

第4章 流程控制语句

南通职业大学 朱亚林



内容提要

- 顺序结构程序设计
- 选择结构程序设计
- 循环结构程序设计



1. 顺序结构程序设计

- 算法的概念

算法是对特定问题求解步骤的描述，是指令的有限序列。

- 算法必须具备下列五个特性：

有穷性：算法需在有穷步后结束

② 确定性：算法的每一步要有明确含义

③ 可行性：算法的每一步需能通过一些基本操作的迭代去实现

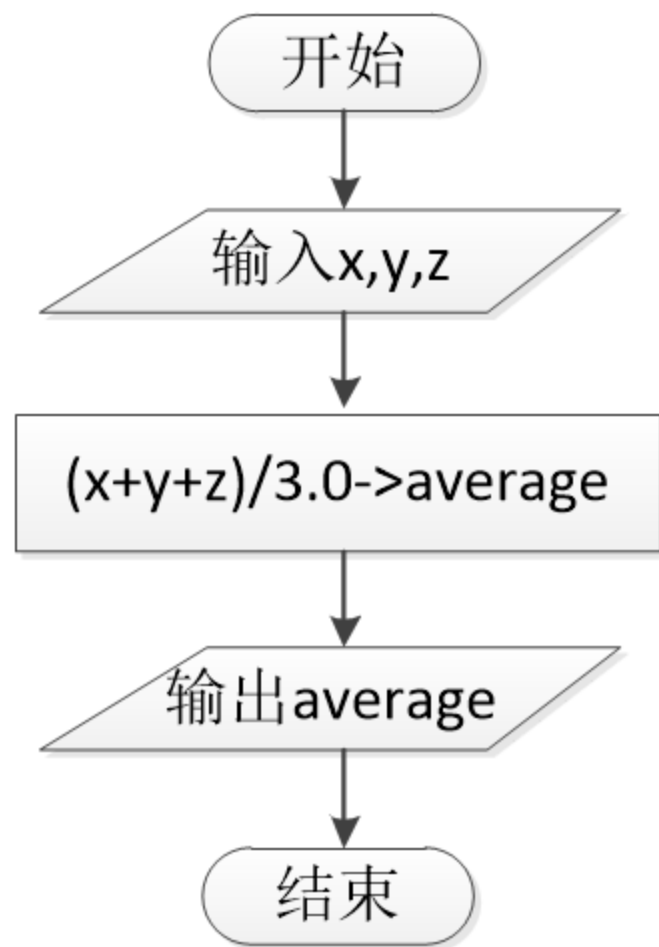
④ 输入：算法可以有0到多个输入

⑤ 输出：算法必须有输出

1.顺序结构程序设计

算法的描述方法：**流程图方式**

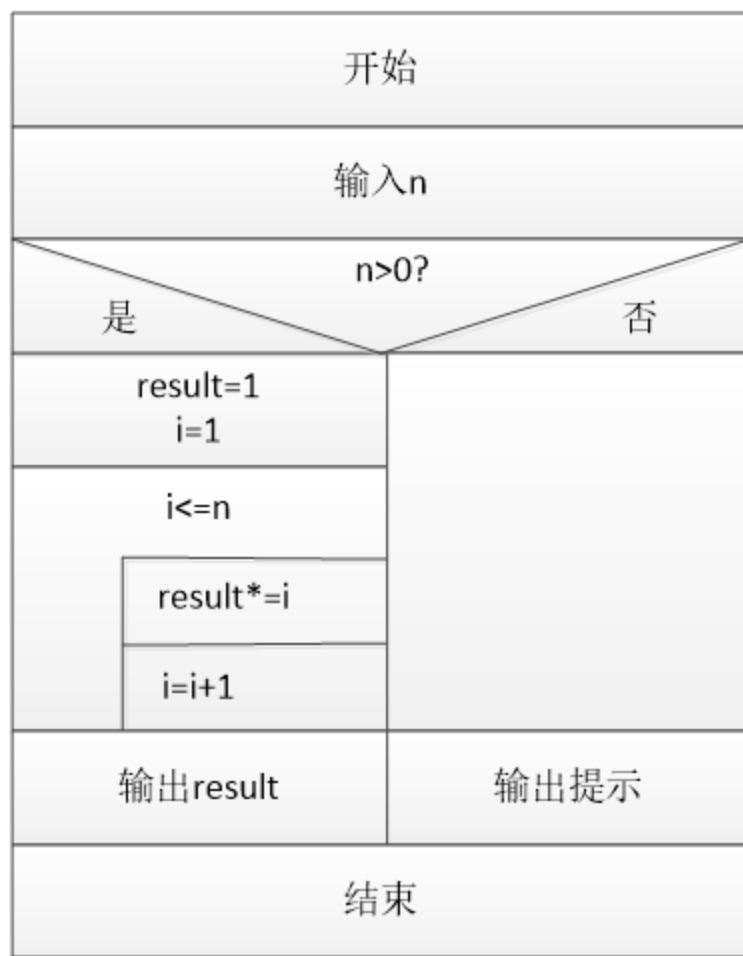
【例4-1】设计算法，求解键盘输入的三个数平均值。



1.顺序结构程序设计

算法的描述方法：N-S图方式

【例4-2】设计算法，求解n的阶乘，
画出N-S图。



1.顺序结构程序设计

【例4-3】编写程序，输入三角形的三边长（设满足构成条件），计算并输出三角形的面积。

分析：设三边长变量分别为 a 、 b 、 c ，则三角形面积计算公式为： $area = \sqrt{s(s-a)(s-b)(s-c)}$ ，其中 $s = ((a+b+c))/2$ 。

```
#exp4-3.py
```

```
import math    #导入math库，以便使用开根号函数  
a,b,c=eval(input('请输入三角形的三边长,逗号间隔: ')) s=(a+b+c)/2.0  
area=math.sqrt(s*(s-a)*(s-b)*(s-c)) #调用开根号函数  
print("area=%6.2f"%area);
```

```
请输入三角形的三边长,逗号间隔: 3,4,5  
area= 6.00
```

1.顺序结构程序设计

【例4-4】编写程序，将键盘输入的三位整数逆置输出。

