



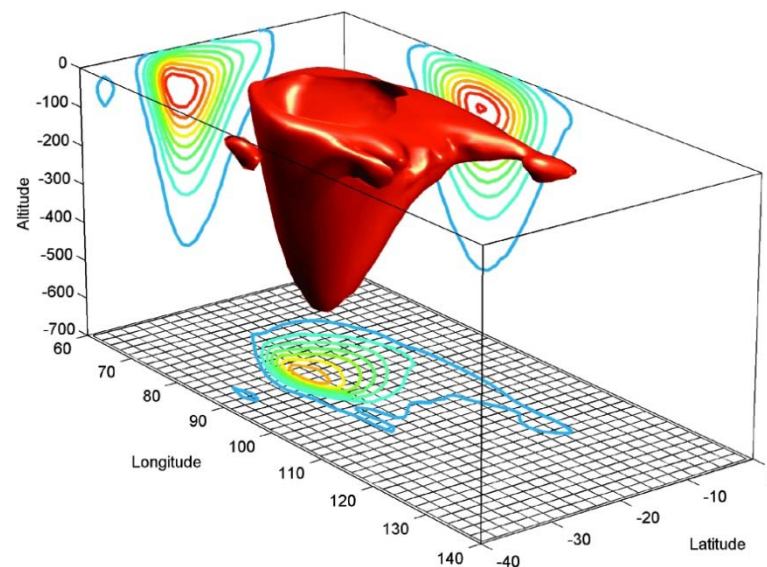
Matlab语言程序设计

■ 第4讲

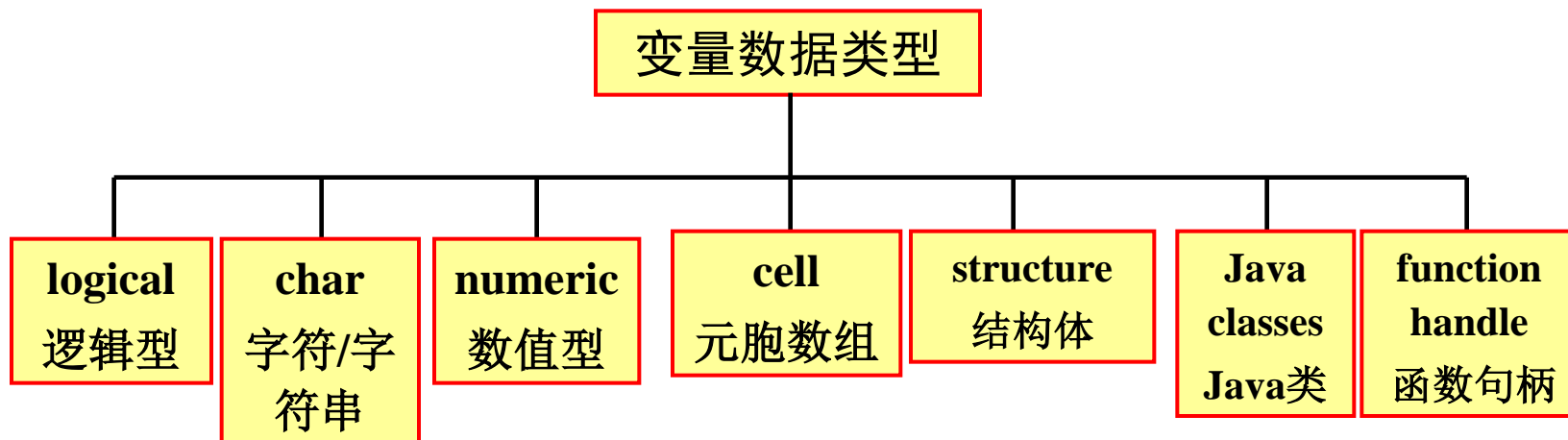
Matlab 数据类型

◆ 主要内容

- 数值类型
- 常量与变量
- 逻辑类型
- 逻辑运算与关系运算
- 字符和字符串
- 元胞数组
- 结构体



MATLAB基本数据类型



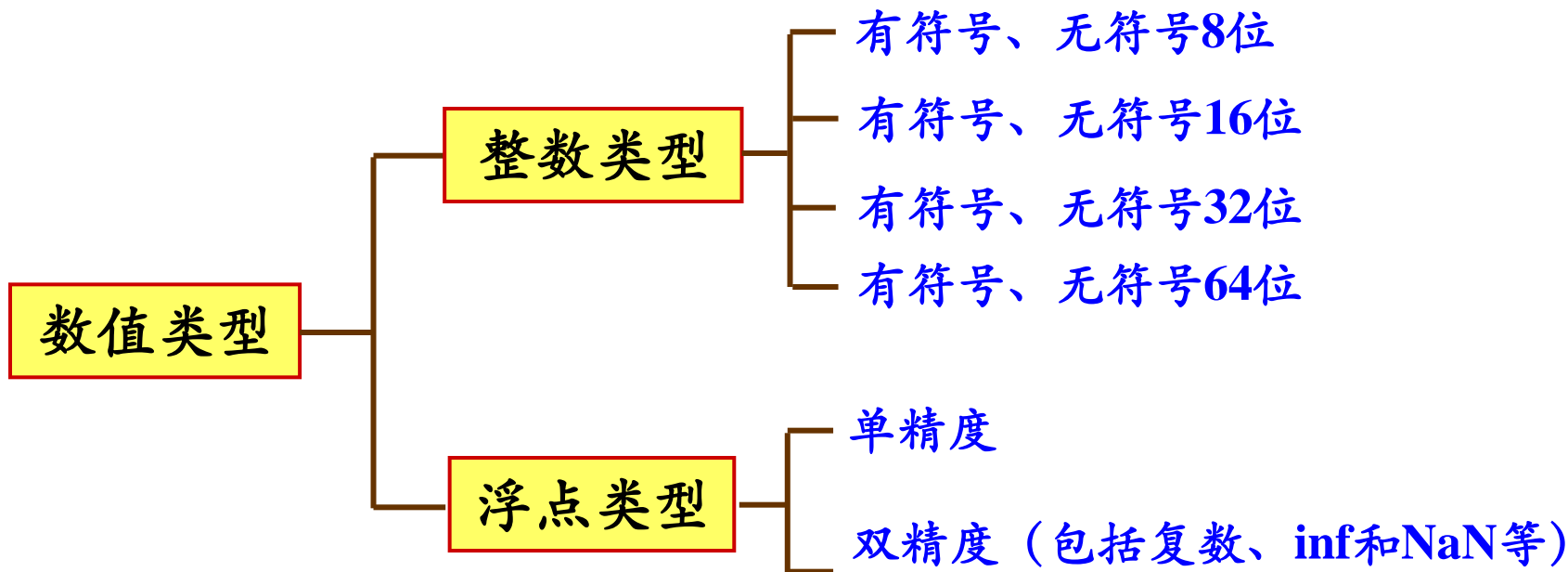
所有类型数据都被存储在**矩阵或数组**中进行操作

MATLAB的**基本数据类型**是**双精度数据类型**和**字符类型**

MATLAB的**不同类型的数据**占用的**内存空间**不同

1、数值类型

✍ 数值类型有符号整数类型、无符号整数类型、单精度浮点类型和双精度浮点类型



1、数值类型

- ✓ 相同的整型数据之间的运算，结果为同类型的整数。
不同的整数型数据之间不能进行运算。
- ✓ 整型数据可以和双精度标量进行运算，结果为整数。
在运算过程中保持浮点运算精度，最后将结果转换成整型。不能与非标量的双精度数组进行运算。

浮点数与其它类型数据运算表

operand	single	double	int/uint	char	logical
single	single	single	X	single	single
double	single	double	int/uint	double	double



2、常量与变量

■ 数值的表示

Matlab的数只采用习惯的十进制表示，可以带小数点和负号；其缺省的数据类型为双精度浮点型（double）

例如：3 -10 0.001 1.3e10 1.256e-6

■ 变量命名规则

- 变量名、函数名对字母的大小写是敏感的。如 **myVar** 与 **myvar** 表示两个不同的变量
- 变量名第一个字母必须是英文字母
- 变量名可以包含英文字母、下划线和数字
- 变量名不能包含空格、标点
- 变量名最多可包含63个字符（6.5及以后的版本）

