

# 第3章 选择结构

# 本章目标

- ◆ **知识目标**：掌握if、if-else、if-else if-else、switch等分支结构，掌握关系表达式和逻辑表达式，以及条件表达式。
- ◆ **能力目标**：具备使用if和switch分支结构来解决现实生活中常见的选择问题。
- ◆ **素养目标**：培养学生解决复杂问题的工程思维能力，通过C语言编程实例培养学生追求卓越的精神和用所学知识服务社会的意识。

## 3.1 一起来评优

**例1：**假设学校这样规定：学生百分制的平均分超过90分，可认定为成绩优秀。用C语言编写一个小程序，根据平均分评判某个学生成绩是否优秀。

```
#include<stdio.h>
int main(){
    int score;
    printf("请输入平均分：");
    scanf("%d",&score);
    if(score>90)
        printf("您的成绩为优秀\n");
    return 0;
}
```

# 单分支选择结构

## (1) 单分支if语句的格式

◆ 单分支if语句的一般格式为：说明：

```
if(表达式)
```

```
{
```

```
语句组
```

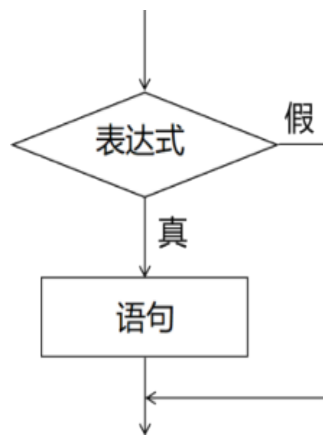
```
}
```

(1) 表达式可以是任意表达式，用来判断条件是否满足，表达式的值为0，表示条件不满足，表达式的值非0，表示条件满足；

(2) 语句组可以是任意合法的C语言语句，如果是单条语句，可以省略大括号。

# 单分支选择结构

## (2) 单分支if语句的执行流程

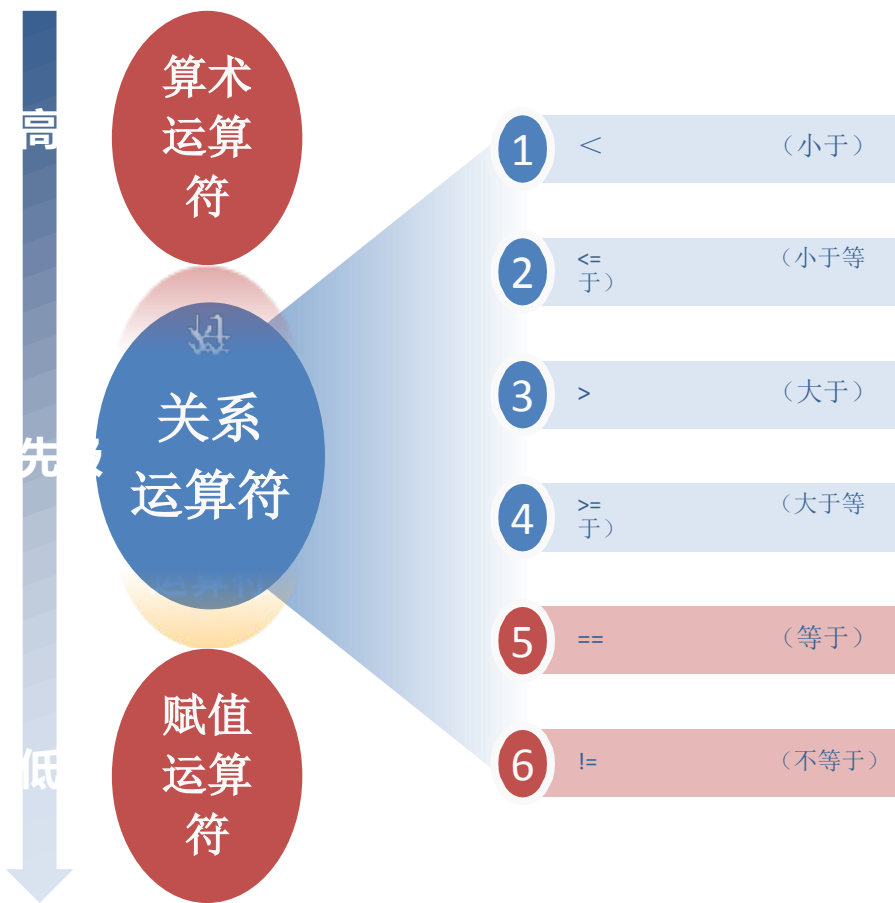


# (3) 关系运算

## 1) 关系运算符

- ✓ 关系运算是对两个操作数进行比较的运算，C语言提供了6种关系运算符：
  - ✓  $<$ ：表示“小于”，等价于数学中的 $<$ ；
  - ✓  $<=$ ：表示“小于等于”，等价于数学中的 $\leq$ ；
  - ✓  $>$ ：表示大于，等价于数学中的 $>$ ；
  - ✓  $>=$ ：表示“大于等于”，等价于数学中的 $\geq$ ；
  - ✓  $==$ ：表示“等于”，等价于数学中的 $=$ ；
  - ✓  $!=$ ：表示“不等于”，等价于数学中的 $\neq$ 。