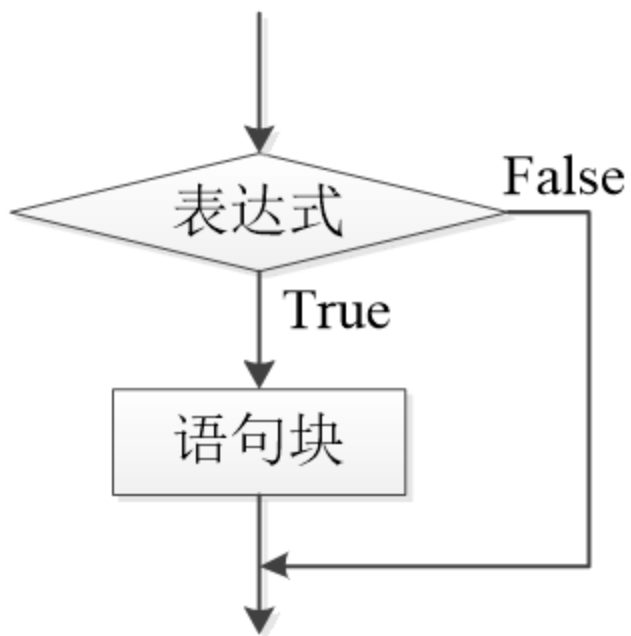
分析：利用整除和求余运算符分别将整数的个位、十位和百位取出，然后从高位到低位输出。

```
#exp4-4.py
x=int(input("请输入一个三位整数： "))
a=x//100
b=x//10%10
c=x%10
print("百位： %d 十位： %d 个位： %d\n"%(a,b,c))
```

请输入一个三位整数： 321
百位： 3 十位： 2 个位： 1

2.选择结构程序设计

■ 单分支



格式:

```
if 表达式:
```

```
    语句块
```

说明:

(1)语句块要用**Tab**键或空格键向右缩进，下同；

(2)表达式后有分号。

2.选择结构程序设计

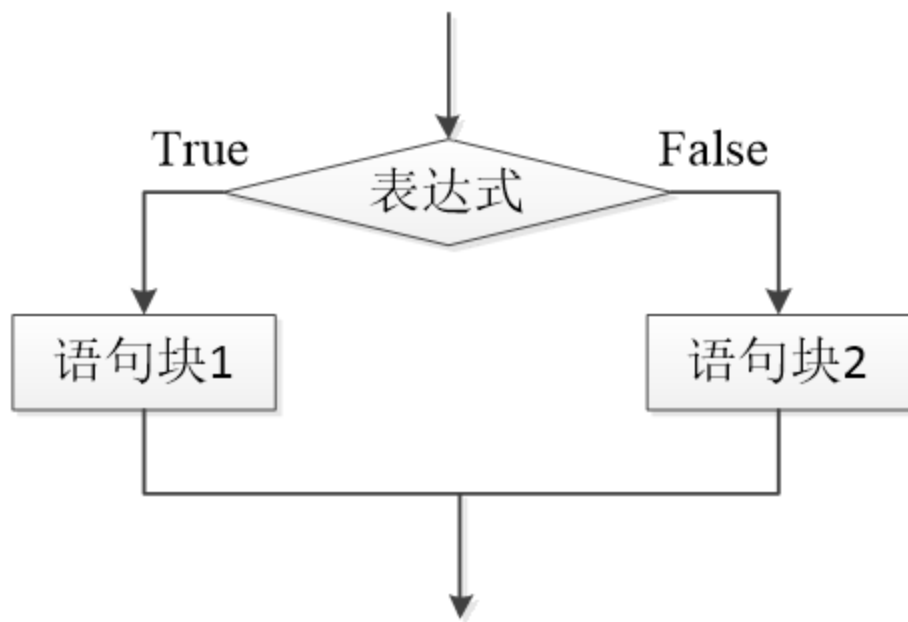
■ 单分支示例

```
#test.py
x=int(input("请输入一个数: "))
if x<0:
    x=-x
print("|x|=%d\n"%(x))
```

请输入一个数: -123
|x|=123

2.选择结构程序设计

■ 双分支



格式:

if 表达式:

 语句块1

else:

 语句块2

说明:

(1)语句块要用**Tab**键或空格键向右缩进;

(2)表达式和**else**后有分号。

2.选择结构程序设计

■ 双分支示例

【例4-6】判断输入的年份是否为闰年。

分析：闰年判断方法：年份能被4整除而不能被100整除，或年份能被400整除。

```
#exp4-6.py
year=eval(input('请输入年份:'))
if (year%4==0 and year%100!=0) or year%400==0:
    print("%d年是闰年!\n"%year)
else:
    print("%d年不是闰年!\n"%year)
```

请输入年份:2020
2020年是闰年!

