



C语言程序设计基础

林川



常见疏漏、错误

- scanf没有取地址

```
int x;
```

```
scanf("%d", x);
```

× 之前遗漏了 &

- scanf类型不匹配

```
int x;
```

```
scanf("%f", &x);
```

× 是整型, 应该用%d



常见疏漏、错误

- 画蛇添足 :for语句多了分号

```
for( i=0; i<n; i++ );  
sum = sum +i;
```

- 复合语句没有括号

```
for( i=0; i<n; i++ )  
item = 1.0/(2*i-1);  
sum = sum + item;
```



常见疏漏、错误

- 变量没有初始化

```
int i, sum;
```

```
sum = 0;
```

```
for( i=1; i<100; i++ );
```

```
sum = sum +i;
```



常见疏漏、错误

- 整数除法

```
int i;
```

```
    . . .
```

```
for( i=1; i<n; i++ )
```

```
    sum = 1/i;
```

应该用 $1.0/i$

```
int farh, cel;
```

```
cel = 5/9*farh - 5/9*32;
```

应该先做乘法： $5*farh/9$



第三章 分支结构

1. 分支结构
2. 字符类型
3. 关系运算
4. 逻辑运算
5. switch 语句
6. break语句



1) 分支结构

- if-else语句
- if语句
- 嵌套的if/if-else语句

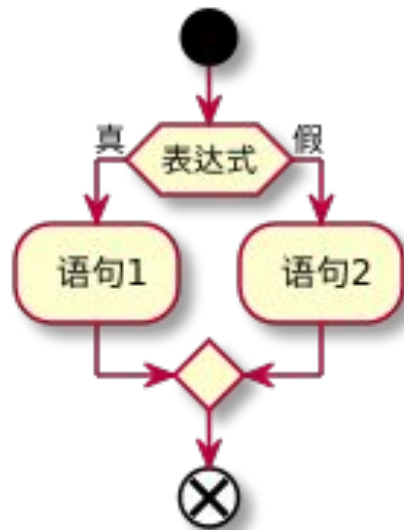


if-else语句

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

阶梯水费计算

```
if( x <= 15 )  
    y = 4 * x / 3;  
else  
    y = 2.5 * x - 10.5;
```



if语句

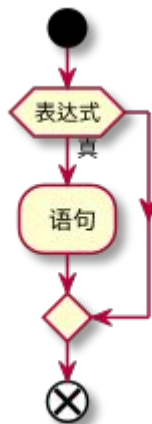


if (表达式)
 语句

/* 统计不及格人数 */

if(score<60)

 failed ++;





复合语句

复合语句由**多条语句**构成，用**{ }**括起来。**(for循环的循环语句已经使用过了)**

```
for (fahr = lower; fahr <= upper; fahr++)  
{  
    celsius = (5.0 / 9.0) * (fahr - 32);  
    printf("%d %6.1f\n", fahr, celsius);  
}
```

{ }里面有**2条**语句，还可以**更多**，理论上**无限制**



在if/if-else中使用复合语句

```
if( choice==apple )
{
    printf("your choice is apple\n");
    printf("the price for apple is 3\n");
}
else
{
    printf("your choice is not apple\n");
    printf("the price is 2\n");
}
```



在if/if-else中嵌套if/if-else语句

if (表达式)
 语句

语句可以是if-else语句或者if
语句

```
if( choice==apple )  
{  
    if( user==student )  
        print("price is 2");  
    else  
        print("price is 3");  
}
```



在if/if-else中嵌套if/if-else语句

```
if( choice!=apple )  
    print("price is 1");  
else  
{  
    if( user==student )  
        print("price is 2");  
    else  
        print("price is 3");  
}
```



在if/if-else中嵌套if/if-else语句

```
if( choice!=apple )
    print("price is 1");
else
{
    if( user==student )
        print("price is 2");
    else
        print("price is 3");
}
```

省略[], 等价于

```
if( choice!=apple )
    print("price is 1");
else if( user==student )
    print("price is 2");
else
    print("price is 3");
```

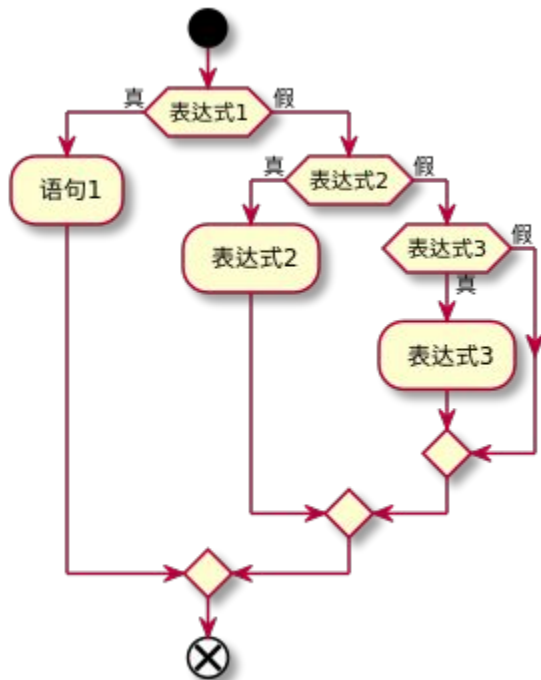
if-else if多路分支语句



```
if (表达式1)
    语句1
else if(表达式2)
    语句2
.....
else if(表达式n-1)
    语句n-1
else
    语句n
```

```
if( choice!=apple )
    print("price is 1");
else if( user==student )
    print("price is 2");
else
    print("price is 3");
```

