

MATLAB程序结构控制

1 M文件

M文件是最常见的MATLAB代码文件，它可以是：命令文件（脚本文件，Script File）函数文件（Function File）

区别在于：①命令文件没有输入，没有返回；②命令文件可以对工作空间的变量操作，结果返回工作空间，而函数文件中的变量为局部变量，函数执行完毕，变量被清除；③命令文件可以直接运行，函数文件需要调用的形式运行（除了特殊的函数文件之外）。

任务 1

现在就试试编写和保存自己的 MATLAB 程序吧。

依次点击“新建 > 脚本”，或者快捷键Ctrl+N，打开一个空白脚本。现在可以输入命令，并保存，请点击“运行（Run）”按钮，所有命令会按顺序进行计算。注意工作区和命令行窗口，分别保存了运算的结果。

```
a=5
```

任务2

接下来，我们来介绍如何规范地编写一个函数文件。

通常，函数文件由函数声明行、H1行、在线帮助文本区、编写和修改记录、函数主体等几个部分组成。格式如下：

function 输出形参表 = 函数名(输入形参表) %%注释说明 函数体 end

```
function output=NonNegative(input)
%% This function makes the negative inputs to be zero, but doesn't change positive inputs.
    if(input<0)
        output=0;
    else
        output=input;
    end

end
```

将上述代码新建一个脚本文件，并保存为同名M文件（即NonNegative.m）。现在，可以在命令行窗口调用该函数，试试使用不同的输入观察结果吧。

任务3

注释部分可以提供help命令的功能。

```
help NonNegative
```

2 函数与函数参数

2.1 函数参数可调标识变量

任务 1

在matlab中，**nargin** (numbers of input arguments)和**nargout** (numbers of output arguments)为输入输出参数个数，调用它们，可以很方便地重载函数。

```
function re = add_multi(a, b, c, d)

    if nargin == 4
        re = a + b + c + d;
    elseif nargin == 3
        re = a + b + c;
    elseif nargin == 2
        re = a + b;
    else
        error('输入参数个数错误');
    end
```

add_multi函数现在可以输入2或3或4个变量，得到它们的和。

任务 2

varargin和**varargout**可以代表未知的输入输出变量。

```
function re = add_var(a, b, varargin)

    narginchk(2, 4);

    if nargin == 2
        re = a + b;
    elseif nargin == 3
        c = varargin{1};
        re = a + b + c;
    elseif nargin == 4
        c = varargin{1};
        d = varargin{2};
        re = a + b + c + d;
    else
        error('wrong');
    end
```

附加资源：Matlab: nargin,nargout,varargout以及varargin的用法 -- http://blog.sina.com.cn/s/blog_67f37e760102v38a.html

2.2 全局变量

任务 1

使用global声明，可以提供不同的M文件访问同一个变量。尝试在非Negative中填加以下两行代码：

```
global last_input;  
last_input=input;
```

在命令行窗口输入 global last_input; 随后可以调用last_input的值。

逻辑运算符: <https://www.mathworks.com/help/matlab/logical-operations.html>

2.3 程序调试

断点和单步； 根据错误提示，初步确定错误内容；

```
function rate = RateofGeomean(X)  
%%for each xi, return xi/Geomean(X)  
[m,n]=size(X);  
  
cumprod=1;  
  
for i=1:m  
    for j=1:n  
        cumprod=cumprod*X(i,j);  
    end  
end  
  
geomean=cumprod^(1/(m*n));  
  
for i=1:m  
    for j=1:n  
        rate(i,j)=X(i,j)/geomean;  
    end  
end  
  
end
```

任务 1

尝试使用断点，发现geomean的错误，并在合适的地方，填加error语句后，再调用函数。

```
X=[1 2 3  
9 0 2  
1 7 8];  
  
>> RateofGeomean(X)  
  
ans =  
  
    Inf    Inf    Inf  
    Inf   NaN    Inf  
    Inf    Inf    Inf
```

```
Y=[3 2 1
6 5 4
9 8 7];
```

任务2

程序控制结构——循环

怎样避免使用循环和提高循环效率？有些可以通过MATLAB的矢量化语言，通过矩阵或者向量操作完成；有些可以通过MATLAB提供的一些特殊操作工具箱函数完成；预分配的使用，会大大增加循环效率

```
function rate = AdvanceRateofGeomean(X)
%%for each xi, return xi/Geomean(X)

    if(~all(X(:)))
        error('0s in X, can't calculate the geomean!!!')
    end
    nums=numel(X);
    cumprod=prod(X(:));
    geomean=cumprod^(1/nums);
    rate=X./geomean;

end
```