请输入年份:2019
2019年不是闰年!

2.选择结构程序设计

【例4-7】完善【例4-3】，要求能对输入的三个数能否构成三角形进行合法性判断，即任何两边之和大于第三边才能进行面积计算，否则给出提示。

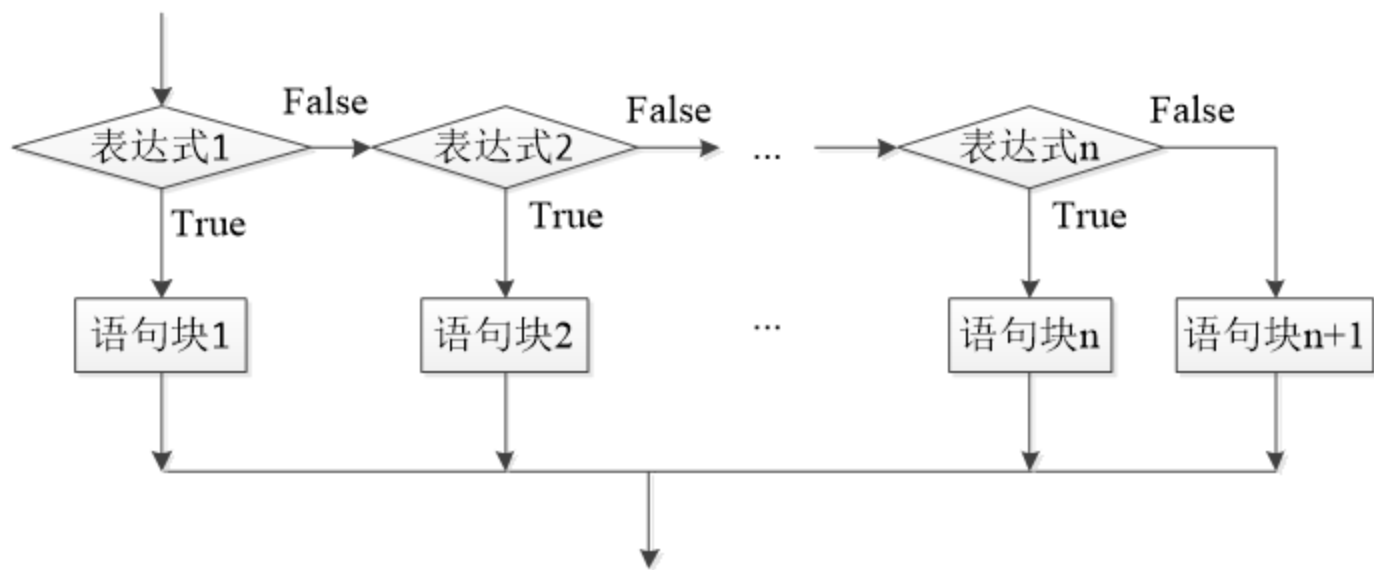
```
#exp4-7.py
import math
a,b,c=eval(input('请输入三角形的三边长,逗号间隔: '))
if a+b>c and a+c>b and b+c>a:
    s=(a+b+c)/2.0
    area=math.sqrt(s*(s-a)*(s-b)*(s-c))
    print("area=%6.2f"%area);
else:
    print("不能构造三角形")
```

请输入三角形的三边长, 逗号间隔: 3, 4, 5
area= 6.00

请输入三角形的三边长, 逗号间隔: 3, 4, 7
不能构造三角形

2.选择结构程序设计

■ 多分支结构



说明:

(1) 一旦有条件成立, 就不再判断后面的条件, 执行完语句块, 分支语句就结束了;

(2) 语句块n+1前面不需再增加判断条件。

2.选择结构程序设计

■ 多分支语句格式:

if 表达式1:

 语句块1

elif 表达式2:

 语句块2

elif 表达式3:

 语句块3

...

else:

 语句块 $n+1$

说明:

(1) elif相当于else if

(2) elif要与前面的if对齐

(3)表达式和else后有分号。

2.选择结构程序设计

【例4-8】将键盘输入的百分制成绩转换成五分制。转换规则：90~100(优)，80~89(良)，70~79(中)，60~69(及格)，60分以下(不及格)，其它成绩非法提示。

```
#exp4-8.py
score=int(input('请输入成绩: '))
if score>100 or score<0:
    print("wrong score! (eg:0~100)")
elif score>=90:
    print("优秀")
elif score>=80:
    print("良好")
elif score>=70:
    print("中等")
elif score>=60:
    print("及格")
else:
    print("不及格")
```

