

# 第七章 循环结构程序设计

1. 循环结构概述
2. 用DO循环语句实现循环
3. 用DO WHILE语句实现循环
4. 用IF和GOTO语句实现循环
5. 强制性终止循环过程
6. 特殊DO循环语句与EXIT语句
7. 循环嵌套
8. 循环结构程序设计应用举例
9. 习题七



# 7.1 循环结构概述



- ◆ 重复问题
- ◆ 循环结构
- ◆ 循环分类

在求解复杂问题时,常常遇到大量重复或相似的处理过程(计算过程、操作过程)。这类问题就属于重复问题。  
如:从键盘输入5个数,求5个数之和,并输出。

求解算法:

- (1)初始化:赋初值0.0至sum。
- (2)输入一个数x,并累加到sum中。
- (3)输入一个数x,并累加到sum中。
- (4)输入一个数x,并累加到sum中。
- (5)输入一个数x,并累加到sum中。
- (6)输入一个数x,并累加到sum中。
- (7)输出结果sum。
- (8)结束。

求解程序:

```
sum=0.0
READ *, x;
sum=sum+x
READ *, x;
sum=sum+x
READ *, x;
sum=sum+x
READ *, x;
sum=sum+x
PRINT *, 'sum=',sum
END
```

这是一个典型的重复问题。只是问题规模较小,可用顺序结构实现,但问题规模较大时(如100个数)就不适用了,需采用循环结构实现。



## 7.1 循环结构概述



- ◆ 重复问题
- ◆ 循环结构
- ◆ 循环分类

对于问题规模较大的重复问题适宜采用循环结构。

如：从键盘输入50个数，求50个数之和，并输出。

求解算法：

求解程序：

(1)初始化：0.0 $\Rightarrow$ sum。

(2)1 $\Rightarrow$ i。

(3)如果i>50,则执行步骤(7)。

(4)输入一个数x,并累加到sum中。

(5)i+1 $\Rightarrow$ i。

(6)转步骤(3)。

(7)输出结果sum。

(8)结束。

```
sum=0.0
```

```
DO I=1,50
```

```
  READ *, x
```

```
  sum=sum+x
```

```
ENDDO
```

```
PRINT *, 'sum=', sum
```

```
END
```

对于上述算法和程序，不管程序规模多大，长度不变。

(示例1) (示例2)



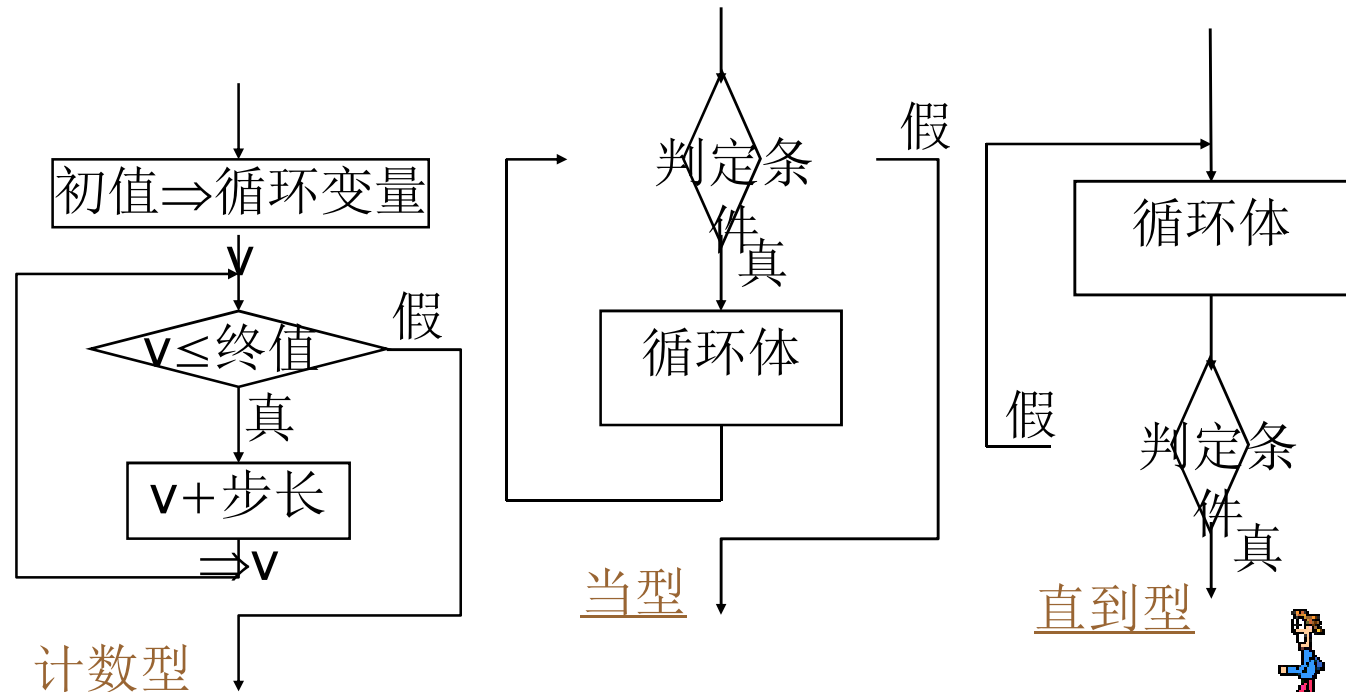
# 7.1 循环结构概述

.2222

- ◆ 重复问题
- ◆ 循环结构
- ◆ 循环分类

循环结构分为两类：

- 循环体的循环次数事先确定(如[例7.1]中循环结构), 这类循环结构称为确定性循环, 或称“计数型”循环。
- 循环体的循环次数事先不确定(如[例7.2]中循环结构), 这类循环结构称为非确定性循环, 或称“条件型”循环。



## 7.1 循环结构概述



- ◆ 重复问题
- ◆ 循环结构
- ◆ 循环分类

[例7.1] 求 $S=1+2+3+\dots+N$ 的值( $N$ 由键盘输入, $N>2$ )。  
编写程序。

解: 求解这一问题,将执行 $N-1$ 次加法运算。

如果将等式写成 $S=0+1+2+\dots+N$ ,则执行 $N$ 次加法运算。

重复问题易采用循环结构实现。

用 $S$ 保存和,用 $N$ 保存最大求和值,用 $I$ 作计数器。编写程序如下:

```
INTEGER S,N,I
READ(*,*)N
S=0
DO I=1,N
  S=S+I
EDNDO
WRITE(*, *) 'S=',S
END
```



## 7.1 循环结构概述



- ◆ 重复问题
- ◆ 循环结构
- ◆ 循环分类

[例7.2] 读入某班级考试成绩(人数不定),计算其平均成绩,并输出。

解: 由于学生人数不定,所以输入的学生成绩个数不确定,以输入-1作为结束标志(因为成绩不可能为负数),从而动态确定学生人数n。

平均成绩计算公式是:  $av = (x_1 + x_2 + \dots + x_n) / n$

```
INTEGER sum,n,x
```

```
REAL av
```

```
sum=0;n=0
```

```
READ *,x
```

```
DO WHILE (x<>-1)
```

```
  n=n+1; sum=sum+x; READ *,x
```

```
ENDDO
```

```
av=sum/n
```

```
WRITE(*, *) '平均成绩为: ',av
```

```
END
```



## 7.1 循环结构概述



- ◆ 重复问题
- ◆ 循环结构
- ◆ 循环分类

[例7.1] 求 $S=1+2+3+\dots+N$ 的值( $N$ 由键盘输入, $N>2$ )。  
编写程序。

解：求解这一问题,将执行 $N-1$ 次加法运算。

如果将等式写成 $S=0+1+2+\dots+N$ ,则执行 $N$ 次加法运算。

重复问题易采用循环结构实现。

用 $S$ 保存和,用 $N$ 保存最大求和值,用 $I$ 作计数器。编写程序如下：

```
INTEGER S,N,I
READ(*,*)N
S=0
DO I=1,N
  S=S+I
EDNDO
WRITE(*,*) 'S=',S
END
```



## 7.1 循环结构概述



- ◆ 重复问题
- ◆ 循环结构
- ◆ 循环分类



[例7.2] 读入某班级考试成绩(人数不定),计算其平均成绩,并输出。

解: 由于学生人数不定,所以输入的学生成绩个数不确定,以输入-1作为结束标志(因为成绩不可能为负数),从而动态确定学生人数n。

平均成绩计算公式是:  $av = (x_1 + x_2 + \dots + x_n) / n$

```
INTEGER sum,n,x
```

```
REAL av
```

```
sum=0;n=0
```

```
READ *,x
```

```
DO WHILE (x<>-1)
```

```
  n=n+1; sum=sum+x; READ *,x
```

```
ENDDO
```

```
av=sum/n
```

```
WRITE(*,*) '平均成绩为:',av
```

```
END
```





