

# 计算机导论与程序设计 [CS006001-60]

段江涛

机电工程学院



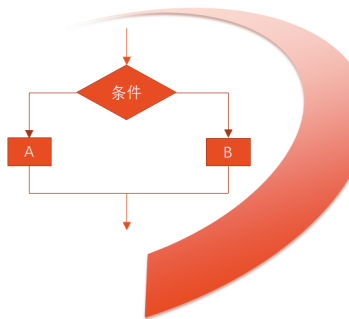
2019 年 9 月

# lecture-6 主要内容

## 选择结构程序设计

- 1 选择结构和条件判断
- 2 if 语句的一般形式
- 3 关系运算符及其优先次序
- 4 逻辑运算符
- 5 条件运算符和条件表达式
- 6 数学表达式与 C 语言表达式的不同
- 7 用 switch 语句实现多分支选择结构

# 选择结构和条件判断



## C 语言有两种选择语句

- if 语句, 用来实现**两个分支**的选择结构
- switch 语句, 用来实现**多分支**的选择结构

# if(条件表达式){ 表达式为真 (非 0) 时执行语句; }

```
#include<stdio.h> // standard input/output编译预处理指令

int main() // 主函数
{ // 函数开始标志

    int a=10; // 定义变量a为整型数值，定义变量时，可以指定变量的初值
    if(a>=10)
    {
        printf("a>=10\n"); // \n为换行符
    }
    else
    {
        printf("a<10\n"); // \n为换行符
    }

    return 0; // 函数执行完毕返回函数值0
} // 函数结束标志
```

[例 4.1 p84] 求  $ax^2 + bx + c = 0$  方程的根。 $a, b, c$  由键盘输入。

```
#include<stdio.h>

#include<math.h> // 数学库函数

int main()
{
    double a,b,c,x1,x2,delta;
    scanf("%lf%lf%lf",&a,&b,&c);
    if(b*b-4*a*c < 0)
    { printf("This equation hasn't real roots!\n"); }
    else
    {
        delta = sqrt(b*b-4*a*c);
        x1 = -b + delta/(2*a); x2 = -b - delta/(2*a);
        printf("x1=%.2lf,x2=%.2lf\n",x1,x2);
    }
    return 0;
}
```

[例 4.2 p85] 输入两个实数, 按由小到大的顺序输出这两个数。

```
#include<stdio.h>

int main()
{
    float a,b,t;
    scanf("%f%f",&a,&b);
    //不好: scanf("%f,%f",&a,&b);
    if(a>b)
    { //将a和b的值互换
        t=a;
        a=b;
        b=t;
    }
    printf("%.2f,%.2f\n",a,b);
    return 0;
}
```

### 两个变量值的互换

a=b; //把变量 b 的值赋给变量 a, a 的值等于 b 的值

b=a; //再把变量 a 的值赋给变量 b, 变量 b 值没有改变

因此, 为了实现互换, 必须借助于第三个变量。

[例 4.3 p86] 输入 3 个数 a, b, c, 要求按由小到大的顺序输出。

```
#include<stdio.h>

int main()
{
    float a,b,c,t;
    scanf("%f%f%f",&a,&b,&c); //不好: scanf("%f,%f,%f",&a,&b,&c);
    if(a>b)
    {
        t=a; a=b; b=t; //借助变量t, 实现变量a和变量b互换值
    } //互换后, a小于或等于b
    if(a>c)
    {
        t=a; a=c; c=t; //借助变量t, 实现变量a和变量c互换值
    } //互换后, a小于或等于c
    if(b>c) //还要
    {
        t=b; b=c; c=t; //借助变量t, 实现变量b和变量c互换值
    } //互换后, b小于或等于c
    printf("%.2f,%.2f,%.2f\n",a,b,c); //顺序输出a,b,c的值
    return 0;
}
```

# if(条件表达式){ 表达式为真 (非 0) 时执行语句; }

条件表达式: 关系表达式; 逻辑表达式; 数值表达式。

## 形式 1(无 else)

```
// 形式1(无else)
if(条件表达式)
{
    多条语句(复合语句);
}
```

## 形式 2

```
// 形式2
if(条件表达式)
{
    多条语句(复合语句);
}
else
{
    多条语句(复合语句);
}
```

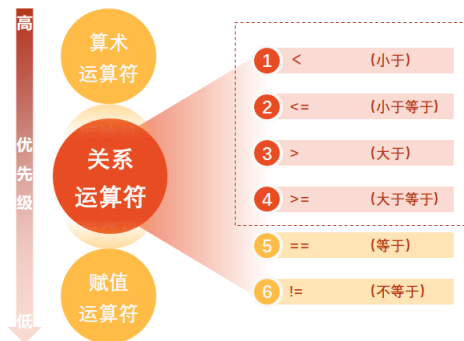
## 形式 3(排除式)

```
// 形式3(排除式)
if(条件表达式1)
{
    多条语句(复合语句);
}
else if(条件表达式2) //可多个
{
    多条语句(复合语句);
}
else
{
    多条语句(复合语句);
}
```