# 关系运算符及其优先次序



- 前 4 种关系运算符的优先级别相同，后 2 种也相同。前 4 种高于后 2 种。
- 关系运算符的优先级低于算术运算符。
- 关系运算符的优先级高于赋值运算符。

$c > a + b$  等效于  $c > (a + b)$  (关系运算符的优先级低于算术运算符)

$a > b == c$  等效于  $(a > b) == c$  (大于运算符  $>$  的优先级高于相等运算符  $==$ )

$a == b < c$  等效于  $a == (b < c)$  (小于运算符  $<$  的优先级高于相等运算符  $==$ )

$a = b > c$  等效于  $a = (b > c)$  (关系运算符的优先级高于赋值运算符)

## 2) 关系表达式

- ◆ 由关系运算符构成的表达式称为关系表达式。例如，下列的表达式都是合法的关系表达式：
  - ✓  $a+b < c*d$
  - ✓  $'a' + 1 \leq b$
  - ✓  $i++ == j$
- ◆ 需要注意的是，关系表达式的值只能为“真”或“假”，其中“真”用“1”表示，而“假”用“0”表示。例如：
  - ✓  $2 > 1$ 的值为“真”，即该表达式的值为1。
  - ✓  $b=2; c=1; \text{求 } a=b \leq c \text{ 的值}$ ，由于 $b \leq c$ 的值为“假”，即为 $a=0$ 。
- ◆ 字符以它对应ASCII码值参与运算。例如：
  - ✓  $'a' > 'b'$  (ASCII码 $97 > 98$ )表达式值为“假”。



## 3.2一起来评优

**例2：若学生平均分超过90分则输出成绩为优秀；否则，提醒学生要努力达到优秀。**

```
#include<stdio.h>
int main(){
    int score;
    printf("请输入平均分：");
    scanf("%d",&score);
    if(score>=90)
        printf("成绩为优秀\n");
    else
        printf("成绩没有达到优秀，仍需继续努力！\n");
    return 0;
}
```

## 3.2一起来评优

### 双分支结构if-else语句

#### (1) 双分支结构if-else语句的格式

if(条件表达式)

