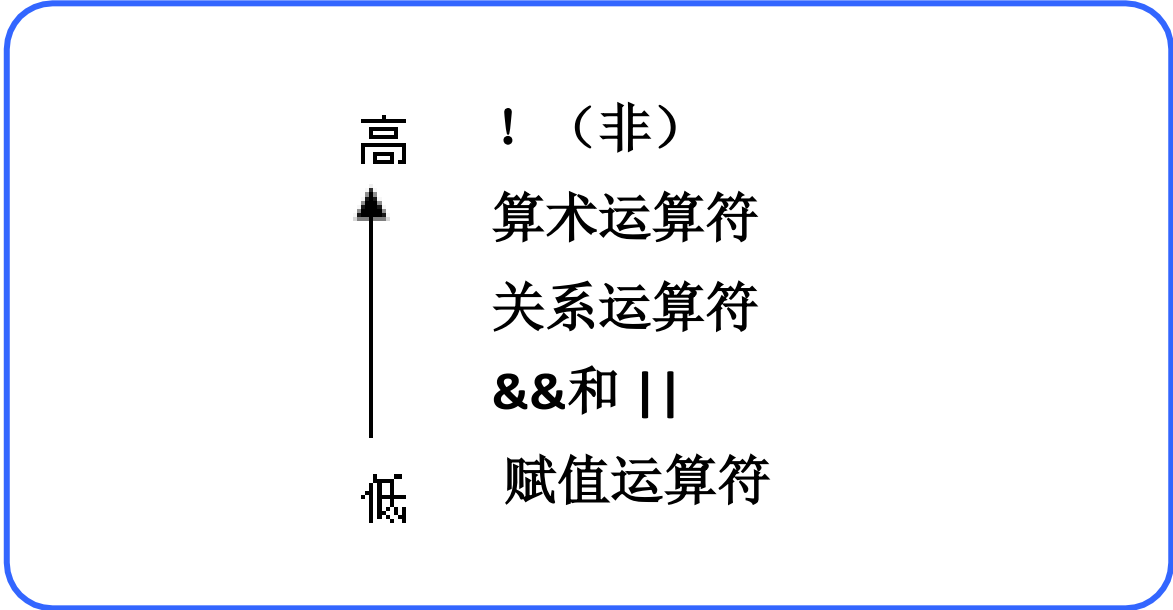
- **&&** 逻辑与（相当于“同步”）
- **||** 逻辑或（相当于“或者”）
- **!** 逻辑非（相当于“否定”）



逻辑运算符及其优先顺序

2. 逻辑运算符的运算优先级

- (1) 在逻辑运算符中，逻辑非! 的优先级最高，逻辑与&&次之，逻辑或||最低。
- (2) 与其他种类运算符的优先关系





逻辑体现式

1. 逻辑体现式的概念

所谓逻辑体现式是指，用逻辑运算符将1个或多种体现式连接起来，进行逻辑运算的式子。在C语言中，能够用逻辑体现式表达多种条件的组合。

例如， $(a+b)\&\&(c\leq 0)$

逻辑体现式的值也是一种逻辑值，“真”或“假”。

2. 逻辑量的真假鉴定

C语言用整数“1”表达“逻辑真”，用“0”表达“逻辑假”。但在判断一种体现式或变量的“真”或“假”时，是以0和非0为根据：假如为0，则鉴定为“逻辑假”；假如为非0，则鉴定为“逻辑真”。



if语句的三种形式

1. 第一种形式：单分支if语句

格式：if (体现式) 〈语句〉；

功能：首先计算体现式的值。若体现式的值为“真”（为非0），则执行语句；若体现式的值为“假”（为0），则直接转到此if语句的下一条语句去执行。其流程图如图4-1(a)所示。