# 关系运算符及其优先次序

```
int a=5,b=10,c=20; //以int为例
if (a<b+c) // 相当于a<(b+c)
{ ... }
if (a<=b+c)
{ ... }
if (a>b+c)
{ ... }
if (a>=b+c)
{ ... }
if (a==b+c) //a是否等于 (b+c) , 与a=(b+c)
            不同
{ ... }
if (a!=b+c) // a不能于 (b+c)
{ ... }
```



分析:

```
if (a>b==c) { ... }
if (a=b>c) { ... }
```

# 关系表达式的值, 非 0 即真

## 关系表达式

- 用关系运算符将两个数值或数值表达式连接起来的式子, 称为关系表达式。
- 关系表达式的值是一个逻辑值, 即“真”或“假”。
- 在 C 的逻辑运算中, 以“1”代表“真”, 以“0”代表“假”。

```
int a=3, b=2, c=1, d1, d2;
```

```
d1 = a > b; // d1=1
```

```
d2 = a > b > c; // 自左至右结合, d2=0
```

```
if (d1)
```

```
{ printf("执行此语句"); }
```

```
if (d2)
```

```
{ printf("不执行此语句"); }
```

```
if (d1 = a > b) // d1的值就是表达式的值
```

```
{ printf("执行此语句"); }
```

```
if (d2 = a > b > c) // d2的值就是表达式的值
```

```
{ printf("不执行此语句"); }
```

# 逻辑运算符

```
int a=5,b=10,c=0; //以int为例
```

```
if(!a) // 逻辑非 (NOT), a是非0, 所以!a的值是0
```

```
{ ... }
```

```
if(a && b) // 逻辑与 (AND), a,b均为非0, 所以(a && b)的值为1
```

```
{ ... }
```

```
if(a || c) // 逻辑或 (OR), a,c之一是非0, 即为真
```

```
{ ... }
```

# 逻辑运算符真值表

a	b	!a	!b	a&&b	a  b
真(非0)	真(非0)	假(0)	假(0)	真(1)	真(1)
真(非0)	假(0)	假(0)	真(1)	假(0)	真(1)
假(0)	真(非0)	真(1)	假(0)	假(0)	真(1)
假(0)	假(0)	真(1)	真(1)	假(0)	假(0)

- “&&” 和 “||” 是双目运算符,要求有两个运算对象(操作数);

“!” 是单目运算符,只要有一个运算对象

- 由高到低优先次序: !(非)→&&(与)→|| (或);

逻辑运算符中的 “&&” 和 “||” 低于关系运算符, “!” 高于算术运算符

- 逻辑运算结果不是 0 就是 1,不可能是其他数值。

而运算对象可以是 0(假) 或任何非 0 的数值(按“真”对待)

## 逻辑运算示例 (1)

判别用 `year` 表示的某一年是否闰年, 可以用一个逻辑表达式来表示。闰年的条件是符合下面二者之一: (1) 能被 4 整除, 但不能被 100 整除。(2) 能被 100 整除, 又能被 400 整除。

```
int year;
scanf("%d",&year);
\\ 闰年
if(year%4 == 0 && year%100 != 0)
{ printf("%d是闰年\n", year); }
else if(year%100 == 0 && year%400 == 0)
{ printf("%d是闰年\n", year); }
else
{ printf("%d不是闰年\n", year); }
```

## 逻辑运算示例 (2)

判别用 year 表示的某一年是否闰年,可以用一个逻辑表达式来表示。闰年的条件是符合下面二者之一: (1) 能被 4 整除,但不能被 100 整除。(2) 能被 100 整除,又能被 400 整除。

```
int year, flag = 'N';
scanf("%d",&year);
\\ 闰年
if(year%4 == 0 && year%100 != 0)
{ flag = 'Y'; }
else
{
    if(year%100 == 0 && year%400 == 0)
    { flag = 'Y'; }
}
if(flag == 'Y')
{ printf("%d是闰年\n", year); }
else
{ printf("%d不是闰年\n", year); }
```