2、常量与变量

■ Matlab预定义的变量

变量名	意义
ans	最近的计算结果的变量名
eps	MATLAB定义的正的极小值=2.2204e-16
pi	圆周率 π
inf	∞ 值，无限大
i或j	虚数单元， $\text{sqrt}(-1)$
NaN	非数，0/0、 ∞/∞

[[说明]]

- 每当MATLAB启动完成，这些变量就被产生。
- MATLAB中，被0除不会引起程序中断，给出报警的同时用inf或NaN给出结果。
- 用户只能临时覆盖这些预定义变量的值，Clear或重启MATLAB可恢复其值。



3、逻辑类型

Matlab用“0”和“1”分别代表“假”和“真”。

任何数值都可以参与逻辑运算

- 非零值看作逻辑真
- 零值看作逻辑假

产生逻辑类型数据的方法：

- 由数值类型转换(logical)
- 函数创建(true、false)
- 逻辑运算

4、逻辑运算与关系运算

Matlab提供了6种逻辑运算符：

&（与）、|（或）、~（非）

&&（先决与）、||（先决或）

逻辑运算符的运算法则：

- 1、在逻辑运算中，确认非零元素为真（1），零元素为假（0）。
- 2、当两个维数相等的矩阵进行比较时，其相应位置的元素按标量关系进行比较，并给出结果，形成一个维数与原来相同的0、1矩阵；
- 3、当一个标量与一个矩阵比较时，该标量与矩阵的各元素进行比较，结果形成一个与矩阵维数相等的0、1矩阵；
- 4、算术运算优先级最高，逻辑运算优先级最低。

4、逻辑运算与关系运算

Matlab提供了6种关系运算符：

<、>、<=、>=、==、~=（不等于）

关系运算符的运算法则：

- 1、当两个标量进行比较时，直接比较两数大小。若关系成立，结果为1，否则为0。
- 2、当两个维数相等的矩阵进行比较时，其相应位置的元素按标量关系进行比较，并给出结果，形成一个维数与原来相同的0、1矩阵。
- 3、当一个标量与一个矩阵比较时，该标量与矩阵的各元素进行比较，结果形成一个与矩阵维数相等的0、1矩阵。

4、逻辑运算与关系运算(续)

【例】建立5阶方阵A，判断其元素能否被3整除。

$A = [24, 35, 13, 22, 63; 23, 39, 47, 80, 80; \dots$

$90, 41, 80, 29, 10; 45, 57, 85, 62, 21; 37, 19, 31, 88, 76]$

A =

24 35 13 22 63

23 39 47 80 80

90 41 80 29 10

45 57 85 62 21

37 19 31 88 76

$P = \text{rem}(A, 3) == 0$ %被3除，求余

P =

1 0 0 0 1

0 1 0 0 0

1 0 0 0 0

1 1 0 0 1

0 0 0 0 0

4、逻辑运算与关系运算(续)

【例】建立矩阵A，找出在[10, 20]区间的元素的位置。

```
A = [4,15,-45,10,6;56,0,17,-45,0];
```

```
find(A>=10 & A<=20) %找到非零元素的位置
```

```
A =
```

```
4    15   -45    10     6
```

```
56     0    17   -45     0
```

```
ans =
```

```
3
```

