

# 第4讲 MATLAB程序设计

## 1 程序结构

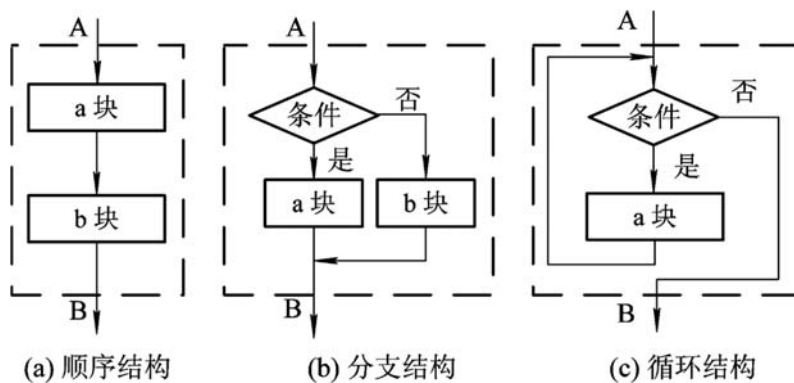
## 2 循环程序

## 3 分支结构



## 1 MATLAB 程序结构

MATLAB作为一种常用的编程语言，支持各种决策或流程控制结构。流程控制极其重要，因为它用过去、现在的计算影响将来的结果。MATLAB程序结构分为顺序结构、分支结构和循环结构。



## 2 循环程序

### 2.1 for循环

#### 1. for循环的一般形式

for循环允许一组命令以固定的和预定的次数重复。For循环的一般形式：

**for** *variable* = values

    commands

**end**

commands是循环体语句组，这里的循环语句是以end结尾的。

values可以是下表中格式之一。

initval:endval	索引变量以增量形式循环，从初始值 <code>initval</code> 到结束值 <code>endval</code> ，增量步长为 1，并重复循环
initval:step:endval	增量步长为 <code>step</code> 代表的值，重复迭代循环
valArray	从 <code>valArray</code> 数组的子列，为每个迭代创建一个列向量的索引。例如，在第一次迭代，指数= <code>valArray(:, 1)</code> 。循环执行 <code>n</code> 次，其中 <code>n</code> 是 <code>valArray</code> 列的列数

(1) initval:step:endval形式。索引变量以增量形式循环，从初始值initval到结束值endval，增量步长为step，并重复循环。如果step为负数，则以减量方式循环，该形式的初始值initval 大于结束值endval。

```
for s=s1:s2:s3
    commands
...
end
```



例 求出  $x = \sum_{i=1}^{100} i$  的值。  
解 该例可以作下列的循环：

```
x=0;
for i=1:1:100
    x=x+i;
end;
x
>> x
结果为x = 5050
```



(2) valArray形式。在每个迭代，从数组valArray列的子列创建一个列向量index。例如，在第一次迭代， $\text{index} = \text{valArray}(:, 1)$ 。循环执行n次，其中n是valArray列数，即for循环按照数组的列数决定循环次数。

```
for index = V
    commands
end
```



例如：

```
A = [ 0.8147    0.6324    0.9575    0.9572    0.4218;
      0.9058    0.0975    0.9649    0.4854    0.9157;
      0.1270    0.2785    0.1576    0.8003    0.7922;
      0.9134    0.5469    0.9706    0.1419    0.9595];
```

```
for n = A
    s = n * 2
end
```

第5次循环结果：

```
0.8436
1.8314
1.5844
1.9190
```



例如，在命令行输入：

```
>> V=[1 2 3 5 7];  
for n=V  
    x(n)=sin(n*pi/10);  
end
```

```
V=ones(2,3,4);  
x=0;  
for n=V  
    x=x+n;  
end
```

则输出结果：

```
>> x
```

```
x = 0.3090 0.5878 0.8090 0 1.0000 0 0.8090
```



实例：

求  $S=1+2*3+3*4+4*5+\dots+n(n+1)$ ，其中  $n$  由键盘输入。

注：键盘输入函数为 `input()`

实现过程：

```
n=input('请输入正整数n= '); %提示输入正整数n  
S=1;  
for i=2:n  
    S=S+i*(i+1);  
end  
S
```