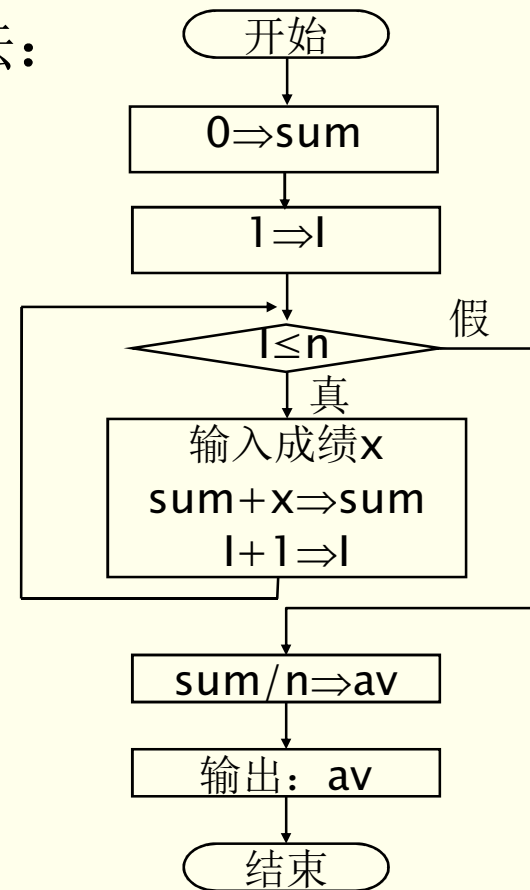
## 计数型循环示例

问题：读入某班级考试成绩(人数30),计算其平均成绩,并输出。编写程序。

程序：

```
INTEGER :: sum,n=30,i,x
REAL av
sum=0;
DO i=1,n
  READ *,x
  sum=sum+x
ENDDO
av=sum/n
WRITE(*,*) '平均成绩为: ',av
END
```

算法：



## 当型循环示例

问题：读入某班级考试成绩(人数不定),计算其平均成绩,并输出。编写程序。

程序：

INTEGER sum,n,x

REAL av

sum=0;n=0

READ \*,x

DO WHILE (x<>-1)

n=n+1; sum=sum+x

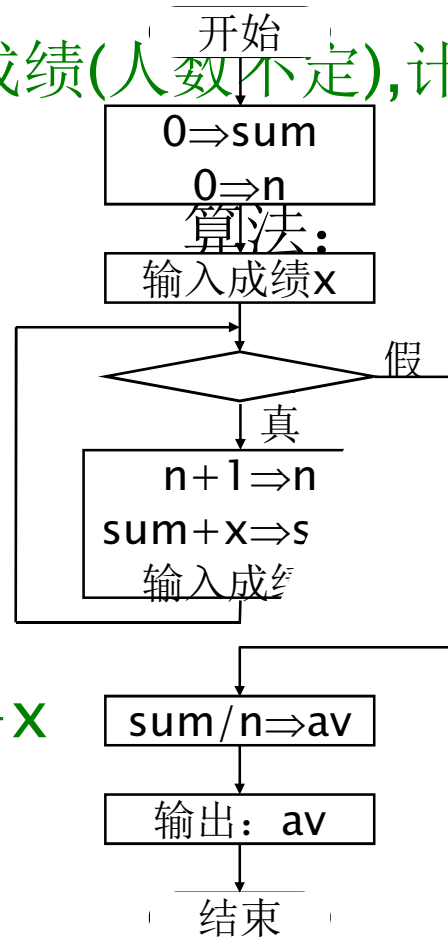
READ \*,x

ENDDO

av=sum/n

WRITE(\*,\*) '平均成绩为:',av

END



## 直到型循环示例

问题：读入某班级考试成绩(人数不定),计算其平均成绩,并输出。编写程序。

程序：

INTEGER sum,n,x

REAL av

sum=0;n=0

READ \*,x

DO

n=n+1; sum=sum+x; READ \*,x

IF (x=-1) EXIT

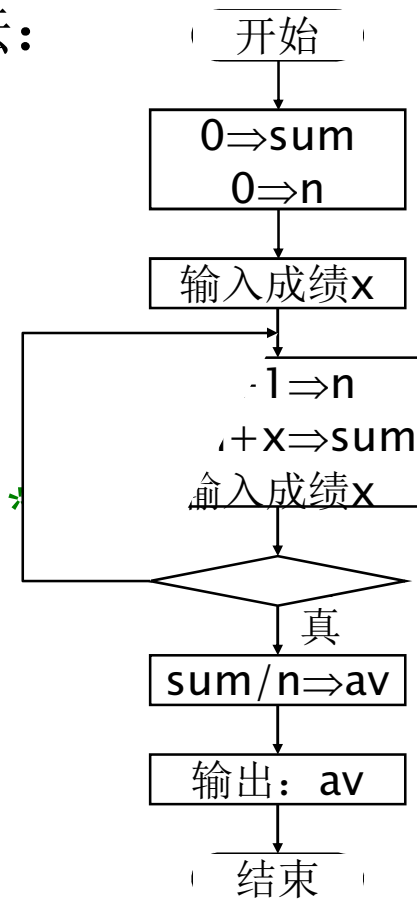
ENDDO

av=sum/n

WRITE(\*,\*) '平均成绩为: ',av

END

算法：



## 7.2 使用DO循环语句实现计数型循环



◆ 一般格式

◆ 执行过程

◆ 示例1

◆ 示例2

<DO循环语句> → [<名字>:] DO

<v> = <e1>, <e2> [, <e3>]

<语句1>

<语句2>

.....

<语句n>

ENDDO [<名字>]

说明:

① **v**为循环变量，一般为整型变量，用来控制循环次数。**e1**、**e2**和**e3**为表达式，其类型与循环变量类型相同。**e1**代表循环变量**v**的初值，**e2**代表循环变量**v**的终值，**e3**代表循环变量**v**的步长，**e3**可缺省，**e3**缺省指**e3**为1。

② 循环次数计算公式是： $\text{MAX}(\lfloor (\text{e2} - \text{e1}) / \text{e3} + 1 \rfloor, 0)$ 。

③ 如果**e3** > 0，则一般要求**e1** ≤ **e2**，否则循环次数为0，即循环体一次也不执行。

④ 如果**e3** < 0，则一般要求**e1** ≥ **e2**，否则循环次数为0，即循环体一次也不执行。

⑤ 在循环体中可引用循环变量**v**的值，但不能赋值，即修改它的值。

(示例)



## 7.2 使用DO循环语句实现计数型循环



◆ 一般格式

◆ 执行过程

◆ 示例1

◆ 示例2

DO循环语句执行过程说明：下面DO循环语句的  
循环次数为5：

M=1;N=5;L=1

DO I=M,N+5,L+1 !输出1、3、5、7、9

PRINT\*,I,M,N,L

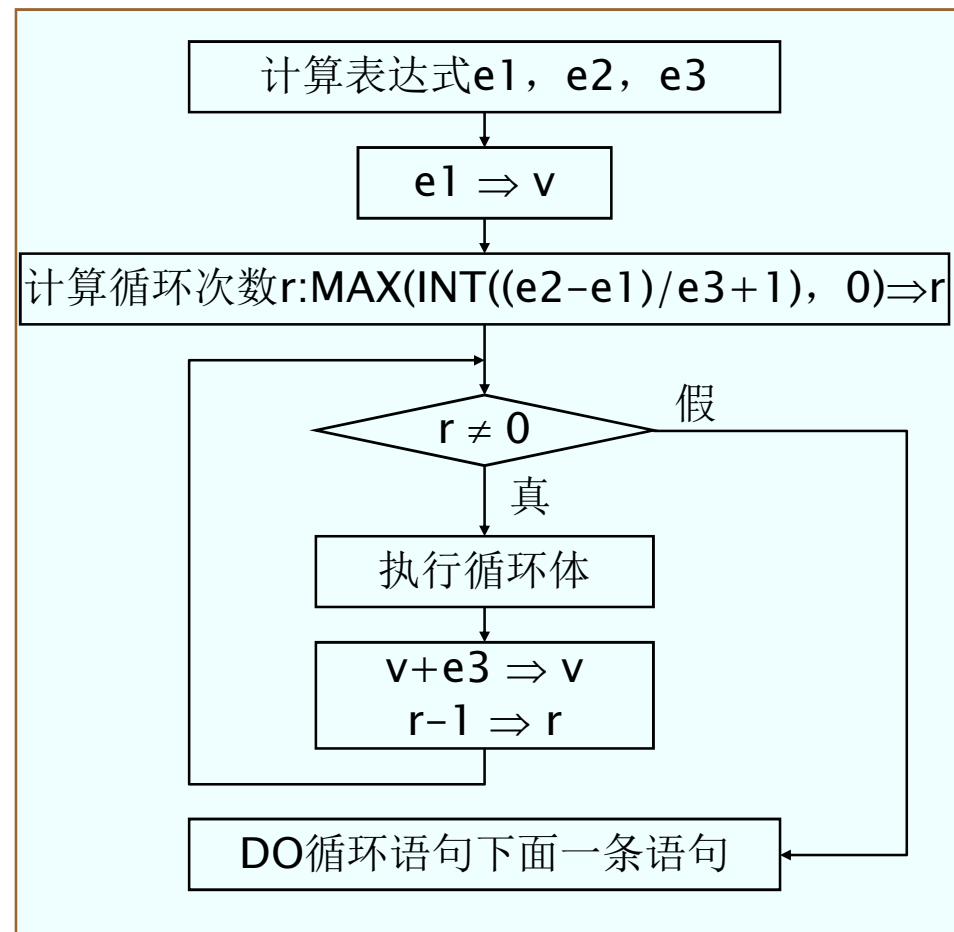
M=M+1;N=N\*I;L=2\*L+1

ENDDO

程序运行后,输出结果为:

1	1	5	1
3	2	5	3
5	3	15	9
7	4	75	23
9	5	525	53





## 7.2 使用DO循环语句实现计数型循环



- ◆ 一般格式
- ◆ 执行过程
- ◆ 示例1
- ◆ 示例2

**[例7.3] 打印5个随机数。**

解：这是一个重复问题,可用循环5次的DO  
循环语句实现。

```
REAL R
DO I=1,5
  CALL RANDOM_NUMBER(R)
  PRINT "(1X,F8.3\)",R
END DO
END
```

程序运行后,输出结果为:

```
0.0000004 0.0 254804 0.3525162
0.6669145 0.9630556
```



## 7.2 使用DO循环语句实现计数型循环



- ◆ 一般格式
- ◆ 执行过程
- ◆ 示例1
- ◆ 示例2

[例7.4] 计算4!,并输出。

解:  $4! = 1 \times 2 \times 3 \times 4$ 。对于n!,计算公式是:

$$n! = 1 \times 2 \times 3 \times \dots \times (n-1) \times n。$$

该问题是一个重复计算问题,需做n-1次乘法。用DO循环语句实现。

→  
PARAMETER(n=4)  
INTEGER : : factor=1  
DO i=2,n  
    factor=factor\*i  
PRINT\*,i,'! =',factor  
END DO  
END

程序运行后。输出结果为:

2! =	2
3! =	6
4! =	24



← Back



## 7.2 使用DO循环语句实现计数型循环



◆ 一般格式

◆ 执行过程

◆ 示例1

◆ 示例2

M=1;N=10;L=1

DO I=1,10,2 ! 输出1、3、5、7、9

PRINT \*,I

ENDDO

DO I=10,1,-2 ! 输出10、8、6、4、2

PRINT\*,I

ENDDO

DO I=M+1,N+2,L+3 ! 输出2、6、10

PRINT \*,I

ENDDO

DO I=1,5 ! 输出1、2、3、4、5

PRINT \*,I

ENDDO



## 7.3 使用DO WHILE循环语句实现循环



- ◆ 一般格式
- ◆ 执行过程
- ◆ 示例

<DO循环语句> → [<名字>:] DO WHILE (<循环条件>)

```
    <语句1>  
    <语句2>  
    .....  
    <语句n>  
ENDDO [<名字>]
```

} 循环体

说明:

- ① 循环条件可以是关系表达式，也可以是逻辑表达式，但不能是算术表达式或字符表达式。
- ② 使用DO WHILE语句时要特别注意死循环的产生，要保证循环体中至少有一条对循环控制条件有影响的语句，否则将产生死循环。  
(示例)

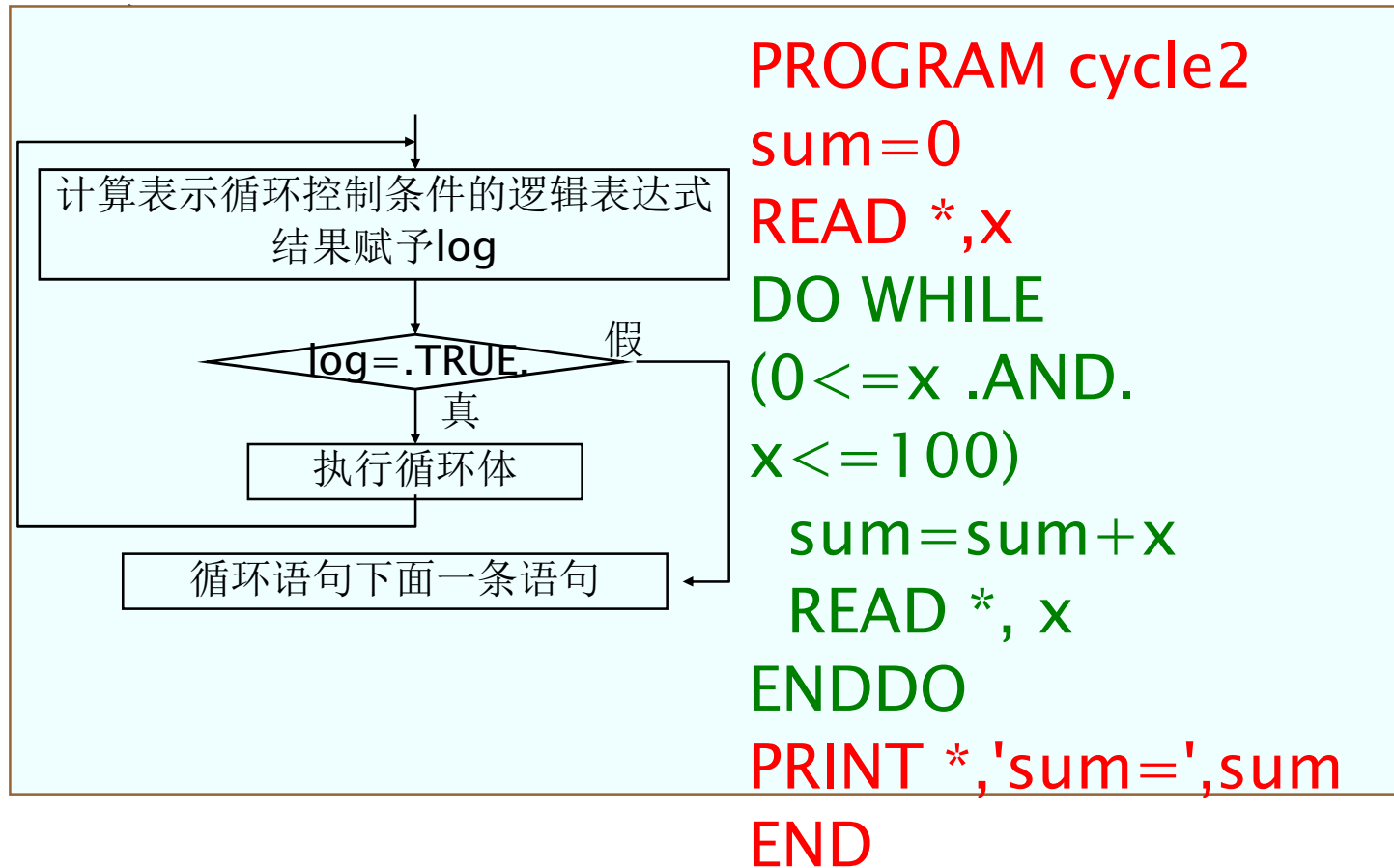


## 7.3 使用DO WHILE循环语句实现循环



- ◆ 一般格式
- ◆ 执行过程
- ◆ 示例

DO WHILE循环语句在每次循环体执行前都要计算表示循环控制条件的逻辑表达式,其计算结果决定循环体是否继续执行,循环体的执行过程必须对循环控制条件产生



## 7.3 使用DO WHILE循环语句实现循环



- ◆ 一般格式
- ◆ 执行过程
- ◆ 示例

[例7.5] 求两整数M和N的最大公约数和最小公倍数。

解:

→ 用 $(R_i, R_{i+1})$ 表示 $R_i$ 和 $R_{i+1}$ 的最大公约数,  $i=1, 2, 3, \dots, n$ , 则有:

$(R_1, R_2) = (R_2, R_3) = \dots = (R_i, R_{i+1}) = \dots = (R_n, 0)$

其中:  $R_1 = M, R_2 = N, R_j$  是  $R_{j-2}$  除以  $R_{j-1}$  的余数,  $j=3, 4, \dots, n+1, R_{n+1} = 0$ 。