## 逻辑运算示例 (3)

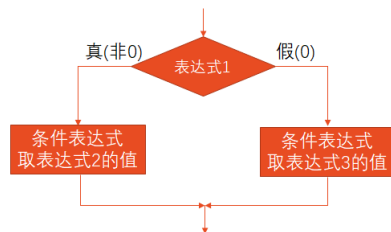
判别用 `year` 表示的某一年是否闰年, 可以用一个逻辑表达式来表示。闰年的条件是符合下面二者之一: (1) 能被 4 整除, 但不能被 100 整除。(2) 能被 100 整除, 又能被 400 整除。

```
int year;
scanf("%d",&year);
\\ 闰年
if((year%4 == 0 && year%100 != 0)|| (year%100 == 0 && year%400 == 0))
{ printf("%d是闰年\n", year); }
else
{ printf("%d不是闰年\n", year); }
```

# 条件运算符和条件表达式

```
int a,b;
scanf("%d%d",&a,&b);
if(a>b)
{ max = a; }
else
{ max = b; }
// 等效为
max = (a>b) ? a : b;
// 或
a>b ? (max=a) : (max=b);
// 甚至用在语句中
printf("%d\n",a>b ? a : b);
```

表达式 1 ? 表达式 2 : 表达式 3





## 例: 大写转小写字母

[例 4.4, p96] 输入一个字符, 判别它是否为大写字母, 如果是, 将它转换成小写字母; 如果不是, 不转换。然后输出最后得到的字符。

```
char ch;

scanf("%c", &ch);

ch = (ch>='A' && ch<='Z') ? (ch+32) : ch;

// 等效于

if(ch>='A' && ch<='Z')
{
    ch = ch+32; // 可简写为 ch += 32;
}

printf("ch=%c\n", ch);
```

# 数学表达式与 C 语言表达式的不同

```
int a = 10;

if (20 <= a && a <= 30) // 表达式的值为假 (0)
{ ... }

if (20 <= a <= 30) // 表达式为真 (1)
{ ... }

// 类似的

// if (a==20) 与 if (a=20) 意义不同.

if (a==20) // 表达式的值是假 (0)
{ ... }

if (a=20) // 表达式的值是10, 非0, 表示为真, 并且a被赋值为20
{ ... }
```

# 用 switch 语句实现多分支选择结构

switch(int 或 char 型表达式)

```
int a;
scanf("%d",&a)
swach(a)
{
    case 10: 多条语句1;
        break;
    case 20: 多条语句2;
        break;
    case 30: 多条语句3;
        break;
    default: 多条语句4;
}
```

```
int a;
scanf("%d",&a)
if(a == 10)
{ 多条语句1; }
else if(a == 20 )
{ 多条语句2; }
else if(a == 30)
{ 多条语句3; }
else
{ 多条语句4; }
```

```
char a; // 或 int a;
```

```
scanf("%c",&a)
```

```
swach(a)
```

```
{
    case 'A':
    case 'a': 多条语句1;
        break;
    case 'B':
    case 'b': 多条语句2;
        break;
    case 'C':
    case 'c': 多条语句3;
        break;
    default: 多条语句4;
}
```

```
char a; // 或 int a;
```

```
scanf("%d",&a)
```

```
if(a == 'A' || a == 'a')
```

```
{ 多条语句1; }
```

```
else if(a == 'B' || a == 'b')
```

```
{ 多条语句2; }
```

```
else if(a == 'C' || a == 'c')
```

```
{ 多条语句3; }
```

```
else
```

```
{ 多条语句4; }
```

[例 4.10,p99] 运输公司对用户计算运输费用。路程越远,运费越低。标准如下:

$s < 250$	没有折扣
$250 \leq s < 500$	2% 折扣
$500 \leq s < 1000$	5% 折扣
$1000 \leq s < 2000$	8% 折扣
$2000 \leq s < 3000$	10% 折扣
$3000 \geq s$	15% 折扣

```
int c,s; //c是分类整数, s是距离
float p,w,d,f; //单价,重量,折扣,运费
// 运费 f = p*w*s*(1-d%)
if(s>=3000) { c = 12; } else {c = s/250; }
switch(c) {
    case 0: d=0; break;
    case 1: d=2; break;
    case 2: case 3: d=5; break;
    case 4: case 5: case 6: case 7:
        d=8; break;
    case 8: case 9: case 10: case 11:
        d=10 break;
    case 12: d=15; break;
}
f = p*w*s*(1-d/100); printf("%.2f\n",f);
```

欢迎批评指正！