{

    语句块1

}

else

{

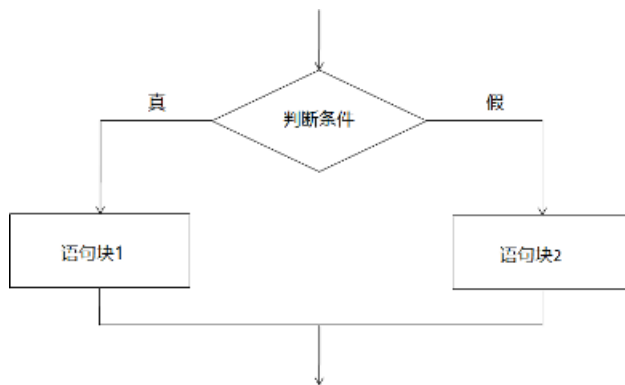
    语句块2

}

## 3.2 一起来评优

### 双分支结构if-else语句

#### (2) 双分支结构if-else语句的执行流程



## 3.3 成绩等级评一评

【例3-3】百分制转五分制，学生考试成绩达到90分及以上输出“你的成绩为A级！”，成绩 $\geq 80$ 分并且 $< 90$ 分输出“你的成绩为B级！”，成绩 $\geq 70$ 分并且 $< 80$ 分输出“你的成绩为C级”，成绩 $\geq 60$ 分并且 $< 70$ 分输出“你的成绩为D级！”，对于 $< 60$ 分的，输出“你的成绩为E级！”。你能编写这样一个百分制转五分制的程序吗？

```
#include<stdio.h>
int main(){
    int score;
    printf("请输入你的成绩:");
    scanf("%d",&score);
    if(score>=90)
        printf("你的成绩为A级！ \n");
    else if(score>=80)
        printf("你的成绩为B级！ \n");
    else if(score>=70)
        printf("你的成绩为C级！ \n");
```

```
    else if(score>=60)
        printf("你的成绩为D级！ \n");
    else
        printf("你的成绩为E级！ \n");
    return 0;
}
```

# 多分支结构if-else if-else语句

## ( 1 ) 多分支结构if-else if-else 语句的格式

```
if(表达式1)
```

```
{  
  语句块1
```

```
}
```

```
else if(表达式2)
```

```
{  
  语句块2
```

```
}
```

```
else if(表达式3)
```

```
{  
  语句块3
```

```
}
```

```
.....
```

```
else if(表达式n-1)
```

```
{  
  语句块n-1
```

```
}
```

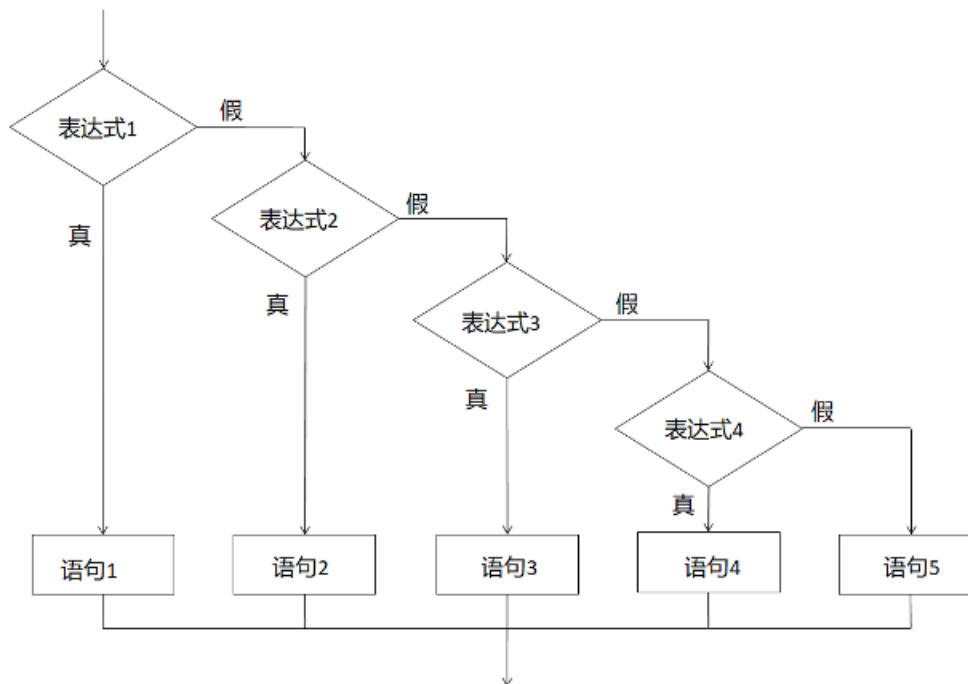
```
else
```

```
{  
  语句块n
```

```
}
```

# 多分支结构if-else语句

## (2) 多分支结构if-else if-else语句的执行流程



# 多分支结构if-else if-else语句

## ( 1 ) 多分支结构if-else if-else 语句的格式

```
if(表达式1)
```

```
{  
语句块1
```

```
}
```

```
else if(表达式2)
```

```
{  
语句块2
```

```
}
```

```
else if(表达式3)
```

```
{  
语句块3
```

```
}
```

```
.....
```

```
else if(表达式n-1)
```

```
{  
语句块n-1
```

```
}
```

```
else
```

```
{  
语句块n
```

```
}
```