如:  $(12, 8) = (8, 4) = (4, 0)$ , 最大公约数为4。  
这是一个重复计算问题, 重复次数不定, 可用DO WHILE循环语句实现。

最小公倍数通过最大公约数计算得到, 用 $\langle M, N \rangle$ 表示最小公倍数, 有:

$$\langle M, N \rangle = M \times N / (M, N)$$

程序:



← Back

## 7.3 使用DO WHILE循环语句实现循环

- ◆ 一般格式
- ◆ 执行过程
- ◆ 示例

```
PROGRAM cycle1
sum=0
READ *,x
DO WHILE (x<>-1)
    sum=sum+x
    READ *,x
ENDDO
PRINT *, 'sum=',sum
END
```

```
PROGRAM cycle2
sum=0
READ *,x
DO WHILE (0<=x .AND.
x<=100)
    sum=sum+x
    READ *,x
ENDDO
PRINT *, 'sum=',sum
END
```

说明:

- ① 循环条件可以是关系表达式，也可以是逻辑表达式，但不能是算术表达式或字符表达式。
- ② 使用DO WHILE语句时要特别注意死循环的产生，要保证循环体中至少有一条对循环控制条件有影响的语句，否则将产生死循环。

(示例)



## 7.3 使用DO WHILE循环语句实现循环

- ◆ 一般格式
- ◆ 执行过程
- ◆ 示例

```
PROGRAM gcd
INTEGER M,N,R1,R2,R3,gcd,lcm
PRINT *, '输入两个自然数: '
READ *,M,N
R1=M;R2=N
DO WHILE(R2<>0)
    R3=MOD(R1,R2) !求R1除以R2的余数
    R1=R2;R2=R3
END DO
gcd=R1;lcm=M*N/gcd
PRINT "(1X,I3,'和',I3,'的最大公约数为:",I3)",M,N,gcd
PRINT "(1X,I3,'和',I3,'的最小公倍数为:",I3)",M,N,lcm
END
```



输入数据：9,12✓

输出[例7.5] 求两整数M和N的最大公约数和最小公倍数。

解：

用 $(R_i, R_{i+1})$ 表示 $R_i$ 和 $R_{i+1}$ 的最大公约数,  $i=1, 2, 3, \dots, n$ , 则有：

$$(R_1, R_2) = (R_2, R_3) = \dots = (R_i, R_{i+1}) = \dots = (R_n, 0)$$

其中： $R_1=M, R_2=N, R_j$ 是 $R_{j-2}$ 除以 $R_{j-1}$ 的余数,  $j=3, 4, \dots, n+1, R_{n+1}=0$ 。

如： $(12, 8) = (8, 4) = (4, 0)$ , 最大公约数为4。

这是一个重复计算问题, 重复次数不定, 可用DO WHILE循环语句实现。

最小公倍数通过最大公约数计算得到, 用 $\langle M, N \rangle$ 表示最小公倍数, 有：

$$\langle M, N \rangle = M \times N / (M, N)$$

程序：

结果为：

9和 12 的最大公约数为： 3

9和 12 的最小公倍数为： 36

## 7.4 使用IF和GOTO语句实现循环



用IF语句实现选择结构,用GOTO语句实现无条件控制转移,两者结合可实现循环结构,特别是“直到型”循环结构。IF语句给出循环条件,GOTO语句控制循环转移。由于GOTO语句易造成程序错误,故现代程序设计方法不提倡使用此法实现循环。

```
INTEGER M,N,R1,R2,R3,gcd,lcm
PRINT *,'输入两个自然数: '
READ *,M,N
R1=M;R2=N
100 R3=MOD(R1,R2) !求R1除以R2的余数
    R1=R2;R2=R3
    IF (R2<>0) GOTO 100
! 或 IF (R2=0)THEN
!   ELSE
!   GOTO 100
!   ENDIF
gcd=R1;lcm=M*N/gcd
PRINT "(1X,I3,'和',I3,'的最大公约数为: ',I3)",M,N,gcd
PRINT "(1X,I3,'和',I3,'的最小公倍数为: ',I3)",M,N,lcm
```



← Back



## 7.5 强制性终止循环(EXIT、CYCLE)



◆ 概述

◆ EXIT

◆

CYCLE

◆ 示例

一般情况下,循环过程都是正常结束。有些特殊问题,在循环处理过程中遇到特殊情况需要提前终止本次循环或整个循环,以免循环陷入死循环,造成严重后果。

终止循环过程有两条语句: **EXIT**、**CYCLE**。



## 7.5 强制性终止循环(EXIT、CYCLE)



◆ 概述

◆ EXIT

◆ CYCLE

◆ 示例

**功能：**在循环体执行过程中强制性终止整个循环语句的执行,转循环语句后的第一条语句执行。只能在DO和DO WHILE循环语句内使用EXIT语句。

!示例程序一：输出奇数1,3,5,7,9

```
DO I=1, 100, 2  
  IF (I>=9) EXIT  
ENDDO  
END
```

!示例程序二：输出奇数1,3,5,7,9

```
I=1  
DO WHILE (I<=100)  
  PRINT *, I  
  IF (I>=9) EXIT  
  I=I+2  
ENDDO  
END
```

!示例程序三：输出奇数1,3,5,7,9

```
I=1  
DO  
  PRINT *, I  
  IF (I>=9) EXIT  
  I=I+2  
ENDDO  
END
```



## 7.5 强制性终止循环(EXIT、CYCLE)



◆ 概述

◆ EXIT



CYCLE

◆ 示例

**功能：**在循环体执行过程中强制性终止本次循环的执行,转循环开始语句(DO语句)执行。只

能  
语

```
DO I=2,5  
  PRINT "(1X, I3\\)",I  
  IF (I>3) CYCLE  
  PRINT "(1X,I3\\)",I  
END DO  
PRINT *,‘结束！’  
END
```

! 执行结果: 2 2 3 3 4 5

CYCLE



## 7.5 强制性终止循环(EXIT、CYCLE)



◆ 概述

◆ EXIT



CYCLE

◆ 示例



[例7.6] 一个猜数游戏。这个程序随机产生1~10之间的一个整数,用户猜测这个数,并输入到程序中,程序判定并输出猜测结果。

解: 算法描述如下:

(1)产生随机数fun\_num;

(2)提示用户输入猜测到的数my\_guess,并输入该数;

(3)如果 $my\_guess < fun\_num$ ,则猜测数太小,建议猜大一点,继续猜;

(4)如果 $my\_guess > fun\_num$ ,则猜测数太大,建议猜小一点;继续猜;

(5)如果 $my\_guess = fun\_num$ ,则猜对了,打印祝贺语;

(6)结束。

这是一个重复处理问题,重复次数不确定,只有猜对后才能终止循环。使用特殊DO循环语句,并使用EXIT语句在猜对后强行终止循环的执行。

可用语句CALL BANDOM\_NUMBE(R)调用标准子程序

BANDOM\_NUMBE(R)在0~1的实数范围内产生随机数R。 $10 * R$ 的范围是[0,10), $10 * R + 1$ 的范围[1,11),也就是1.000000 and 10.999999之间。用INT(R)将得到1~10范围的整数值。

程序:



← Back

## 7.5 强制性终止循环(EXIT、CYCLE)



[例7.6] 一个猜数游戏。这个程序随机产生1~10之间的一个整数,用户猜测这个数,并输入到程序中,程序判定并输出猜测结果。

◆ 概述

◆ EXIT



CYCLE

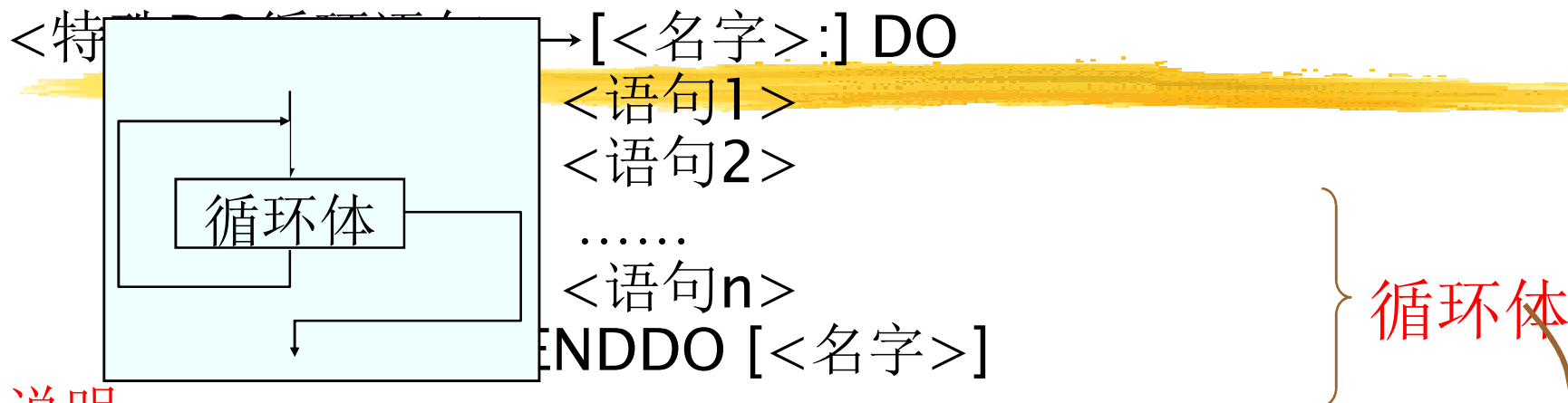
◆ 示例



```
INTEGER fun_num, my_guess
REAL R
CALL RANDOM_NUMBER(R)
fun_num=MOD(INT(10E9*R+1),10)
DO
  PRINT *, '输入你猜测的整数(1-10): '
  READ*, my_guess
  IF (my_guess>fun_num) THEN
    PRINT *, '猜测数太大, 小一点, 再试一次。'
  ELSEIF (my_guess<fun_num) THEN
    PRINT *, '猜测数太小, 大一点, 再试一次。'
  ELSE
    PRINT *, '猜中了! 祝贺你!'
    EXIT
  ENDIF
END DO
END
```



## 7.6 特殊DO循环语句与EXIT语句



说明:

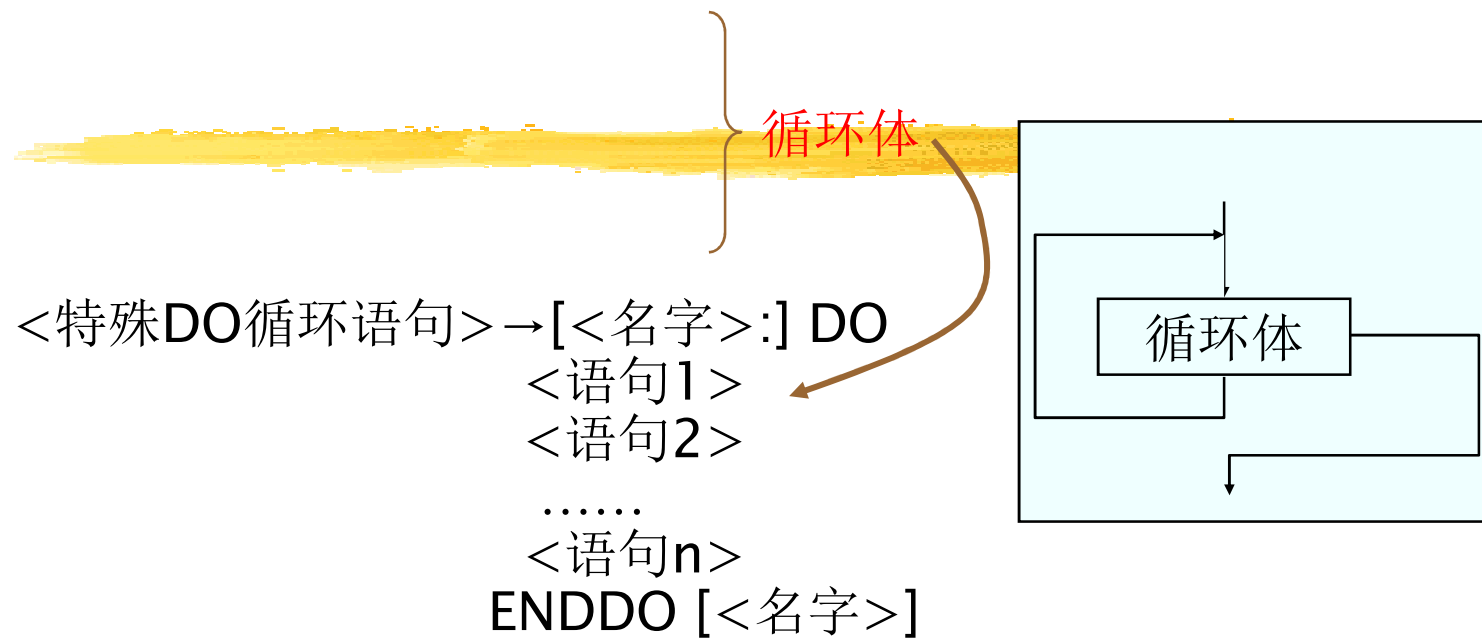
- ① 特殊DO循环语句必须与EXIT语句配合使用。
- ② 使用特殊DO循环语句可使循环变得短小、简洁、清晰。

执行过程:

- (1) 按顺序依次执行循环体语句。
- (2) 循环执行过程中,若满足循环结束条件,则执行EXIT语句退出循环。
- (3) 无条件转(1)。

(示例)

## 7.6 特殊DO循环语句与EXIT语句



说明:

- ① 特殊DO循环语句必须与EXIT语句配合使用。
- ② 使用特殊DO循环语句可使循环变得短小、简洁、清晰。

执行过程:

- (1) 按顺序依次执行循环体语句。
- (2) 循环执行过程中,若满足循环结束条件,则执行EXIT语句退出循环。
- (3) 无条件转(1)。

(示例)

```
INTEGER M, N, R1, R2, R3, gcd, lcm
PRINT *, '输入两个自然数: '
READ *, M, N
R1 = M; R2 = N
DO
  IF (R2 == 0) EXIT
  R3 = MOD(R1, R2) !求R1除以R2的余数
  R1 = R2; R2 = R3
END DO
gcd = R1; lcm = M * N / gcd
PRINT "(1X,I3,'和',I3,'的最大公约数为: ',I3)", M, N, gcd
PRINT "(1X,I3,'和',I3,'的最小公倍数为: ',I3)", M, N, lcm
END
```



## 7.7 循环嵌套

◆ 概述

◆ 示例

1

◆ 示例

2

◆ 示例

3

◆ 示例

4



- 前面介绍的循环结构和循环语句都是一层循环,许多实际问题的求解算法和程序往往具有多重循环结构。
- 多重循环结构称为循环嵌套,循环嵌套是指在一个循环结构或循环语句的循环体中又含有循环结构或循环语句。对于n层循环嵌套称为n重循环。嵌套的循环层数原则上不限,但不易太多。
- 计数型、当型、直到型循环结构都可相互嵌套,DO语句、DO WHILE语句、特殊DO循环语句都可相互嵌套。根据实际问题的求解需要决定循环嵌套的形式和层数。
- 如果有n重循环,且循环次数分别为 $r_1$ 、 $r_2$ 、...、 $r_n$ ,则一般最内层循环体的循环总次数为:  $r_1 \times r_2 \times \dots \times r_n$ 。

一层循

环

二层循

环

三层循

环

四层循

环

三层循

环

二层循

环

## 7.7 循环嵌套



◆ 概述

◆ 示例

1

◆ 示例

2

◆ 示例

3

◆ 示例

4

[例7.7] 打印九九乘法表。

解：求解这一问题的算法结构具有“计数型”循环结构与“计数型”循环结构嵌套的二重循环结构。程序中用两个DO循环语句嵌套来实现二重循环结构。

1\*1=1

1\*2=2

.....

1\*9=9

2\*1=2

2\*2=4

.....

2\*9=18

.....

9\*9=81

```
INTEGER i, j, product
DO i=1,9
  DO j=1,9
    product=i*j
    PRINT *,i,'*',j,'=',product
  ENDDO
ENDDO
END
```

! 内循环体执行81次



## 7.7 循环嵌套



◆ 概述

◆ 示例

1

◆ 示例

2

◆ 示例

3

◆ 示例

4



[例7.8]求  $\sum_{n=1}^{10} n!$

解：求解这一问题的算法结构具有“计数型”循环结构与“计数型”循环结构嵌套的二重循环结构。程序中用两个DO循环语句嵌套来实现二重循环结构。

```
INTEGER :: i, j, fac, fac_sum=0
```

! 循环嵌套

```
DO i=1,10 ! 用外层循环控制求和次数
```

```
  fac=1
```

```
  DO j=1,i ! 用内层循环控制阶乘
```

```
    fac=fac*j
```

```
  ENDDO
```

```
  fac_sum=fac_sum+fac
```

```
ENDDO
```

```
PRINT *, '前10个自然数的阶乘和=', fac_sum
```

```
END
```

! 赋值语句fac=fac\*j被执行55次

! 赋值语句fac\_sum=fac\_sum+fac被执行10

次

## 7.7 循环嵌套



◆ 概述

◆ 示例

1

◆ 示例

2

◆ 示例

3

◆ 示例

4

[例7.9]给出2和3组成的所有可能三位数,并输出这些三位数。

解: 求解这一问题的输出结果为:

222

223

232

233

322

323

332

333

共组成三位数



```
INTEGER :: i, j, k
INTEGER :: n      ! 存放组成的三位数
INTEGER :: num=0  ! 存放三位数的个数
PRINT *, '由2和3组成的三位数有: '
! 循环嵌套
DO i=2,3          ! 用外层循环控制百位数字
  DO j=2,3        ! 用内层循环控制十位数字
    DO k=2,3      ! 用内层循环控制个位数字
      n=100*i+10*j+k ! 计算组成的三位数
      num=num+1      ! 累计三位数的个数
      PRINT *, n
    ENDDO
  ENDDO
ENDDO
PRINT *, '共组成三位数', num, ' 个'
END
```

## 7.7 循环嵌套



◆ 概述

◆ 示例

1

◆ 示例

2

◆ 示例

3

◆ 示例

4

[例7.10]输入某班学生四门课成绩,求这些成绩中的最高成绩,并输出。

解: 该问题求解算法和程序具有二重循环结构。由于学生人数不定,所以控制学生人数的外层循环用DO WHILE语句实现,输入成绩为负数结束循环。由于课程门数确定,所以控制门数的内层循环用DO语句实现。二重循环结构由“当型”循环结构和“计数型”循环结构嵌套构成。

```
max=0.0
READ *, x
DO WHILE (x>=0)
  DO i=1, 4
    IF (x>max) max=x
    READ*, x
  END DO
END DO
WRITE(*, *) '最高成绩为', max
END
```



← Back

## 7.8 循环结构程序设计举例



- ◆ 例1 →
- ◆ 例2
- ◆ 例3
- ◆ 例4
- ◆ 例5
- ◆ 例6
- ◆ 例7
- ◆ 例8

[例7.11] 用迭代法求 $\sqrt{a}$ 的二次方根,即求 $\sqrt{a}$ 。精度要求小数点后7位。

解: 求正整数 $x$ 的平方根,可通过牛顿迭代法,只用加、减和除就可实现。

求 $\sqrt{a}$ 的牛顿迭代法公式是:

设: 初值 $x_0=1$

$$x_{i+1} = (x_i + a/x_i) / 2$$

$$i=1, 2, 3, \dots, n, |x_{n+1} - x_n| \leq 10^{-7}, x_{n+1} \approx \sqrt{a}$$

根据上面迭代公式,求 $a$ 的平方根的算法如图7-9所示。

根据算法,编写程序。

输入数据:

输入计算平方根的数: 2 ✓

输出数据为:

2.000000 平方根为: 1.414214

## 7.8 循环结构程序设计举例



- ◆ 例1
- ◆ 例2
- ◆ 例3
- ◆ 例4
- ◆ 例5
- ◆ 例6
- ◆ 例7
- ◆ 例8

[例7.12]根据下面泰勒级数求x的正弦值。

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

解：我们已经熟悉FORTRAN 90提供的SIN(x)、COS(x)、EXP(x)等内部函数,我们可直接使用这些内部函数计算正弦、余弦、指数等值。那么这些函数是如何计算的呢？一般都是通过其泰勒级数计算。下面我们不用内部函数,自己编写程序计算正弦值,然后与内部函数计算值进行比较。

遇到这类问题,必须将泰勒级数用通项公式表示,然后得出迭代公式。

$$\sin x = \sum_{n=1}^{\infty} (-1)^{n+1} x^{2n-1} / (2n-1)!$$

其中:  $n=1,2,3,\dots$ 。

设:  $T_n = (-1)^{n+1} x^{2n-1} / (2n-1)! = T_{n-1} (-1) x^2 / ((2n-2)(2n-1))$ ,  
 $n=1,2,3,\dots$

$\sin x \approx T_1 + T_2 + \dots + T_n$ ,  $|T_{n+1}| \leq 10^{-7}$ ,  $10^{-7}$ 决定了计算精度。

上述计算问题变成了一个求和问题,可用一个DO WHILE语句实现。

根据分析,设计算法,编写程序。

## 7.8 循环结构程序设计举例



- ◆ 例1
- ◆ 例2
- ◆ 例3
- ◆ 例4
- ◆ 例5
- ◆ 例6
- ◆ 例7
- ◆ 例8

[例7.13]向某银行贷款1000元,年利率为: 10%、11%、...、20%,计算不同年利率15年,20年和25年的月偿还金,并输出。

解: 设贷款本金为a,年利率为r,贷款年数为n,月偿还金为p。

$$p = \frac{a \cdot \frac{r}{12} \cdot (1 + \frac{r}{12})^{12n}}{(1 + \frac{r}{12})^{12n} - 1}$$

该问题用二重循环实现,外层循环控制不同利率,内层循环控制年限。

根据分析,设计算法,编写程序。

输入数据: 0.1,0.2,0.01 ✓

输出数据为:

利率	15年	20年	25年
10.00%	10.75	9.65	9.09
11.00%	11.37	10.32	9.80
.....			
20.00%	17.56	16.99	16.78





数列按下面递推公式计算得到:

$$F_n = F_{n-1} + F_{n-2} \quad (n \geq 3)$$

这是一个重复计算问题,由于重复次数确定,所以用**DO**循环语句实现。

根据分析，设计算法，编写程序。

输出数据:

1	1	2	3	5	8	13	21	34	55
89	144	233	377	610	987	1597	2584		
4181	6765								

## 7.8 循环结构程序设计举例



[例7.15] 编写程序验证下面公式:

$$1^2 + 2^2 + \cdots + n^2 = \frac{1}{6}n(n+1)(2n+1)$$

- ◆ 例1
- ◆ 例2
- ◆ 例3
- ◆ 例4
- ◆ 例5
- ◆ 例6
- ◆ 例7
- ◆ 例8

解: 设: L表示等式左端的值,R表示等式右端的值。  
分别计算L和R,判断是否相等,如果相等,则等式成立,  
否则不成立。

计算L,通过DO循环语句实现。计算R,通过赋值语句实现。

这是一个重复计算问题,由于重复次数确定,所以用DO循环语句实现。

根据分析,设计算法,编写程序。

输入数据:

输入项数n: 10✓

输出数据:

公式正确!!!

## 7.8 循环结构程序设计举例



- ◆ 例1
- ◆ 例2
- ◆ 例3
- ◆ 例4
- ◆ 例5
- ◆ 例6
- ◆ 例7
- ◆ 例8

[例7.16] 计算并输出所有“水仙花数”。

所谓“水仙花数”是指各位数字的三次方之和等于自身的三位数。

如：371是一个水仙花数。

解：三位数*i*的范围是：100~999, 采用试探法求水仙花数。

根据分析，设计算法，编写程序。

输入数据：（无）

输出数据：

三位数的水仙花数有：

153是一个水仙花数。

370是一个水仙花数。

371是一个水仙花数。

407是一个水仙花数。

三位数的水仙花数有 4 个。

## 7.8 循环结构程序设计举例



- ◆ 例1
- ◆ 例2
- ◆ 例3
- ◆ 例4
- ◆ 例5
- ◆ 例6
- ◆ 例7
- ◆ 例8

[例7.17]输入一个班学生成绩(人数不定)。

按优、良、中、及格和不及格五个等级统计人数,及占总人数的百分比。

优: 90~100,良: 80~89,中: 70~79,及格: 60~69,不及格: 0~59。

解: 设: 学生成绩为 $x$ ,从键盘输入,输入成绩为负数表示输入结束。

学生总人数为 $n$ ,成绩为优的学生人数为 $n1$ ,成绩为良的学生人数为 $n2$ ,成绩为中的学生人数为 $n3$ ,成绩为及格的学生人数为 $n4$ ,成绩为不及格的学生人数为 $n5$ 。根据分析,设计算法,编写程序。

输入数据: 输入学生成绩(整数、占四位、负数结束)!