请输入成绩: 80
良好

请输入成绩: 92
优秀

请输入成绩: 55
不及格

2.选择结构程序设计

【例4-9】阶梯电费计算。

$$y = \begin{cases} 0.56 * x & (x \leq 180) \\ 180 * 0.56 + (x - 180) * 0.61 & (180 < x \leq 260) \\ 180 * 0.56 + 80 * 0.61 + (x - 260) * 0.86 & (x > 260) \end{cases}$$

```
#exp4-9.py
x=int(input('请输入本月用电量: '))
if x<=180:
    y=0.56*x
elif x>180 and x<=260:
    y=180*0.56+(x-180)*0.61
else:
    y=180*0.56+80*0.61+(x-260)*0.86
print("用电量为: %d时, 电费为: %.2f\n"%(x,y))
```

请输入本月用电量: 220
用电量为: 220时, 电费为: 125.20

2.选择结构程序设计

【例4-10】一元二次方程根的求解。其中各项系数由键盘输入。

分析：设一元二次方程形式为 $ax^2+bx+c=0(a\neq 0)$ ，则方程的根有以下三种形式：（1） $b^2-4ac=0$ ，有两个相等实根；（2） $b^2-4ac>0$ ，有两个不等实根；（3） $b^2-4ac<0$ ，无实数根。

```
#exp4-10.py
import math
a,b,c=eval(input('请输入方程的各项系数：'))
delt=b*b-4*a*c
if delt==0:
    print("两个相等实根为： %f"%(-b/(2*a)));
elif delt>0:
    x1=(-b+math.sqrt(delt))/(2*a)
    x2=(-b-math.sqrt(delt))/(2*a)
    print("实数根1： %f,实数根2： %f"%(x1,x2))
else:
    print("无实数根")
```

请输入方程的各项系数： 1, 2, 1
两个相等实根为： -1.000000

请输入方程的各项系数： 1, -5, 6
实数根1： 3.000000, 实数根2： 2.000000

2.选择结构程序设计

- 选择结构的嵌套：当语句块又是分支结构时就会形成各种嵌套分支形式。

格式：

if 表达式1:

 if 表达式2:

 语句块1

 else:

 语句块2

说明：

(1)else总与离它最近又没有配对的if配对；

(2)注意语句块的缩进；

(3)表达式和else后有分号。

2.选择结构程序设计

【例4-12】 键盘输入两个整数，请比较两者间的大小关系。

分析：利用嵌套分支进行判断。

```
#exp4-12.py
print("请输入两个整数：");
a=int(input('a:'))
b=int(input('b:'))
if a!=b:
    if a>b:
        print("%d > %d"%(a,b));
    else:
        print("%d < %d"%(a,b));
else:
    print("%d = %d"%(a,b))
```

请输入两个整数：

a:3

b:4

3 < 4

请输入两个整数：

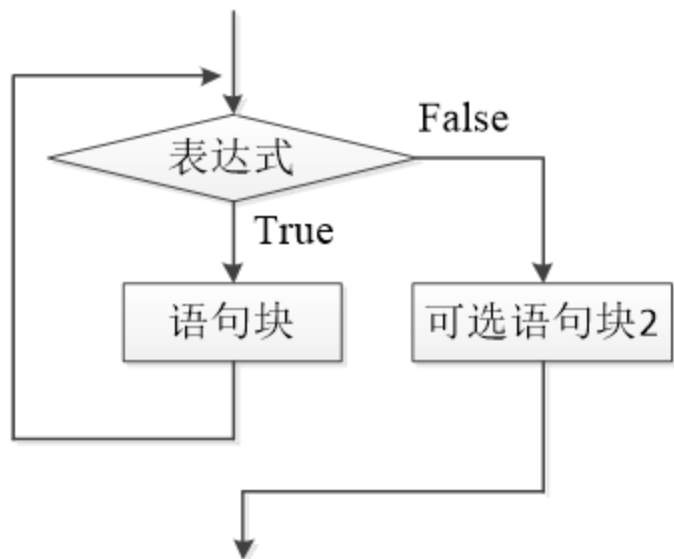
a:5

b:5

5 = 5

3.循环结构程序设计

■ while语句



格式:

while 条件表达式:

语句块1

[else:

语句块2]

说明:

- (1) 条件表达式可以是任意类型，但一般是用关系表达式或逻辑表达式；
- (2) 当语句块有多条语句时，缩进时要对齐；
- (3) 若循环体只有一条语句时，可与while写在同一行；
- (4) 若有else则表示条件为假时，将执行语句块2；
- (5) 为了避免无限循环，循环体中需有修改循环变量的语句，确保循环条件不会一直为真。也可以通过break语句来中断循环。

3.循环结构程序设计

■ while语句示例

【例4-13】求 $s=1+2+\dots+n$ ，其中 n 由键盘输入。

分析：这是一个有规律数的累加问题。可通过循环变量保存累加项，每次累加后循环变量加1。

```
#exp4-13.py
n=int(input("请输入正整数n="))
s=0
i=1
while i<=n:
    s=s+i
    i=i+1
print("s=%d\n"%s);
```