例如：

```
if (x>y) printf ("%d", x) ;
```

假如 $x>y$ 为真，则打印 x 的值，不然执行下面的语句。



if语句的三种形式

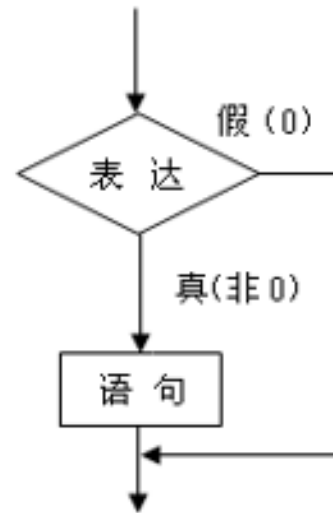
2. 第二种形式: 双分支if语句

格式: `if (体现式) <语句1>;`
`else <语句2>;`

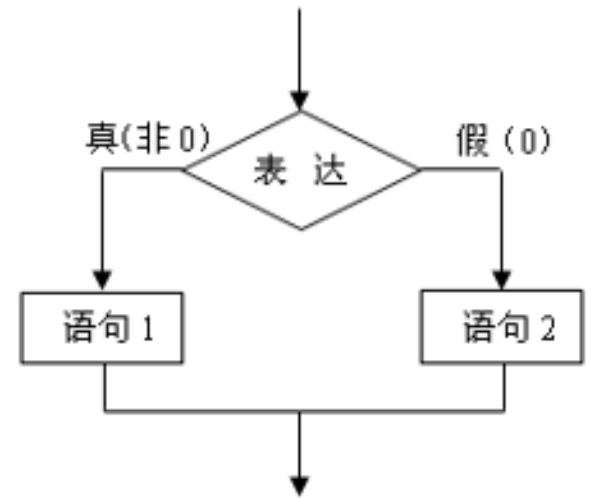
功能: 首先判断体现式的值, 若体现式的值为“真”(非0), 则执行语句1; 不然, 执行语句2。其流程图如图4-1 (b) 所示。

例如:

```
if (x>y) printf ("%d", x);  
else printf ("%d", y);
```



(a)



(b)



if语句的三种形式

3. 第三种形式：多分支if语句

格式：

```
if (体现式1) 〈语句1〉 ;  
else if (体现式2) 〈语句2〉 ;  
...  
else if (体现式n) 〈语句n〉 ;  
else 〈语句n+1〉 ;
```

功能：从体现式1的值开始进行判断，当出现某个体现式的值为真时，则执行其相应分支的语句，然后跳出整个if语句，执行后续语句。若全部体现式的值都为“假”（为0），则执行语句n+1。