52 68 79 85 54 98 85 75 70 73 -1 ✓

输出数据: 学生总人数有: 10人。

成绩为优的人数有: 1人,占10.00%。

成绩为良的人数有: 2人,占20.00%。

成绩为中的人数有: 4人,占40.00%。

成绩为及格的人数有: 1人,占10.00%。

成绩为不及格的人数有: 2人,占20.00%。

## 7.8 循环结构程序设计举例



- ◆ 例1
- ◆ 例2
- ◆ 例3
- ◆ 例4
- ◆ 例5
- ◆ 例6
- ◆ 例7
- ◆ 例8

[例7.18]求2~100之间的素数。

所谓素数是指只能被1和自身整除而不能被其它数整除的整数(除1以外)。

解：根据素数定义可知：2、3、5等是素数,4、6、8、9等不是素数。

要判定整数n是否是素数,可用2、3、...、n-1逐个去除n,如果有一个能整除n,则n不是素数,否则是素数。程序中可用逻辑型变量flag作为决定是否素数的标志,若flag=.TRUE.,则为素数,否则不是素数。用整型变量count对素数进行计数。已证明整除至即可。

该问题可用二重循环结构实现。根据分析,设计算法,编写程序。

输入数据：（无）

输出数据：

2至100之间的素数有：

2 3 5 7 11 13 17 19 23 29  
31 37 41 43 47 53 59 61 67 71  
73 79 83 89 97

共有素数25个。



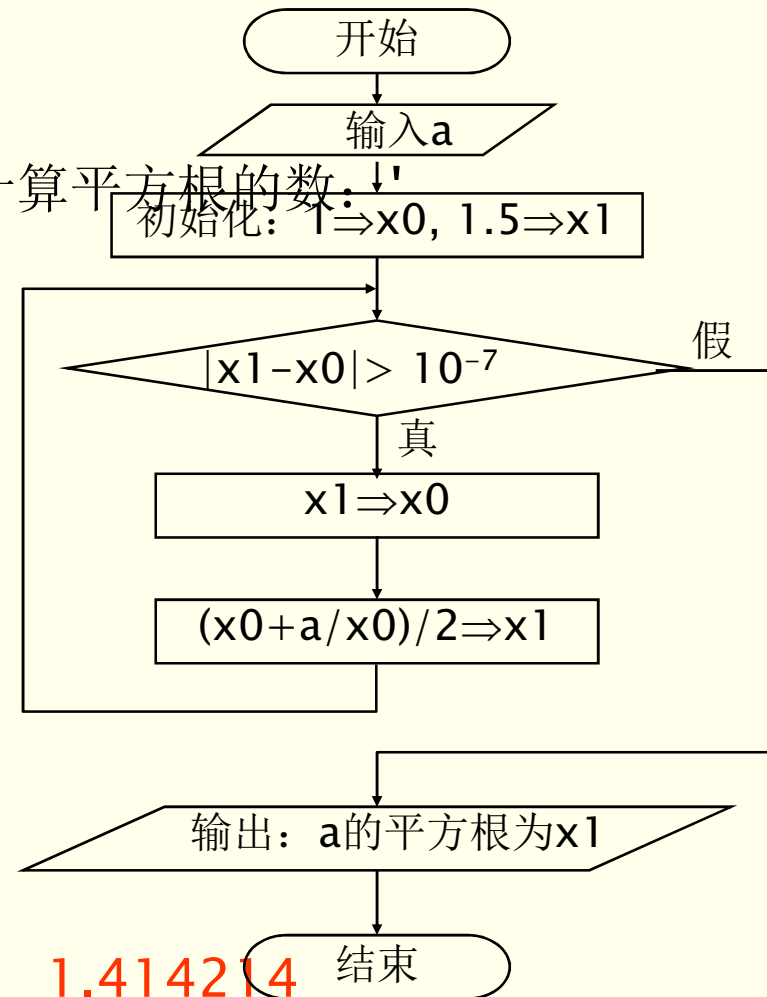
← Back

# 例711算法和程序

```
PROGRAM Newton
IMPLICIT NONE
REAL a,x0,x1
WRITE(*, 10, ADVANCE='NO') '输入计算平方根的数: '
10 FORMAT(A)
READ *, a
x0=1; x1=1.5
DO WHILE (ABS(x1-x0)>1e-7)
  x0=x1
  x1=(x0+a/x0)/2
END DO
PRINT *
PRINT *, a,'平方根为: ', x1
END
```

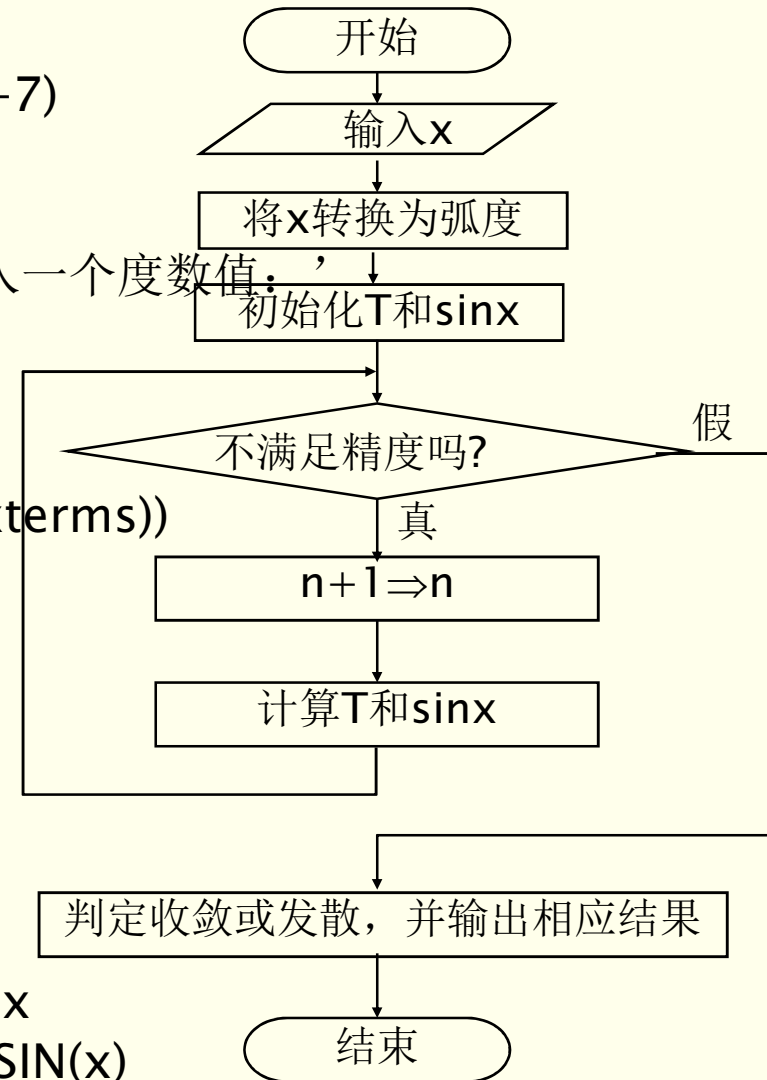
! 输入数据: 输入计算平方根的数: 2✓

! 输出数据为: 2.000000 平方根为: 1.414214



## 例712算法和程序

```
PROGRAM Sin_x
PARAMETER(pi=3.14159278, eps=1E-7)
INTEGER :: n=1, maxterms=10
REAL x, T, sinx
WRITE(*, "(A)", ADVANCE='NO') '输入一个度数值: '
READ*, x
x=x*pi/180 ! 转换为弧度值
T=x;sinx=T
DO WHILE((ABS(T)>ERR).AND.(n<=maxterms))
  n=n+1
  T=-T*x*x/((2*n-2)*(2*n-1))
  sinx=sinx+T
END DO
IF(ABS(T)>eps)THEN
  PRINT*, '发散级数, 不能得到结果。'
ELSE
  PRINT*, '程序计算得到的正弦值: ', sinx
  PRINT*, 'FORTRAN 90内部函数值: ', SIN(x)
END IF
END
```





输入数据:

输入一个度数值: 45✓

输出结果为:

程序计算得到的正弦值: 0.7071068

Fortran 90内部函数值: 0.7071068



输入数据: 0.1,0.2,0.01 ✓

输出结果为:

```
      利率   15年   20年   25年
10.0INTEGER : : ms=12, n
INTEGER cycle_count
REAL : : a=1000, p
!r为年利率, r0为最低利率, r1为最高利率, rc为利率增量
REAL r,r0,r1,rc
READ*, r0,r1,rc          !输入最低利率, 最高利率, 利率增量
cycle_count=INT((r1-r0)/rc)+1 !计算循环次数
r=r0
PRINT*, '利率   15年   20年   25年'
DO i=1, cycle_count
  WRITE(*, "(F5.2, '%')", ADVANCE='NO') 100*R
  DO n=15, 25, 5
    p=r/ms*a*(1+r/ms)**(ms*n)/((1+r/ms)**(ms*n)-1)
    WRITE(*, "(F10.2)", ADVANCE='NO') p
  END DO
  PRINT *
  r=r+rc
END DO
END
0%  10.75  9.65  9.09
    11.00% 11.37 10.32 9.80
    .....
    20.00% 17.56 16.99 16.78
```