请输入正整数n=100
s=5050

3. 循环结构程序设计

■ while语句示例

【例4-14】判断由键盘输入的整数是几位数，并输出其每位上的数字。

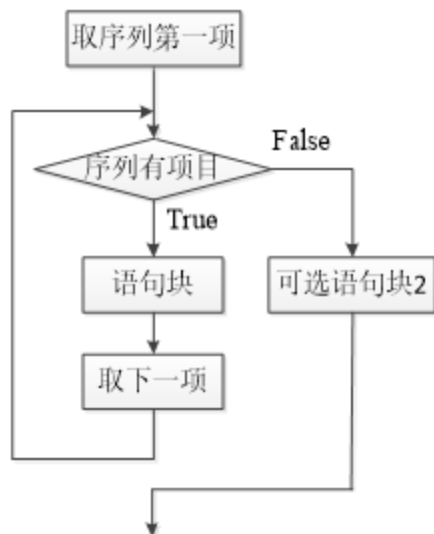
分析：利用求余运算求出整数的个位并输出，然后用整除运算丢掉个数。

```
#exp4-14.py
n=int(input("请输入正整数n="))
i=0
print("该数从低位到高位依次为: ");
while n>0:
    print("%d"%(n%10))
    i=i+1
    n=n//10
print("该数是%d位数.\n"%(i));
```

请输入正整数n=54321
该数从低位到高位依次为：
1
2
3
4
5
该数是5位数。

3. 循环结构程序设计

■ for语句



说明:

- (1) 常用列表、元组、字符串来作为序列对象组织for循环;
- (2) 当语句块1有多条语句时，缩进时要对齐;
- (3) 若语句块1只有一条语句时，可与for写在同一行;
- (4) 若有else则表示序列对象中没有项目时，将执行语句块2;
- (5) 循环体可以通过break语句来中断循环;
- (6) Python 3.X中可用内建函数range()来产生序列对象控制for循环，range()函数格式为：`range([start] stop [,step])` 不含stop

格式:

for 循环变量 in 序列:

语句块1

[else:

语句块2]

3.循环结构程序设计

■ for语句示例

【例4-15】用for循环求 $s=1+2+\dots+n$ ，其中n由键盘输入。

分析：利用range()来产生指定范围的序列对象。

```
#exp4-15.py
n=int(input("请输入正整数n="))
s=0
for i in range(1,n+1):
    s=s+i
print("s=%d\n"%(s));
```

请输入正整数n=100
s=5050

3. 循环结构程序设计

for语句示例

【例4-16】判断由键盘输入的数n是否为素数。素数也称为质数，是指只能被1和自身整除的数。

分析：根据定义一般只需测试n是否能被2, 3, ..., n/2(或 \sqrt{n})整除，只要能被其中一个数整除，则n就不是素数。

```
#exp4-16-1.py
n=int(input("请输入正整数n="))
flag=1
for i in range(2,n//2+1): #序列结束值采用n//2
    if n%i==0:
        flag=0
        break          #结束循环
if flag==1:
    print("%d是素数"%(n));
else:
    print("%d不是素数"%(n));
```

请输入正整数n=15
15不是素数

请输入正整数n=17
17是素数

```
#exp4-16-2.py
import math
n=int(input("请输入正整数n="))
flag=1
for i in range(2,int(math.sqrt(n))+1): #序列结束值采用 $\sqrt{n}$ 
    if n%i==0:
        flag=0
        break
if flag==1:
    print("%d是素数"%(n));
else:
    print("%d不是素数"%(n));
```

3.循环结构程序设计

■ for语句示例

【例4-18】输出所有的三位水仙花数。所谓水仙花数是指该数各位数字的立方和等于该数本身。

分析：先对三位数进行数字分解，然后判断各位数字的立方和是否等于该数本身。

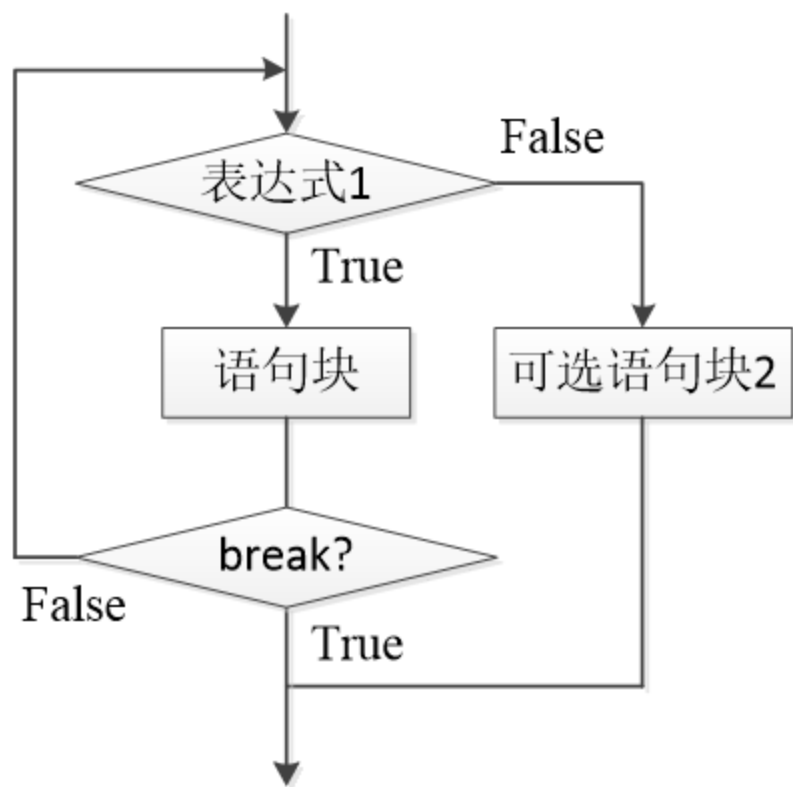
```
#exp4-18.py
print("水仙花数为: ",end=" ");
for i in range(100,1000):
    a=i//100
    b=i//10%10
    c=i%10
    if a**3+b**3+c**3==i:
        print(i,end=" ");
```