if语句的三种形式

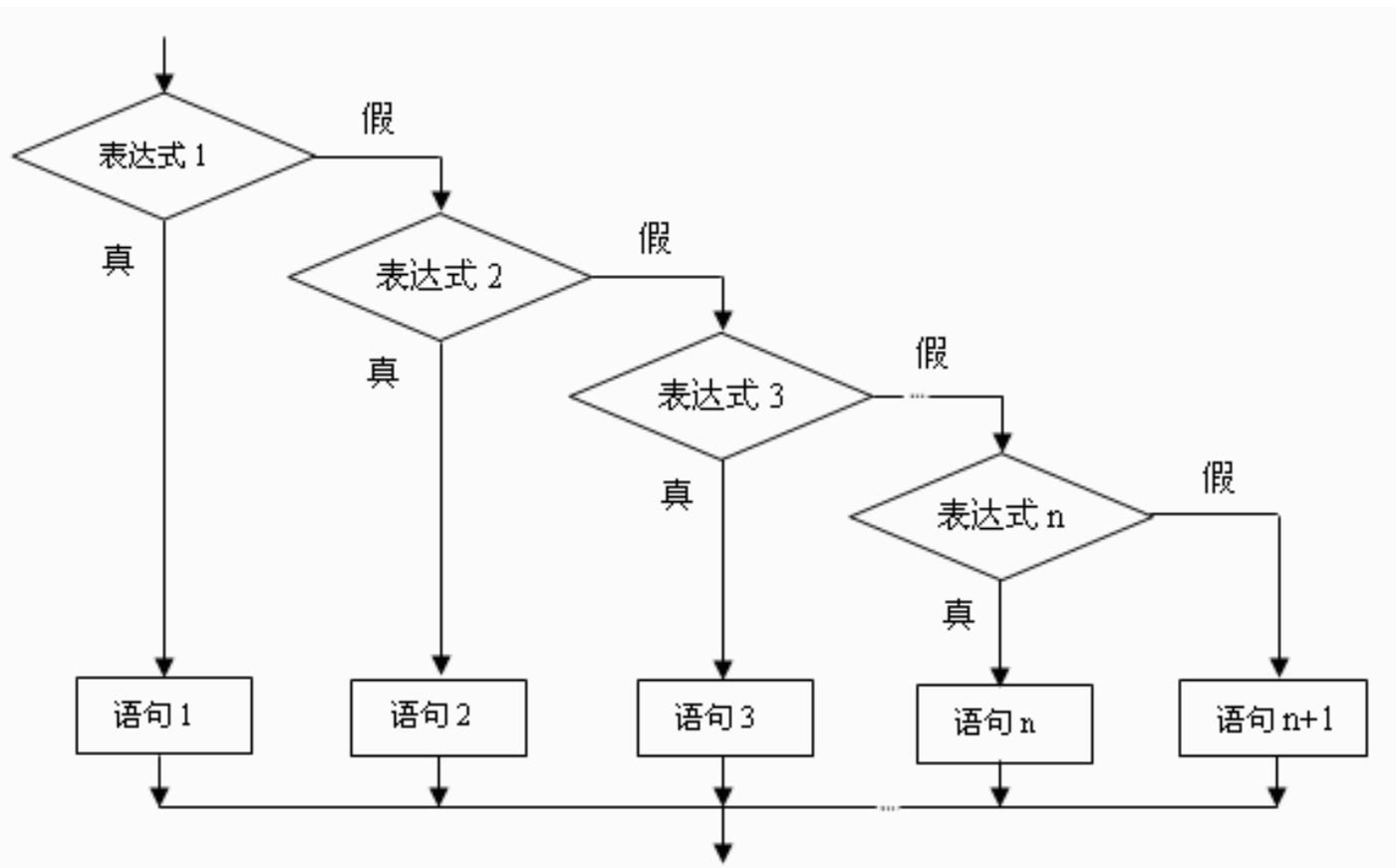


图4-2 第三种形式if语句的流程图表达



if语句的嵌套

一种if语句又包括一种或多种if语句，称为if语句的嵌套。在if语句中能够根据需要，用if语句的三种形式进行相互嵌套。一般形式如下：

1. if ()

```
if ( ) <语句1>
else <语句2> } 内嵌 if
```

else

```
if ( ) <语句3>
else <语句4> } 内嵌 if
```



if语句的嵌套

2. if ()

if () <语句1> 内嵌if单分支

else

if () <语句2>
else <语句3> } 内嵌 if



条件运算符 (? :)

1. 条件运算符的格式:

变量 = 〈体现式1〉 ? 〈体现式2〉 : 〈体现式3〉 ;

执行过程: 当体现式1的值为“真”时, 取体现式2的值赋给变量; 当体现式1的值为“假”时, 取体现式3的值赋给变量。

即: 不论体现式1为“真”还是为“假”, 都执行一种赋值语句且给同一种变量赋值。能够看出, 用条件运算符“? : ”的功能, 能够实现if 语句的第2种格式。



条件运算符 (? :)

2. 分析:

“体现式3”又是一种条件体现式，因条件体现式其结合性为“从右到左”，所以，应该先计算右边的体现式3，公式相当于：

$q=w>x?w:(y<z?y:z);$

执行环节如下：

体现式 $y<z?y:z$ ，因为 $y<z$ 的值为“真”，所以取 y 的值3。

体现式 $q=w>x?w:3$ ；因为 $w>x$ 的值为“假”，所以取3赋值给变量 q 。

运营成果： $q=3$



if语句一般合用于两个分支的情况，即在两个分支中选择其中一种执行。尽管能够经过if语句的嵌套形式来实现多路选择的目的，但这么做的成果使得if语句的嵌套层次太多，降低了程序的可读性。C语言中的switch语句，提供了更以便地进行多路选择的功能。

一般形式：

```
switch (体现式)
{
    case 常量体现式1 : 语句1 ; [break;]
    case 常量体现式2 : 语句2 ; [break;]
    ...
    case 常量体现式n : 语句n ; [break;]
    default: 语句n+1 ;
}
```


以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/936225003021010241>