## 例713程序

输入数据: 0.1,0.2,0.01✓

输出结果为:

```
      利率  15年  20年  25年
INTEGER : : ms=12, n
INTEGER cycle_count
REAL : : a=1000, p
!r为年利率, r0为最低利率, r1为最高利率, rc为利率增量
REAL r,r0,r1,rc
READ*, r0,r1,rc          !输入最低利率, 最高利率, 利率增量
cycle_count=INT((r1-r0)/rc)+1 !计算循环次数
r=r0
PRINT*, '利率  15年  20年  25年'
DO i=1, cycle_count
  WRITE(*, "(F5.2, '%')", ADVANCE='NO') 100*R
  DO n=15, 25, 5
    p=r/ms*a*(1+r/ms)**(ms*n)/((1+r/ms)**(ms*n)-1)
    WRITE(*, "(F10.2)", ADVANCE='NO') p
  END DO
  PRINT *
  r=r+rc
END DO
END
```

```
10.00% 10.75  9.65  9.09
11.00% 11.37 10.32  9.80
.....
20.00% 17.56 16.99 16.78
```

## 例714程序

```
PROGRAM fibonacci
INTEGER : : F1=1, F2=1, F3
WRITE(*,"(1X,2X,I5,2X,I5\\)") F1, F2
DO i=3, 20
  F3=F2+F1
  WRITE(*, "(2X,I5\\)") F3
  F1=F2; F2=F3
  IF (MOD(i,10)= =0) WRITE(*,"(/,' ')")
ENDDO
END
```

! 输出数据:

```
! 1 1 2 3 5 8 13 21 34 55
! 89 144 233 377 610 987 1597 2584 4181 6765
```

## 例715程序

```
PROGRAM validate
INTEGER : : L=0, R, n
WRITE(*,"(1X,'输入项数n: ')",ADVANCE='NO')
READ *, n
DO i=1, n          !计算等式左端值L
  L=L+i*i
ENDDO
R=n*(n+1)*(2*n+1)/6 !计算等式右端值R
IF (L= =R) THEN
  PRINT *,‘公式正确!!! ’
ELSE
  PRINT *,‘公式不正确!!! ’
ENDIF
END
```

! 输入数据: 输入项数n: 10✓

! 输出数据: 公式正确!!!

## 例716程序

```
PROGRAM narcissus
INTEGER : : i,n1,n2,n3,num=0
PRINT *, '三位数的水仙花数有：'
DO i=100, 999
    n3=i/100      !求百位数字
    n2=(i-n3*100)/10 !求十位数字
    n1=i-n3*100-n2*10 !求个位数字
    temp=n3*n3*n3+n2*n2*n2+n1*n1*n1
    IF (temp= i) THEN
        PRINT *,i,'是一个水仙花数。'
        num=num+1
    ENDIF
ENDDO
PRINT "(1X, '三位数的水仙花数有', I2, ' 个。')", num
END
```

输出结果为：

三位数的水仙花数有：

153是一个水仙花数。

370是一个水仙花数。

371是一个水仙花数。

407是一个水仙花数。

三位数的水仙花数有 4 个。

## 例717程序

```
PROGRAM statistics
INTEGER : : x,n=0,n1=0,n2=0,n3=0,n4=0,n5=0
PRINT *, '输入学生成绩(整数、占四位、负数结束)!'
READ(*,'(I4)',ADVANCE='NO') x
DO WHILE (x>=0)
    n=n+1
    IF (x<=59) THEN
        n5=n5+1
    ELSEIF (x<=69) THEN
        n4=n4+1
    ELSEIF (x<=79) THEN
        n3=n3+1
    ELSEIF (x<=89) THEN
        n2=n2+1
    ELSEIF (x<=100) THEN
        n1=n1+1
    ENDIF
    READ(*,'(I4)',ADVANCE='NO') x
ENDDO
20 FORMAT(1X,'成绩为',A, '的人数有: ', I2, '人, 占', F5.2,'%。')
```

```
PRINT*
PRINT "(1X,'学生总人数有: ', I2, '人。')", n
PRINT 20,'优', n1, 1.0*n1/n*100
PRINT 20,'良', n2, 1.0*n2/n*100
PRINT 20,'中', n3, 1.0*n3/n*100
PRINT 20,'及格', n4, 1.0*n4/n*100
PRINT 20,'不及格', n5, 1.0*n5/n*100
END
```



输入数据:

输入学生成绩(整数、占四位、负数结束)!

52 68 79 85 54 98 85 75 70 73 -1 ✓

输出结果为:

学生总人数有: 10人。

成绩为优的人数有: 1人,占10.00%。

成绩为良的人数有: 2人,占20.00%。

成绩为中的人数有: 4人,占40.00%。

成绩为及格的人数有: 1人,占10.00%。

成绩为不及格的人数有: 2人,占20.00%。

## 例718程序

```
PROGRAM prime
INTEGER : : n, count=0
LOGICAL flag
PRINT *, '2至100之间的素数有: '
DO n=2,100
  Flag=.TRUE.
  DO i=2,n-1
    IF (MOD(n,i)==0) THEN
      Flag=.FALSE.
      EXIT
    ENDIF
  ENDDO
  IF (flag) THEN
    count=count+1
    WRITE(*, "(2X,I2\\)") n
    IF (MOD(count,10) == 0) WRITE(*,"(/\\)")
  ENDIF
ENDDO
PRINT*
PRINT "(1X, '共有素数', I2, '个。')",count
END
```

输出结果为:

2至100之间的素数有:

2 3 5 7 11 13 17 19 23 29

31 37 41 43 47 53 59 61 67 71

73 79 83 89 97

共有素数25个。



## 7.9 习题七

1. 已知:  $x = 0^\circ, 10^\circ, 20^\circ, \dots, 180^\circ$ , 编写程序计算并输出  $x$ 、 $\sin(x)$ 、 $\cos(x)$ 、 $\tan(x)$  的值。
2. 已知:  $x = 1.0, 1.1, 1.2, \dots, 2.9$ , 编写程序计算并输出  $x$ 、 $x^2$ 、 $e^x$ 、 $\ln(x)$  的值。
3. 输入10个整数, 计算它们的和、积、平方和及和的平方。编写程序实现之。
4. 输入20个数, 统计其中整数、负数和零的个数。编写程序实现之。
5. 求满足不等式  $1^1 + 2^2 + 3^3 + \dots + n^n > 10000$  的最小项数  $n$ 。编写程序实现之。
6. 输入100名学生的学号和五门课的成绩, 要求统计并打印出成绩最高者的学号、各门课成绩、总成绩及平均成绩。编写程序实现之。
7. 在1~500中, 找出能同时满足用3除余2, 用5除余3, 用7除余2的所有整数。编写程序实现之。
8. 输入  $x$  值, 按下列公式计算  $\cos(x)$ 。精度要求7位有效数字, 最后一项  $< 10^{-7}$ 。编写程序实现之。

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$



## 7.9 习题七



9. 输入 $x$ 值( $|x| < 1$ ),按下列公式计算 $s$ 。精度要求5位有效数字,最后一项 $< 10^{-5}$ 。编写程序实现之。

$$s = x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \dots$$

10. 输入 $x$ ,用迭代法按下列迭代公式求  $y = x^{\sqrt{x}}$  的值。初始值 $y_0 = x$ ,精度要求4位有效数字。编写程序实现之。

$$y_{n+1} = \frac{2}{3}y_n + \frac{x}{3y_n^2}$$

11. 编写程序计算 $f_{ij}, s_i$ ,其中:  $i=1,2,3,4,5; j=1,2,3,4,5,6,7,8,9,10$ 。已知:

$$x_i = 1.25, 2.30, 3.35, 4.40, 5.45$$

$$y_j = 3.0, 3.1, 3.2, 3.3, \dots, 3.9$$

$$f_{ij} = e^{-x_i} \ln(x_i + y_j)$$

$$s_i = \prod_{j=1}^{10} f_{ij}$$



## 7.9 习题七



12. 编写程序验证哥德巴赫猜想(任何充分大的偶数都可由两个素数之和表示)。将4至100之间所有偶数分别用两个素数之和表示。输出格式为:

$$4=2+2$$

$$6=3+3$$

$$8=3+5$$

.....

$$100=3+97$$

13. 编写程序求2至10000之间的所有“完数”。所谓“完数”是指除自身之外的所有因子之和等于自身的数。

如28是一个完数,因为28的因子有: 1、2、4、7、14,,且:

$$28=1+2+4+7+14$$





14. 编写程序输出以下图形。

a b  
a b c  
:  
a b c ... x y z

15. 编写程序输出以下图形。

1  
1 2 1  
1 2 3 2 1  
1 2 3 4 3 2 1  
1 2 3 4 5 4 3 2 1  
1 2 3 4 5 6 5 4 3 2 1  
1 2 3 4 5 6 7 6 5 4 3 2 1  
1 2 3 4 5 6 7 8 7 6 5 4 3 2 1  
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1



h e

e n d

文件名格式：班级 学号 姓名 简略实验名称

邮件标题同文件名

Any questions please 发送至

xingzhengwu@163.com