if/else 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n-1)
    语句n-1
else
    语句n
```

《黑镜：潘达斯奈基》

嵌套的 if – else 语句



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



省略 else 部分

- if-else语句可以省略else部分

if(表达式)

语句1

else

语句2

- 如果内嵌的if语句省略了else部分，容易产生歧义，从而导致逻辑错误

省略 else 部分的嵌套if-else语句



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

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

- 体现了else与if的匹配原则
: else找前面最近的未被匹配的if, 与之配对
- 使用{ }可以改变层次, 改变配对和逻辑关系

```
if(表达式1)
{
    if(表达式2) 语句1
}
else
    if(表达式3) 语句3
    else 语句4
```



if/else语句应用 (I)

判断整数number的奇偶性

```
if( number%2==0 )  
    printf("The number is even.\n");  
else  
    printf("The number is odd.\n");
```



if/else语句应用 (II)

计算并输出浮点数 x 的绝对值

```
if(  $x \geq 0$  )  
    printf("abs(%f) = %f\n", x, x);  
else  
    printf("abs(%f) = %f\n", x, -x);
```

if/else语句应用 (III)



计算分段函数：

```
double x, y;
```

```
if( x < 0 )
```

```
    y = 0;
```

```
else if ( x <= 15 )
```

```
    y = 4 * x / 3;
```

```
else
```

```
    y = 2.5 * x - 10.5;
```

$$f(x) = \begin{cases} 0 & x < 0 \\ \frac{4}{3}x & 0 \leq x \leq 15 \\ 2.5x - 10.5 & x > 15 \end{cases}$$



作业

根据输入的月份输出该月的天数，假设该年份不是闰年
怎么用最少的分支结构完成



2) 数据类型

- 整型 int
- 浮点型 float
- 双精度 double
- 字符型 char
 - 小写字母: 'a' 'b' 'c' ... 'z'
 - 大写字母: 'A' 'B' 'C' ... 'Z'
 - 数字: '0' '1' '2' ... '9'
 - 括号、标点符号、运算符
 - 特殊字符(空格、换行)
 - ASCII码表

字符的输入与输出



```
char ch;
```

```
scanf("%c", &ch); /* 读入一个字符 */
```

```
printf("%c", ch); /* 输出字符ch */
```

```
ch = getchar(); /* 读入一个字符 */
```

```
putchar( ch ); /* 输出字符ch */
```

字符的ASCII码



- 每个字符都有一个次序值
称为它的ASCII码
- 字符类型占1个字节(Byte)
- 字符类型可以认为是“很短”整数, 值等于其它的ASCII码值。
 - 字符可以比较大小(根据ASCII码值)

字符的ASCII码(续)



- 字母的ASCII码值具有连续的、升序
 - `abcd...z` 的ASCII码为 `97,98,99,...,122`
 - `ABCD...Z` 的ASCII码为 `65,66,67,...,90`
- 数字也具有连续的、升序的ASCII码值
 - `012...9` 的ASCII码为 `48,49,50,...57`



3) 关系运算

- 关系运算有：

> 例如 $a > b$

< 例如 $a < b$

>= 例如 $a \geq b$

<= 例如 $a \leq b$

== 例如 $a == b$

!= 例如 $a != b$

= 和 ==

- 结果为逻辑值：真或假

- 真的值为1，假的值为0



4) 逻辑运算

&& 逻辑与: $a \&\& b$ 为真 $\Leftrightarrow a$ 和 b 都为真

$a \&\& b$ 为假 $\Leftrightarrow a$ 和 b 不全为真

(至少一个为假)

|| 逻辑或: $a || b$ 为真 $\Leftrightarrow a$ 和 b 不全为假

(至少一个为真)

! 逻辑非: $!a$ 为真 $\Leftrightarrow a$ 为假

$!a$ 为假 $\Leftrightarrow a$ 为真



a与b的逻辑运算

a	b	a&&b	a b	!a
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

逻辑运算运用



- 判断字符 `ch` 是否为数字字符

```
ch>='0' && ch<='9'
```

```
if( ch>='0' && ch <='9' )
```

```
    printf("It is a digital\n");
```

```
else
```

```
    printf("It is NOT a digital\n");
```



逻辑运算运用(续)

- 判断字符 `ch` 是否为小写字母

`ch>='a' && ch<='z'`

- 判断字符 `ch` 是否为大写字母

`ch>='A' && ch<='Z'`

- 判断字符 `ch` 是否为字母

`(ch>='a' && ch<='z') || (ch>='A' && ch<='Z')`



逻辑运算运用(续)

- 闰年判断
 - 被4整除, 但是不能被100整除
 - 或者被400整除