# if语句的嵌套

## ( 1 ) 嵌套的if语句格式

```
if(表达式1)    /*外层if语句*/  
if(表达式2)   }  
    语句1      /*内层if语句*/  
    else  
    语句2  
else  
if(表达式3)   }  
    语句3      /*内层if语句*/  
    else  
    语句4
```



# if语句的嵌套

## (2) 嵌套的if语句分类

```
if (表达式) 语句1  
[ else 语句2 ]
```

“表达式”可以是关系表达式、逻辑表达式，甚至是数值表达式

方括号内的部分(即else子句)为可选的，既可以有，也可以没有

语句1和语句2可以是一个简单的语句，也可以是一个复合语句，还可以是另一个if语句

形式1 没有else子句部分

```
if(表达式) 语句1
```

形式2 有else子句部分

```
if(表达式)  
    语句1  
else  
    语句2
```

形式3 在else部分又嵌套了多层的if语句

```
if(表达式1)    语句1  
else if(表达式2) 语句2  
else if(表达式3) 语句3  
□□:           :  
else if(表达式m) 语句m  
else           语句m+1
```

# 选择结构的嵌套

```
if()  
    if() 语句1  
    else 语句2  
else  
    if() 语句3  
    else 语句4
```

内嵌if

内嵌if

注意

**if与else的配对关系。**

else总是与它上面的最近的未配对的if配对。

如果if与else的数目不一样，为实现程序设计者的思想，可以加花括号来确定配对关系。

```
if()  
    if() 语句1  
else  
    if() 语句2  
else 语句3
```

程序员把else写在与第1个if(外层if)同一列上，意图是使else与第1个if对应，但实际上else是与第2个if配对，因为它们相距最近。

```
if()  
{  
    if() 语句1  
}  
else  
    语句2
```

内嵌if

# if语句的嵌套

## 例3.4 使用嵌套if语句求解一元二次方程 $ax^2+bx+c=0$ 的根。

```
#include <stdio.h>
#include <math.h>
int main(){
    /*delta判别式,RealPart实部,
    ImagPart虚部 */
    float a,b,c,delta,x1,x2,m,n,
        RealPart,ImagPart;
    printf("请输入一元二次方程系数: a,b,c\n");
    scanf("%f,%f,%f",&a,&b,&c);
    delta=b*b-4*a*c;
    m=-b/(2*a);
    if (delta>=0)        /*外层if语句*/
        if(delta>0)
            /*两个不等的实根, 内层if语句*/
```

```
        n=sqrt(delta)/(2*a);
            x1=m+n;
            x2=m-n;
            printf("x1=%6.3f x2=%6.3f\n",x1,x2);
        }
    else /*两个相等的实根*/
        printf("x1=x2=%6.3f\n",m);
    else{ /*两个不相等的虚根*/
        RealPart=m;
        ImagPart=sqrt(-delta)/(2*a);
        printf("x1=%6.3f+%6.3fi\n",RealPart,
            ImagPart);
        printf("x2=%6.3f-%6.3fi\n",
            RealPart,ImagPart);
    }
    return 0;
}
```

## 3.4 今年究竟多少天？

### 例3-5 判断某一年是否为闰年。

```
#include <stdio.h>
int main()
{
    int year ;
    scanf("%d",&year);
    if((year%4 == 0 && year%100 !=0) || year%400 == 0)
        printf("%d年是闰年\n",year);
    else
        printf("%d年不是闰年\n",year);
    return 0;
}
```

# 逻辑运算符及其优先次序

运算符	含义	举例	说明
!	逻辑非 (NOT)	!a	如果a为假，则!a为真;如果a为真，则!a为假
&&	逻辑与 (AND)	a && b	如果a和b都为真，则结果为真，否则为假
	逻辑或 (OR)	a    b	如果a和b有一个以上为真，则结果为真，二者都为假时，结果为假

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/356134212051010221>