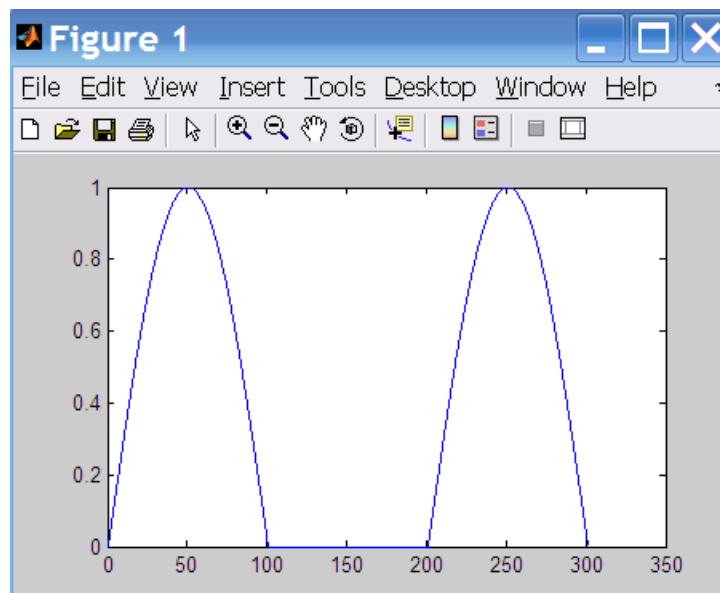
```
6
```

```
7
```

4、逻辑运算与关系运算(续)

【例】在 $[0, 3\pi]$ 区间，求 $y = \sin x$ 的值。要求
消去负半波，即 $(\pi, 2\pi)$ 区间内的函数值置零。

```
x = 0:pi/100:3*pi;  
y = sin(x);  
y1 = (y>=0).*y; %消去负半波
```



4、逻辑运算与关系运算(续)

优先级别

运算符

1	括号 ()	成员符 .					
2	转置 .'	共轭转置 '	数组幂 .^	矩阵幂 ^			
3	代数正 +	代数负 -	逻辑非 ~				
4	数组乘 .*	数组除 ./	数组除 ./	矩阵乘 *	矩阵左除 \	矩阵右除 /	
5	加 +	减 -					
6	冒号 :						
7	小于 <	大于 >	等于 ==	不小于 >=	不大于 <=	不等于 ~=	
8	逻辑与 &						
9	逻辑非						
10	先决与 &&						
11	先决非						
12	赋值 =						



5、字符和字符串

- 字符 (Characters) 可以构成一个字符串 (Strings)，或字符数组(character array)。
- 一个字符串是被视为一个行向量 (row vector) 。
- 字符串中的每一个字符 (含空格)，以其 ASCII 码的形式存放于行向量中，是该字符串变量的一个元素 (element) 。

5、字符和字符串

- Matlab 用「单引号」来界定一个字符串。
- 方括号“[]”直接连接多个字符串变量，得到一个新字符串变量。

【例3-2】创建字符串变量

% 建立字符串变量 str1

```
>>str1 = 'I like MATLAB,';
```

% 建立字符串变量str2

```
>>str2 = 'JavaScript, and Perl!';
```

% 直接连接str1及str2，以建立str3

```
>>str3 = [str1 str2]
```

```
str3 =
```

```
I like MATLAB, JavaScript, and Perl!
```




5、字符和字符串

- 如要输入的字符串中有单引号，则由两个连续的单引号来表示。

【例3-2】含单引号字符串的输入

```
>>str1 = 'I've got a date!';
```

```
>>['xlswrite('data',num2str(i),'.xlsx',',','data',num2str(i),') ']
```

5、字符和字符串

■ 二维字符数组 (Two Dimensional Character Arrays)

【例3-5】：多行字符串变量

```
>>str1= ['ee '; 'cs '; 'econ']
```

使用“,”间隔不同字符串，相当于扩展字符串成为更长的字符串向量。

使用“;”间隔不同字符串之间的，相当于扩展字符串成为二维或者多维的数组。要求不同行上的字符串必须具有同样的长度。

5、字符和字符串

■ 使用char函数创建二维字符数组

% 注意空格及「,」的使用

```
>>str1 = char('ee', 'cs', 'econ')
```

🔔 不同行上的字符串长度可以不一样

```
>>str2=char('这', '字符', '串数组', ',', '由5行组成')  
str2=
```

```
    这  
    字  
    符  
    串  
    数  
    组
```

```
    由5行组成
```

5、字符和字符串

■ 转换函数char和double

char函数

按照ASCII码表将数字映射成字符。控制字符或通信专用字符(0 ~127)。一次不能超过65535 。

例：

char(32:127)

🔔 ASCII码表第32至127个的字符

double函数

将参数y转为双精度浮点类型，如果y是字符，将返回字符的ASCII码值。

例：

double('21 22 23')