```
int year;
```

```
scanf( "%d", &year);
```

```
if( (year%4==0 && year%100!=0) || year%400==0 )
```

```
    printf("%d is a leap year\n", year);
```

```
else
```

```
    printf("%d is NOT a leap year\n", year);
```



逻辑运算运用(续)

[例3-7] 输入10个字符，统计其中的英文字母、数字字符和其他字符的个数。

```
int letter, digit, other, i;
```

```
char ch;
```

```
letter = 0;
```

```
digit = 0;
```

```
other = 0;
```

作业 - 统计字符





5) switch语句

switch(表达式)

{

case 常量表达式1 : 语句段1;

case 常量表达式2 : 语句段2;

case 常量表达式k : 语句段k;

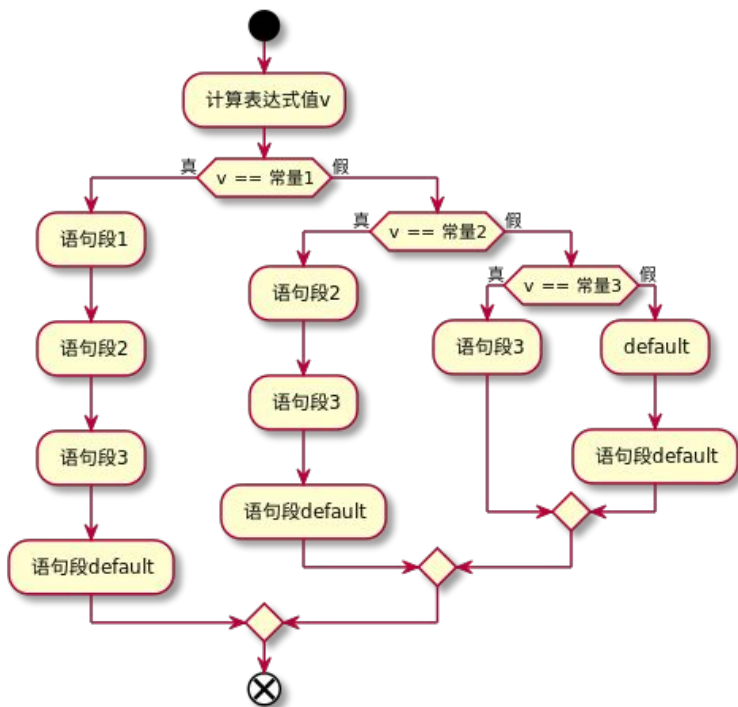
case 常量表达式n : 语句段n;

default : 语句段n+1;

}

- 3个关键字
 - switch
 - case
 - Default
- n+1个表达式
 - n个常量表达式
 - 常量组成
- n+1个语句段
 - 每语句段可能有0条或多条语句

switch语句流程



```
switch (表达式v) {  
    case 常量1:  
        语句段1;  
    case 常量2:  
        语句段2;  
    case 常量3:  
        语句段3;  
    default:  
        default语句段;  
}
```

- 从哪个语句段开始执行？
 - 有 $n+1$ 个入口可以选择
 - 分别对应 n 个case的入口和1个缺省(default)的入口
- 根据表达式的值决定执行入口
 - 如果表达式等于常量表达式 k , 那么从语句段 k 开始执行



break语句

- 在switch语句中使用break语句
 - 从break所在地跳转出switch语句



switch语句应用

例3-5 输入一个形式如“**操作数****运算符****操作数**”的算式，输出运算结果。

例如

输入：3.1+4.8

输出：7.9

```
double value1, value2;
```

```
char op;
```

```
printf("Type in an expression: ");
```

```
scanf("%lf%c%lf", &value1, &op, &value2);
```



Type in an expression: 3.1+4.8 =7.9

```
switch( op )
{
    case '+':
        printf("=%.2f\n", value1 + value2);
        break;
    case '-':
        printf("=%.2f\n", value1 - value2);
        break;
    case '*':
        printf("=%.2f\n", value1 * value2);
        break;
    case '/':
        printf("=%.2f\n", value1 / value2);
        break;
    default:
        printf("Unknown operator\n");
        break;
}
```




switch语句应用

- 输入10个字符，分别统计出其中空格或回车、数字和其他字符的个数。

```
int blank, digit, other, i;
```

```
char ch;
```

```
blank = digit = other = 0;
```

例3-10



```
printf("Enter 10 characters: ");
for(i = 1; i <= 10; i++)
{
    ch = getchar();
    switch( ch )
    {
        case ' ' :
        case '\n': blank ++;
                    break;

        case '0' : case '1' : case '2' : case '3' : case '4' :
        case '5' : case '6' : case '7' : case '8' : case '9' :
                    digit ++;
                    break;

        default : other ++;
                    break;
    }
}
printf("blank=%d, digit=%d, other=%d\n", blank, digit, other);
```



复习与思考要点

- if与else的配对原则，及改变配对的方法
- break语句在switch中的作用
- 字符数据的输入输出函数
- 字符数据的判别式(数字，大小写字母)
- 字符常量的书写方法
- 逻辑运算的规律，以及在if-else语句中的使用