MATLAB高效实现过程： `S=sum([1:n].*[2:n+1])-1`



实例:

求  $S=1+2*3+3*4+4*5+\dots+n(n+1)$ , 其中n由键盘输入。

注: 键盘输入函数为 **input** ( )

```
x = input('What is the original value?');  
y = x*10
```

```
str = input('Do you want more? Y/N [Y]: ','s');  
if isempty(str)  
    str = 'Y';  
end  
str
```



## 2. for循环的嵌套

在一个for循环中, 可以根据需要嵌套另外的多个for循环。例如:

```
for n=1:5  
    for m=5: -1:1  
        B(n,m)=n^2+m^2;  
    end  
    disp(n)  
end
```



例 列出构成hibert矩阵的程序。

$$\begin{bmatrix} 1 & 1/2 & 1/3 & \dots & 1/n \\ 1/2 & 1/3 & 1/4 & \dots & 1/(n+1) \\ 1/3 & 1/4 & 1/5 & \dots & 1/(n+2) \\ \dots & \dots & \dots & \dots & \dots \\ 1/n & 1/(n+1) & 1/(n+2) & \dots & 1/(2n-1) \end{bmatrix}$$



```
format rat
n=input('n=');
for i=1:n
    for j=1:n
        h(i,j)=1/(i+j-1);
    end
end
```

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



## 2.2 while 循环

与for循环以固定次数求一组命令的值相反，while循环以不定的次数求一组语句的值。while循环的一般形式是

**while** expression

    commands

**end**

只要表达式expression里的所有元素都为真，就继续执行while和end 语句之间的commands。通常，表达式的求值给出一个标量值，但数组值也同样有效。在数组情况下，所得到数组的所有元素必须都为真。



例 while循环。

用while实现1到100数字序列的和

```
x = 0;  
i=1;  
while i<=100  
    x=x+i;  
    i=i+1;  
end  
x
```





例 用循环求解  $\sum_{i=1}^m i > 1000$  中求最小的  $m$  值。

解 程序如下：

```
s=0; m=0;  
while s<=1000  
    m=m+1;  
    s=s+m;  
end;  
[s,m]
```



## 2.3 break语句

**break**命令强迫for循环或while循环提早结束，当执行**break**语句时，MATLAB跳到循环体外下一个语句。

如果一个**break**语句出现在一个嵌套的for循环或while循环结构里，那么MATLAB只跳出**break**所在的那个循环，不跳出整个嵌套结构。



break终止循环示例：

k从1循环到10，如果遇到k等于4，break跳出循环。

```
for k=1:10
    if k==4
        break;
    end
    k
end
```

k =	1
k =	2
k =	3



## 2.4 continue语句

continue命令一般用在for循环或while循环中，通过if语句使用continue命令，当满足语句if条件时，continue命令被调用。与break语句不同的是，执行continue命令后，系统只是不再执行相关命令，而不跳出当前循环体外。



continue终止循环示例。

for k=1:7	k = 1
if k==4	k = 2
continue;	k = 3
end	k = 5
k	k = 6
end	k = 7



## 3 分支结构

### 3.1 条件转移结构

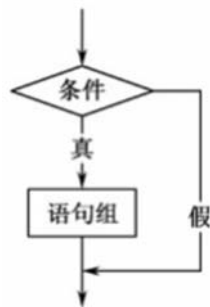
条件转移结构包括以下三种：if...end、if...else...end、if...elseif...else...end。判断表达式紧跟在关键字if后面，使得它可以首先被计算，判断其值为真否。若计算判断表达式的结果为1，判断值为真，则执行其后的执行语句；若结果为0，判断值为假，则跳过、不予执行。



## 1. if...end

当只有一种选择时，使用该形式。此时的程序结构如下：

```
if expression
    commands
...
end
```



`expression`是条件表达式，`commands`是执行代码块。这是最简单的判断语句，只有一个判断语句。其中的表达式为逻辑表达式，当表达式`expression`结果为真时，执行相应的语句`commands`，否则，直接跳到`end`下一段语句。