水仙花数为： 153 370 371 407

3. 循环结构程序设计

■ break语句和continue语句

➤ break语句用于结束循环体的执行，从循环中跳出来转向循环后面的语句。



#测试break语句的作用

```
for i in range(1,11):
```

```
    if i%5==0:
```

```
        break
```

```
    else:
```

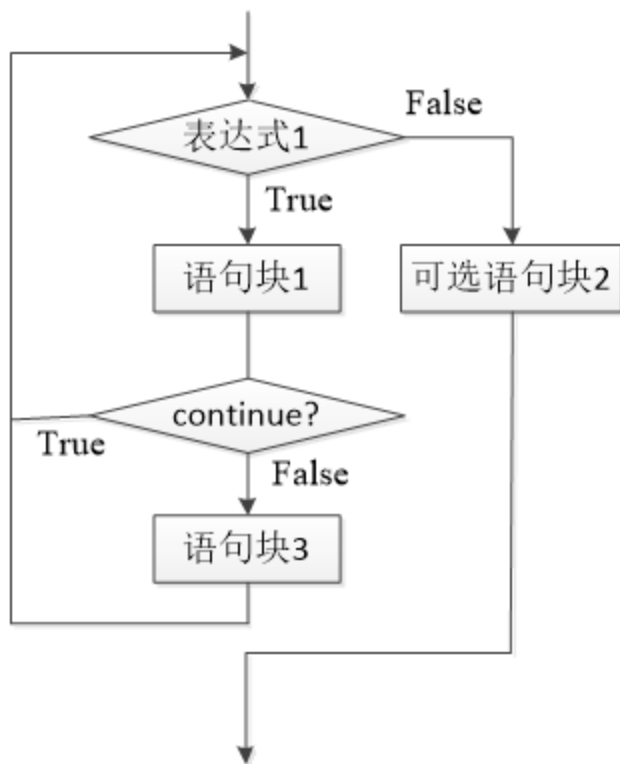
```
        print(i,end=" ")
```

1 2 3 4

3. 循环结构程序设计

■ break语句和continue语句

- **continue**语句用于终止本次循环体的执行，跳过本次循环体中剩余的语句，开始下一次循环，直到循环条件不成立为止。



#测试continue语句的作用

```
for i in range(1,11):
```

```
    if i%5==0:
```

```
        continue
```

```
    print(i,end=" ")
```

1 2 3 4 6 7 8 9

3.循环结构程序设计

■ 循环的嵌套

如果一个循环的循环体内又包含了循环结构，则称为循环的嵌套，也称为多重循环。

```
for 变量1 in 序列1
    语句块1
    for 变量2 in 序列2
        语句块2
    语句块3
```

```
while 条件1:
    语句块1
    while 条件2:
        语句块2
    语句块3
```

3. 循环结构程序设计

■ 循环的嵌套示例

【例4-19】输出九九乘法表

说明：外循环控制行数，内循环控制每行输出的乘式数。注意内循环的终值为外循环变量的当前取值。

```
#exp4-19.py
for i in range(1,10):
    str=""
    for j in range(1,i+1):
        str=str+"%d*%d=%-2d "%(i,j,i*j) #将乘法算式拼接成字符串
    print(str)
```

```
1*1=1
2*1=2 2*2=4
3*1=3 3*2=6 3*3=9
4*1=4 4*2=8 4*3=12 4*4=16
5*1=5 5*2=10 5*3=15 5*4=20 5*5=25
6*1=6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36
7*1=7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49
8*1=8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64
9*1=9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81
```

3. 循环结构程序设计

■ 循环的嵌套示例

【例4-20】输出100以内所有的素数，每行输出10个。

说明：外循环控制判断范围，内循环判断每个外循环变量是否为素数。并增加了统计变量n，当其是10的倍数时换行。

```
#exp4-20.py
n=0
str=""
print('100内的素数为：')
for i in range(2,101):
    flag=1
    for j in range(2,i//2+1):
        if i%j==0:
            flag=0
            break    #结束循环
    if flag==1:
        n+=1
        str=str+"%-2d"%(i)
        if n%10==0:
            print(str)
            str=""
print(str)
```