🔔 结果是数值矩阵，元素个数为8

5、字符和字符串

■ 转换函数num2str和str2num

str2num函数

将字符串转换成数字

>>str2num('a') ☒

>>str2num('21') ☒

🔔 字符串必须是数值字符

str2num('21 22 23')与double('21 22 23')的区别

■ 注意空格

>>str2num('1+2i')

>>str2num('1 + 2i')

>>str2num('1 +2i')

🔔 用str2double避免这种情况

5、字符和字符串

■ 转换函数num2str和str2num

num2str函数

将数字转换成字符串

例： num2str(43)

🔔 结果是'43'，有两个字符

eval('y=sin(1)')等同于执行命令 y=sin(1)

```
>>xlswrite('data1.xlsx',data1)
```

.....

```
>>xlswrite('data100.xlsx',data100)
```

```
>>eval(['xlswrite(''data'',num2str(i),'.xlsx',...  
'','data',num2str(i),' '])])
```

6、元胞数组 (cell array)

元胞数组 (单元数组) 的概念

- 特殊的数据类型，在一个数组中存放各种不同类型的数据
- 每个单元相当于一个“盒子”
- “盒子”可存储各种不同类型的MATLAB数据

元胞数组例子：

Cell 1,1	Cell 1,2	Cell 1,3
0	Mainland Beijing	[1 2 3 4] test

6.1 元胞数组的创建

■ 创建方法

■ 赋值语句

■ cell函数

■ 赋值语句创建元胞数组

■ 元胞索引(cell indexing)方式

格式: $a(1,2)=\{\dots\dots\dots\}$

■ 元胞内容索引(content indexing)方式

格式: $a\{1,2\}=[\dots]$ or $'\dots'$

■ 直接用大括号一次把所有元素括起来

$b = \{'James Bond', [1\ 2;3\ 4;5\ 6]; pi, ones(5)\}$

6.1 元胞数组的创建（续）

■ Cell indexing方式创建元胞数组

```
>> a(1,1) = {[1 4 3; 0 5 8; 7 2 9]};
```

```
>> a(1,2) = {'Anne Smith'};
```

```
>> a(2,1) = {3+7i};
```

```
>> a(2,2) = {-pi:pi/10:pi};
```

■ “{ }”表示空元胞数组

■ Content indexing方式创建元胞数组

```
>> b{1,1} = 'James Bond';
```

```
>> b{1,2} = [1 2;3 4;5 6];
```

```
>> b{2,1} = pi;
```

```
>> b{2,2} = zeros(5);
```

6.1 元胞数组的创建（续）

■ cell函数创建元胞数组

```
>>b = cell(2, 3)
```

```
b =
```

```
    []    []    []  
    []    []    []
```

```
>> whos b
```

Name	Size	Bytes	Class
b	2x3	24	cell array

注意：每个cell占有4个字节的空间

```
>>b(1,3) = {1:3}; % or b{1,3} = 1:3;
```

6.1 元胞数组的创建（续）

■ 用[] 包括元胞数组元素

```
>>c=[a b]
```

```
>>c=[a; b]
```

将数组的每一个元素用{}括起来，然后再用数组创建的符号[]将数组的元素括起来构成一个元胞数组。

例：

```
>> B=[{zeros(2,2,2)},{ 'Hello'};{17.35},{1:100}]
```

6.3 元胞数组的索引

元胞数组的索引有元胞索引 (cell indexing) 和内容索引 (content indexing) 两种。

- `()` 用于元胞索引 (cell indexing) ，获得的是一个元胞数组。
- `{}` 用于内容索引 (content indexing) ，获得的是元胞数组元素的内容。

```
>> A=[{zeros(2,2,2)},{'Hello'};  
      {17.35},{1:100}]
```

```
>> A(1,2)
```

```
>> A{1,2}
```