例如：

```
>> apples=10;
>> cost=apples*25
cost =
    250
>> if apples>5 %如果购买量大于5，给予20%的价格折扣
    cost=(1-0.2)*cost;
end
>> cost
cost =
    200
```



例 用 if...end 结构求解  $\sum_{i=1}^m i > 1000$  中求最小的 m 值。

解 程序如下：

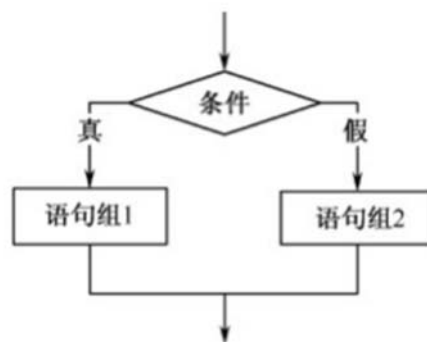
```
s=0;
for i=1:1000
    s=s+i;
    if s>1000
        break;
    end
end
m=i;
[s,m]
```



## 2. if...else...end

当程序有两个选择时，可以选择 if...else...end 结构，此时的程序结构如下：

```
if expression
    commands 1
else
    commands 2
end
```



当判断表达式 expression 为真时，执行代码块 commands1，否则执行代码块 commands2。



例 输入数n，判断其正负性。

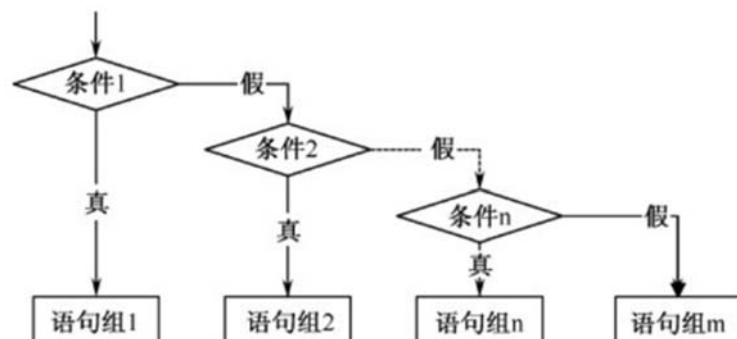
```
m=input('m= ');
if m<0
    disp('m为负数! '),
else
    disp('m为正数! '),
end
```



### 3. if...elseif...else...end

上面的两种形式中，分别包含一个选择和两个选择，当判断包含三个或多个选择时，可以采用elseif语句，其结构如下：

```
if expression1
    commands1
elseif expression2
    commands2
elseif ...
...
...
else
    commands
end
```



例 输入数n，判断其正负及奇偶性。

解 程序如下：

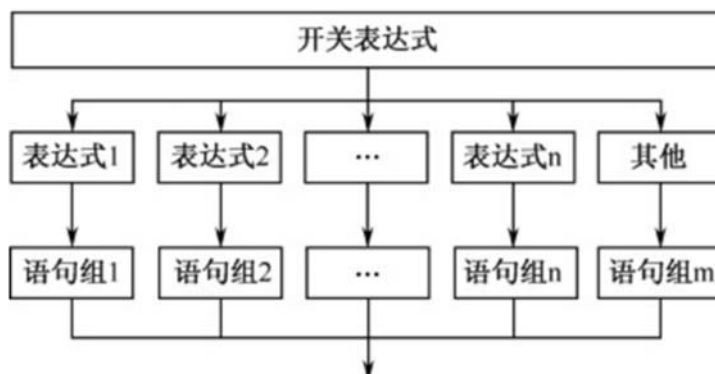
```
n=input('n=');  
if n<0  
    A='负数'  
elseif rem(n,2)==0  
    A='偶数'  
else  
    A='奇数',  
end
```



### 3.2 switch开关结构

MATLAB中的另一种分支结构为开关分支语句。开关分支语句的结构如下：

```
switch expression  
case expression_1  
    commands_1  
case expression_2  
    commands_2  
...  
otherwise  
    commands  
end
```