使用下列语句，测试优化的效果：

```
Y=[3 2 1
6 5 4
9 8 7];

iter=100000;

tic;
for i=1:iter
    RateofGeomean(Y);
end
t1=toc;

tic;
for i=1:iter
    AdvanceRateofGeomean(Y);
end
t2=toc;
disp(t1/t2)
```

3 程序控制结构

3.1 分支

任务 1

分支结构，又称之为选择结构，包括if分支，switch分支和try分支；

%%单分支if语句 语法：

```
%if 条件
%      语句组
%end
A = input('input A');
if A > 10
    disp(A);
end
```

%%双分支if语句 语法：

```
%if 条件
%      语句组1
%else
%      语句组2
%end
x = input('x');
if x > 10
    y = log(x);
else
    y = cos(x);
end
```

%%多分支if语句 语法：

```
%if 条件1
%      语句组1
%elseif 条件2
%      语句组2
%elseif 条件m
%      语句组m
%else
%      语句组n
%end
c = input('input a character', 's');
if c >= 'A' & c <= 'Z'
    disp(char(abs(c) + abs('a') - abs('A')));
elseif c >= 'a' & c <= 'z'
    disp(char(abs(c) - abs('a') + abs('A')));
elseif c >= '0' & c <= '9'
    disp(c);
end
```

%%switch分支语句 语法：

```
%switch 表达式
%    case 表达式1
%        语句组1
%    case 表达式2
```

```

%           语句组2
%   case 表达式m
%           语句组m
%otherwise
%           语句组n
%end
price = input('input price');
switch fix(price/100)
    case {0, 1}
        rate = 0;
    case {2, 3, 4}
        rate = 0.03;
    case {5, 6, 7, 8, 9}
        rate = 0.05;
    otherwise
        rate = 0.1;
end

```

附加资源： if 语句: <http://www.mathworks.com/help/matlab/ref/if.html> switch 语句: <http://www.mathworks.com/help/matlab/ref/switch.html>

3.2 循环

包括for循环和while循环。

```

%% for循环: 语法 (常用的一种形式)
% for 循环变量 = 表达式1:表达式2:表达式3
%     循环体
% end
% 注意循环变量自动增加, 在循环体内对循环变量的赋值操作会带来不可预料的程序执行
A = 1:100;
sumA = 0;
for k = 1:100
    sumA = sumA + A(k);
end

%% while循环 语法:
% while 条件
%     循环体
% end
% while循环
while 1
    c = input('input a character', 's');
    if isempty(abs(c))
        break;
    end
end
end

```

附加资源： for 循环: <http://www.mathworks.com/help/matlab/ref/for.html> while 循环: <http://www.mathworks.com/help/matlab/ref/while.html>

4 结构体

直接使用`s=struct`就可以创建一个空结构体`s`。

```
struct('field',value);
```

可以创建字段为`field`，其值为`value`。当`value`是一个有`n`个元素的元胞数组时，创建的结构体长度也为`n`，每个结构体的`field`字段有元胞数组的一项。

```
%%该函数将生成一个具有指定字段名和相应数据的结构数组，其包含的数据values1、values2等必须为具有相同维数的数据，数据的存放位置域其他结构位置——对应的。对于struct的赋值用到了元胞数组。数组values1、values2等可以是元胞数组、标量元胞单元或者单个数值。每个values的数据被赋值给相应的field字段。  
sturct('field1',values1,'field2',values2,...);//注意引号
```

任务 1

当`valuesx`为元胞数组的时候，生成的结构数组的维数与元胞数组的维数相同。而在数据中不包含元胞的时候，得到的结构数组的维数是`1×1`的。观察下列结构体的维数：

```
s = struct('type',{'big','little'},'color',{'blue','red'},'x',{3,4})
```

任务2

对于产生的结构体数组`s`，使用`s(i)`访问其中的第`i`个结构体。结构体数组也是从1开始的，长度为2的结构体`s`的两个元素是`s(1)`和`s(2)`

```
>> s(1)  
  
    type: 'big'  
   color: 'blue'  
        x: 3  
  
>> s(2)  
  
    type: 'little'  
   color: 'red'  
        x: 4
```

任务3

直接用英文句号调用结构体字段元素。

```
>> s(2).type  
'little'
```

任务

给`s`的第一个结构体的字段`color`赋值为`'black'`。