6.3 元胞数组的索引

元胞索引和内容索引获得的数据类型不同。

```
>> A=[{zeros(2,2,2)},{ 'Hello'};
```

```
      {17.35},{1:100}]
```

```
>>class(A(1,2))
```

```
ans =
```

cell

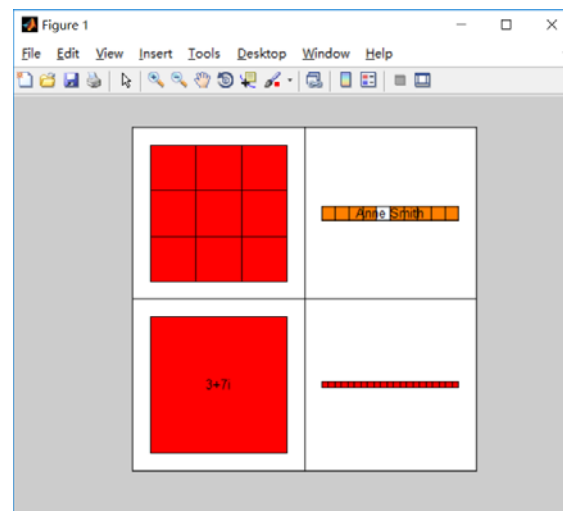
```
>>class(A{1,2})
```

```
ans =
```

char

%图形方式显示元胞数组的结构

```
>>cellplot(a)
```



6.3 元胞数组的索引

■ 元胞数组元胞内的数据的索引

```
>> b{1,1} = 'James Bond' ;
```

```
>> b{1,2} = [1 2;3 4;5 6];
```

```
>> b{2,1} = pi;
```

```
>> b{2,2} = zeros(5);
```

```
>> e = b{1,1}(1,7)
```

读取元胞数组元胞内的数据：

元胞数组内容索引，再加上一般数组的索引，
即： $a\{x,y\}(r,c)$

6.3 元胞数组的索引

- 例：读取元胞数组a第1行的所有元胞。

```
>>f=a(1,:)
```

```
f =
```

```
[3x3 double] 'Anne Smith'
```

- 例：删除元胞数组a第1行的所有元胞。

```
>>a(1,:) = []
```

```
a =
```

```
[3.0000 + 7.0000i] [1x21 double]
```

注意: 这里使用的是元胞索引, 而不是内容索引。



7.1 结构与结构数组的概念

■ 结构(structure)

- 由字段（或域，fields）组成
- 每个字段可以是任一种Matlab数据类型的数据或变量
- 与C语言的结构类型相似

■ 结构数组(structure array)

- 多个结构构成结构数组(structure array)
- 结构数组的元素就是一个结构

7.2 结构数组

■ 直接键入创建结构体

创建一个关于学生信息的结构数组，每个结构包含学生姓名(name)、学号(id)、成绩(scores)。

```
>>clear student           % 清除 student 变量
>>student.name = '张三';  % 加入 name 字段
>>student.id = 'mr871912'; % 加入 id 字段
>>student.scores = [58, 75, 62]; % 加入 scores 字段
>>student                 % 显示结构变量的数据
student =
    name: '张三'
    id: 'mr871912'
   scores: [58,75,62]
```

此时，Matlab视 student 为一个1x1结构数组

7.2 结构数组变量的创建（续）

建立第二个学生的信息

```
>>student(2).name = '张宁';           % 加入 name 字段
>>student(2).id = 'mr871913';        % 加入 id 字段
>>student(2).scores=[68, 85, 92];    % 加入 scores 字段
>>student                             % 显示变量信息
```

student =

1x2 struct array with fields:

name

id

scores



7.2 结构数组变量的创建（续）

- 此时，`student`变量已成为一个1x2的结构数组
- 只是简单输入变量名`student`的话，Matlab不再显示其所有的数据信息，只是给出结构
- 如何显示结构数组变量`student`的数据？
 - `student(1)`
 - 或
 - `student(1).name`

7.2 结构数组变量的创建（续）

■ 使用struct函数创建结构

格式

```
struct_name = struct(field1, value1, field2,  
                    value2,...)
```

```
struct_name = struct(field1, { }, field2, { },...)
```

field1、field2、...是结构的字段名，

value1、value2、...则是相应字段所包含的数据。

7.2 结构数组变量的创建（续）

使用struct创建结构数组变量（一次建立多个元素）

```
>>clear student
```

```
>>student=struct('name','张三','scores',[50 60]);
```

```
>> student(2)=struct('name','李四','scores',[60 70]);
```

```
>> student(1), student(2) %显示student(1),student(2)
```

```
ans =
```

```
    name: '张三'
```

```
    scores: [50 60]
```

```
ans =
```

```
    name: '李四'
```

```
    scores: [60 70]
```