其中的分支语句开关表达式`expression`为一个变量，可以是数值变量或者字符串变量，如果该变量的值与某一条件`expression_n`相符，则执行相应的语句；否则执行`otherwise`后面的语句。在每一个条件中，可以包含一个条件语句，可以包含多个条件，当包含多个条件时，将条件以单元数组的形式表示。例如：



```
method = 'Bilinear';
switch lower(method)
    case {'linear', 'bilinear'}
        disp('Method is linear')
    case 'cubic'
        disp('Method is cubic')
    case 'nearest'
        disp('Method is nearest')
    otherwise
        disp('Unknown method.')
end
```





### 实例解析:

问题: 某商场对顾客所购买的商品实行打折销售, 标准如下:

$\text{price} < 200$	不打折
$200 \leq \text{price} < 500$	3%折扣
$500 \leq \text{price} < 1000$	5%折扣
$1000 \leq \text{price} < 2500$	8%折扣
$2500 \leq \text{price}$	10%折扣

输入所售商品的价格, 求其实际销售价格。

注: 获取输入数据的函数: **input()**, 小数取整函数:  
**fix()**, **ceil()**, **round()**



### 解析:

使用switch结构时, 不能直接选用price作为开关表达式, 因为price可以是小数, 其取值非有限个, 需要将其划为整数。根据题目条件, 可选用fix()函数, 进行趋零取整。



```
price = input('请输入商品价格'),
switch fix(price)
    case num2cell(0:199)
        rate = 0;
    case num2cell(200:499)
        rate = 3/100;
    case num2cell(500:999)
        rate = 5/100;
    case num2cell(1000:2499)
        rate = 8/100;
    otherwise
        rate = 10/100;
end
sale_price = price*(1-rate)
```



给出学生百分制成绩，要求转化为等级制输出。

90分及以上，等级制输出为“优秀”；80~89输出“良好”；

70~79输出“中等”；60~69输出“及格”；其它“不及格”。



```
s=input('请输入学生百分制成绩');  
switch fix(s/10)    %函数fix(), 朝零方向取整  
    case {10,9}    %{10,9}为胞元数组  
        G='优秀'  
    case 8  
        G='良好'  
    case 7  
        G='中等'  
    case 6  
        G='及格'  
    otherwise  
        G='不及格'  
end
```



某旅行团有男人，女人，小孩共计30人，在华盛顿地区一家饭店吃饭，该饭店按人数收费。男人每餐30美元、女人每餐20美元，儿童每餐10元。饭店收到500美元，男人、女人、儿童的人数共有多少种可能？



```
for man=1:30
    for woman=1:30
        for children=1:30
            if(man*30+woman*20+children*10==500)...
                &(man+woman+children==30)
                fprintf('Man: %d\n',man)
                fprintf('Woman: %d\n',woman)
                fprintf('Children: %d\n',children)
            end
        end
    end
end
```



```
j=0;
for man=1:30
    for woman=1:30
        for children=1:30
            if(man*30+woman*20+children*10==500)...
                &(man+woman+children==30)
                j=j+1;
                A(j,1)=man;
                A(j,2)=woman;
                A(j,3)=children;
            end
        end
    end
end
Man____Woman____Children=A
```



## 作业

1、必须使用for或while来解决以下问题

(1) 使用for循环遍历向量A，返回一个元素为逻辑值的新向量B。B中的true对应A中的正值，false对应A中的其他值。例如，如果A=[-300 2 5 -63 4 0 -46], B=[false true true false true true false].

(2) 使用while循环遍历向量A，返回一个新向量B。B中的true对应A中的正值，false对应A中的其他值

(3) 使用for循环遍历逻辑数组N，返回一个新向量M。M中的2对应N中的true，-1（不是逻辑值）对应N中的false。例如，如果N=[true false false true true false true], M=[2 -1 -1 2 2 -1 2]。

(4) 使用while循环遍历数组Z。在遇到一个大于50的数字之前，用3替代Z中的元素，例如，如果Z=[4 3 2 5 7 9 0 64 34 43]，运行代码后，Z=[3 3 3 3 3 3 3 34 43]