100内的素数为：

```
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
```

3. 循环结构程序设计

■ 循环的嵌套示例

【例4-21】 现有两个列表，list1=['赵', '钱', '孙', '李'], list2=[1,2]。要求从两个表中各取一个元素形成新的列表。

说明：外循环控制从第一个列表取值，内循环控制从第二个列表取值，利用append方法添加组合数据到第三个列表中。

```
#exp4-21.py
list1=['赵','钱','孙','李']
list2=[1,2]
list3=[]
for i in list1:
    for j in list2:
        list3.append([i,j])
print('重组后的列表为：')
print(list3)
```

重组后的列表为：
[['赵', 1], ['赵', 2], ['钱', 1],
['钱', 2], ['孙', 1], ['孙', 2],
['李', 1], ['李', 2]]

4.循环结构程序设计综合案例

【例4-23】求键盘输入两整数的最大公约数和最小公倍数。

分析：可以用辗转相除法求最大公约数：利用大数除以小数，若余数不为0，则将小数送给大数，余数送给小数；余数为0时，小数变量中的数即为最大公约数。最小公倍数等于两数乘积除以最大公约数。

4.循环结构程序设计综合案例

【例4-24】 利用下式求解 π 的近似值，直到累加项的绝对值小于 10^{-5} 为止。

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} \dots$$

分析： 这是一个累加问题，利用累加项的绝对值作为循环条件。

4.循环结构程序设计综合案例

【例4-25】将一个正整数分解质因数。如：输入90，输出 $90=2*3*3*5$ 。

分析：设待分解的数为 x ，分解质因数过程可按以下步骤进行：

- (1) 首先找到 x 的一个最小质因数 $k(2\sim x)$ ；
- (2) 若这个质因数 k 等于 x ，则说明分解结束，跳出循环，打印最后一个质因数 x ；
- (3) 若这个质因数 k 不等于 x ，但因 x 能被 k 整除，所以 k 是 x 的一个质因数，应打印出 k 的值，并用 x 整除 k 的商作为新的正整数 x ，重复执行第(1)步。

4.循环结构程序设计综合案例

【例4-26】输出右边菱形图案。

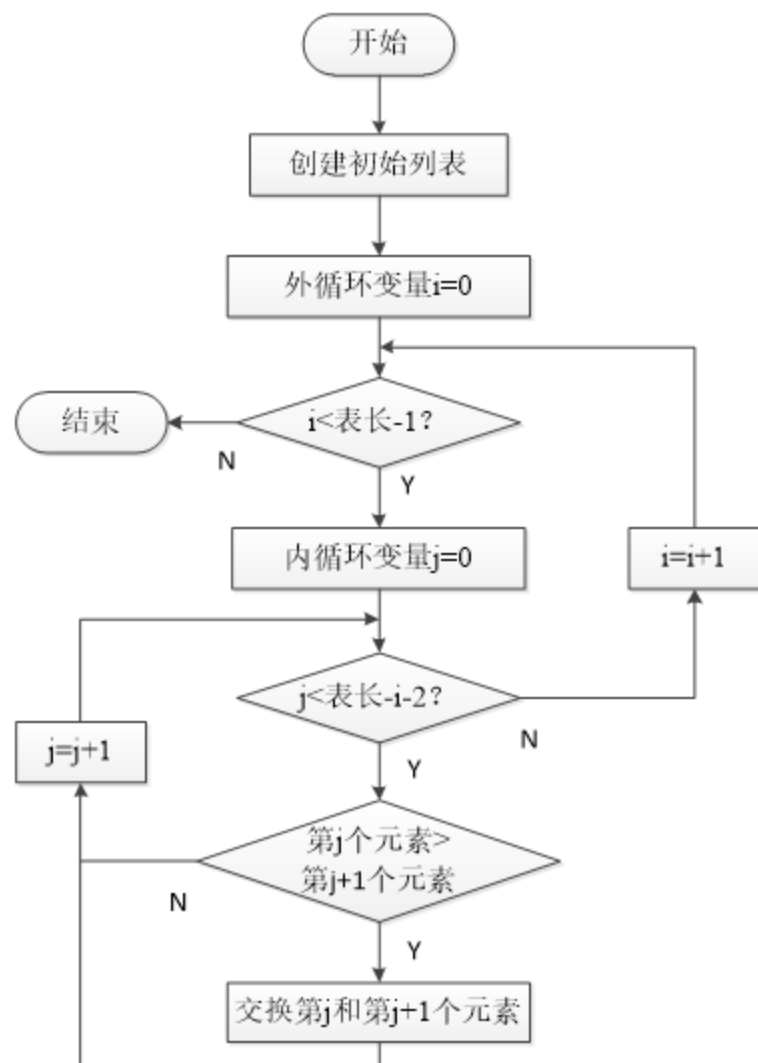
分析：利用两个嵌套循环。第一个双重循环控制前6行输出，每行先输出递减个数的空格，然后再输出*；第二个双重循环控制后5行输出，每行先输出递增个数的空格，然后再输出*。每行结束要换行。

```
      *
     ***
    *****
   *********
  ***********
 *****
  *****
   *****
    *****
     *****
      *****
       *
```

4.循环结构程序设计综合案例

【例4-27】排序问题。设有一批整数保存在列表中，请将列表排成有序表。

方法一：采用**冒泡排序**。冒泡排序思想：每一趟在无序序列中，通过相邻元素比较确定一个最大（升序）元素的位置。 N 个关键字序列，最多只需 $N-1$ 趟即可完成排序。具体排序思想可参考流程图。