7.2 结构数组变量的创建（续）

使用struct创建结构数组变量（一次建立多个元素）

```
>>clear student
```

```
>>student = struct('name',{'张三','李四'}, 'scores',  
    {[50 60],[60 70]});
```

```
>> student(1), student(2) %显示student(1), ...
```

```
ans =
```

```
    name: '张三'
```

```
    scores: [50 60]
```

```
ans =
```

```
    name: '李四'
```

```
    scores: [60 70]
```

7.2 结构数组变量的创建（续）

■ 结构数组嵌套（nested）

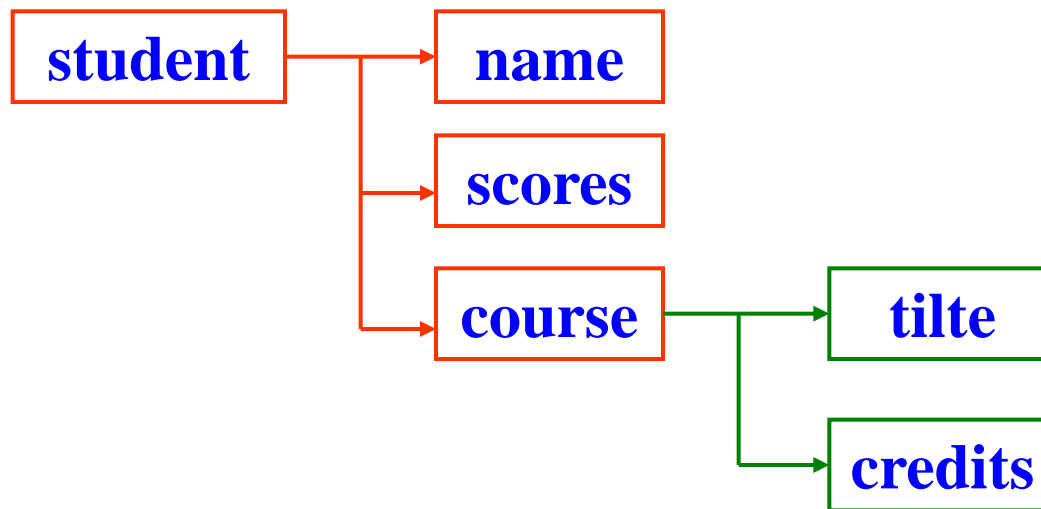
■ 结构数组中结构的字段还可以是结构。

```
student = struct('name', {'张三', '李四'}, 'scores',  
    ... {[55 60], [65 80]});  
student(2).course(1).title = 'Web Programming';  
student(2).course(1).credits = 2;  
student(2).course(2).title = 'Numerical Method';  
student(2).course(2).credits = 3;  
student(2).course  
ans =  
1x2 struct array with fields:  
    title  
    credits
```

7.2 结构数组变量的创建（续）

■ 结构数组嵌套（nested）

student结构数组变量的结构层次：



7.3 结构数组的索引

■ 格式:

使用结构数组的名称和字段的名称以及操作符“.”完成相应的操作，如：

struct_name.field_name

```
>>student_name=student(1).name
```

```
student_name=
```

```
张三
```

或者struct-name.(expression)

expression为动态字段

7.3 结构数组的索引

■ 动态字段:

使用字符变量索引不同的字段，如：

`struct_name.(expression)`

`expression` 为字符变量

例如： `>>student(1).(a)`

`>>a='name'`

`>>student(1).(a)`

等同于 `student(1).name`

`>>a='score'`

`>>student(1).(a)`

等同于 `student(1).score`

7.3 访问结构数组变量的内容（续）

■ 对结构数据的计算

例如：

```
>> student=struct('name',{'Deni','Sherry'},'age',{22,24},...  
'grade',{2,3},'score',{rand(3)*10,randn(3)*10});
```

■ 对结构数据的索引

```
>> student(2).score
```

```
>> student(2).score(1,:)
```

 与矩阵索引相同

■ 对结构数据的计算

```
>> mean(student(1).score)
```

```
>> mean([student.score])
```

 使用[]索引所有数据