4.循环结构程序设计综合案例

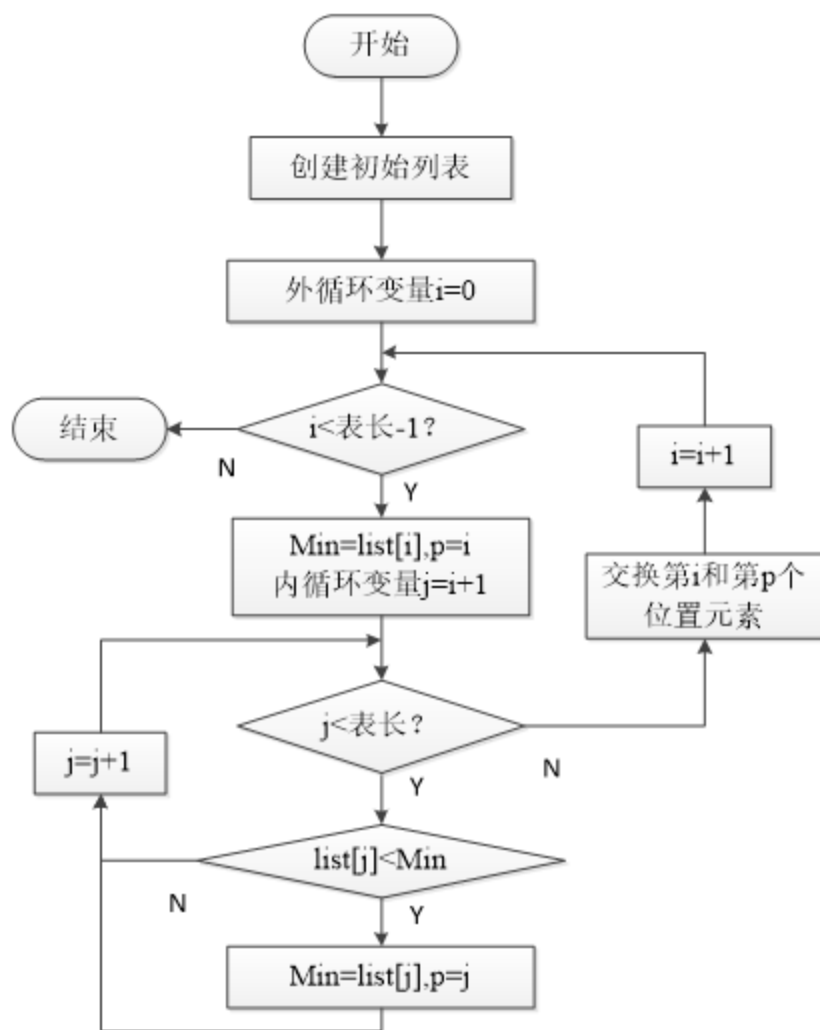
冒泡排序逻辑过程

初始	1	2	3	4	5	6
49	38	38	38	38	13	13
38	49	49	49	13	27	27
65	65	65	13	27	38	38
97	76	13	27	49	49	49
76	13	27	49	49	49	49
13	27	49	65	65	65	65
27	49	76	76	76	76	76
49	97	97	97	97	97	97

4.循环结构程序设计综合案例

【例4-27】排序问题。设有一批整数保存在列表中，请将列表排成有序表。

方法二：采用**选择排序**。选择排序思想：每一趟在无序序列中，选择一个最小（升序）元素与当前位置的元素交换。**N**个关键字序列，最多只需**N-1**趟即可完成排序。具体排序思想可参考流程图。



4.循环结构程序设计综合案例

选择排序逻辑过程

小结

顺序结构:介绍了算法的概念、特性与算法描述方法

01

选择结构:介绍了选择结构的不同形式
(单分支、双分支、多分支和嵌套形式)

02

循环结构:介绍了循环结构的不同形式
(while语句、for语句和嵌套形式),同时介绍了break语句和continue语句的作用。

03

练习

1. 以下对Python程序缩进格式描述错误的选项是_____。
 - A. 不需要缩进的代码顶行写,前面不能留空白
 - B. 缩进可以用tab键实现,也可以用多个空格实现
 - C. 严格的缩进可以约束程序结构,可以多层缩进
 - D. 缩进是用来格式美化Python程序的

练习

2. 下列关于分支结构描述中，正确的是_____。

- A. 分支结构必须要有else子句
- B. if-else构成双分支结构
- C. 双分支结构有一种紧凑形式,使用保留字if和elif实现
- D. 分支结构的嵌套只能在else子句中

练习

3. 下列关于循环结构的描述中，错误的是_____。

- A. break语句可用于终止循环执行
- B. for循环可以用while循环改写
- C. while循环只能实现无限循环
- D. while语句和for语句可以相互嵌套

练习

4、下列程序输出结果是_____。

```
x=[10,20,30,40]
```

```
s=0
```

```
for i in x:
```

```
    if i>30:
```

```
        break
```

```
    s=s+i
```

```
print(s)
```

A、 30

B、 60

C、 100

D、 0