Παράδειγμα για διαχείριση strings

Ας υποθέσουμε ότι έχουμε ένα string που αποτελείται από μια γωνία ακολουθούμενη από ένα γράμμα ή d για μοίρες ή r για ακτίνια. Για παράδειγμα μπορεί να είναι ένα string που δηλώνεται από τον χρήστη

```
degrad = input( ' Enter angle and d/r: ', ' s ');
Enter angle and d/r: 54r
```

Πως μπορούμε να ξεχωρίσουμε το string στην γωνία και στον χαρακτήρα και μετά να υπολογίσουμε το ημίτονο της γωνίας με την χρήση ή της εντολής sin ή sind, αναλόγως (sin για ακτίνια και sind για μοίρες)

Πρώτα απ' όλα μπορούμε να ξεχωρίσουμε το string σε δυο τμήματα:

```
>> angle = degrad(1:end-1)
angle =
54

>> dorr = degrad(end)
dorr =
r
```

Μετά με ένα if-else μπορούμε να διαλέξουμε αν θα χρησιμοποιήσουμε την sin ή την sind συνάρτηση με βάση την τιμή της συνάρτησης dorr. Ας υποθέσουμε ότι η τιμή της είναι 'r' οπότε θέλουμε να χρησιμοποιή-σουμε την συνάρτηση sin.

- Η μεταβλητή angle είναι ένα string οπότε το ακόλουθο δεν θα δουλέψει

```
>> sin(angle)

??? Function 'sin' is not defined for values

of class 'char'.

>>
```

Αντί αυτού μπορούμε να χρησιμοποιήσουμε την str2num για να μετατρέψουμε το string σε αριθμό ή εναλλακτικά να δημιουργήσουμε ένα string 'sin(54)' και να το περάσουμε στην eval συνάρτηση

```
>> eval( [ ' sin( ' angle ' ) ' ] )

ans =

-0.5588

---------------------------------

>> sin(str2num(angle))

ans =

-0.5588
```

script M-file

```
degrad = input('Enter angle and d/r: ', 's');
angle = degrad(1:end-1);
dorr = degrad(end);
% Έλεγχος για επιβεβεβαίωση ότι ο χρήστης εισάγει 'd' ή 'r'
while dorr ~= ' d ' && dorr ~= 'r'
    disp( ' Error! Enter d or r with the angle. ' )
    degrad = input(' Enter angle and d/r: ', 's');
    angle = degrad(1:end-1);
    dorr = degrad(end);
end
anglenum = str2num(angle);
fprintf('The sine of %.1f ', anglenum)
if dorr == 'd'
    fprintf('degrees is %.3f.\n', sind(anglenum))
else
    fprintf('radians is %.3f.\n', sin(anglenum))
end
```

Δομές (structures) και κελιά (cells) στο Matlab

Οι δομές στο Matlab μας επιτρέπουν να βάλουμε διαφόρων τύπων δεδομένα στα πεδία των δομών.

```
>>foititis.onoma =  'Panos Panoy';

>>foititis.at = 'PANPAN003';

>>foititis.vathmoi = [36 49 74];
```

Για να δούμε όλη την δομή πατάμε

```
>>foititis
```

Αν θέλουμε να δούμε συγκεκριμένα στοιχεία πχ στο 2ο στοιχείο στο πεδίο vathmoi πατάμε

```
>>foititis.vathmoi(2)
```

Σημειώστε την τελεία όταν δηλώνουμε το πεδίο και θέλουμε να επαναφέρουμε.

Για να προσθέσουμε ένα δεύτερο στοιχείο στην δομή foititis πατάμε

```
>>student(2).onoma = ’Nikos Nikoy’

>>student(2).at = ’NIKNIK007’

>>student(2).vathmoi = [49 98]
```

Το αρχικό πεδίο foititis έρχεται με πατώντας foititis(1). Πληροφορίες για την δομή παίρνουμε με την εντολή **fieldnames(foititis)**

Μπορούμε να δημιουργήσουμε μια δομή mathima η οποία να αντιστοιχεί σε μαθήματα που παίρνουν οι φοιτητές στην οποία το ένα πεδίο είναι το όνομα του μαθήματος και στο άλλο οι φοιτητές.

```
>>mathima.onoma = ' MAS191a ';

>>mathima.class = foititis;

>>mathima
```

Ένα δεύτερο μάθημα για ένα διαφορετικό τμήμα φοιτητών δηλώνεται με

```
>>mathima(2).onoma = ' MAS191b '

>>mathima(2).class =  …
```

Για να δούμε όλα τα μαθήματα

```
>>mathima(1:2).onoma
```

Για να δούμε τους φοιτητές ενός μαθήματος πατάμε

```
>>mathima(1).class(1:2).onoma
```

- Με την εντολή rmfield μπορούμε να σβήσουμε πεδία από μια δομή

Ένθετες Δομές

Μια ένθετη δομή είναι μια δομή της οποίας τουλάχιστον ένα πεδίο είναι η ίδια μια άλλη δομή. Για παράδειγμα μια δομή ενός ευθύγραμμου τμήματος μπορεί να αποτελείται από τα σημεία x y που παριστάνουν τα σημεία που ορίζουν το ευθύγραμμο τμήμα.

Σημειο 1		Σημείο 2	
x	y	x	y
2	4	1	6

Ένας τρόπος για να ορίσουμε τα σημεία αυτά στο Matlab είναι

```
>> lineseg = struct( ' endpoint1 ' , struct(' x ',2,' y ',4 ), . . .
' endpoint2 ', struct(' x ',1,' y ',6) )
```

Ένας εναλλακτικός τρόπος είναι να δημιουργήσουμε την ένθετη δομή ορίζοντας ένα πεδίο την φορά

```
>>lineseg.endpoint1.x = 2;

>>lineseg.endpoint1.y = 4;

>>lineseg.endpoint2.x = 1;

>>lineseg.endpoint2.y = 6;

>> lineseg.endpoint1

ans 1⁄4

x: 2

y: 4
```

Διανύσματα από ένθετες δομές

Συνδυάζοντας διανύσματα και ένθετες δομές μπορούμε να δημιουργήσουμε διανύσματα από δομές όπου ορισμένα πεδία είναι τα ίδια δομές.

- Για παράδειγμα, ένα εργοστάσιο κατασκευάζει κυλίνδρους από διαφορετικά υλικά για βιομηχανική χρήση. Πληροφορίες σχετικά με τους κυλίνδρους αποθηκεύονται σε μια δομή δεδομένων σε ένα πρόγραμμα

  Η μεταβλητή cyls είναι ένα διάνυσμα από δομές που κάθε δομή έχει πεδία code, dimensions, και weight. Οι διαστάσεις είναι δομές που αποτελούνται από πεδία rad και height για την ακτίνα και το μήκος του κάθε κυλίνδρου.

`cyls`

		dimensions		
	code	rad	height	weight
1	x	3	6	7
2	a	4	2	5
3	c	3	6	9

Ένας τρόπος να εισάγουμε τα δεδομένα είναι

```
>> cyls(3) = struct('code', 'c', 'dimensions',. . .
struct('rad', 3, 'height', 6), 'weight', 9);
>> cyls(1) = struct('code', 'x', 'dimensions',. . .
struct('rad', 3, 'height', 6), 'weight', 7);
>> cyls(2) = struct('code', 'a', 'dimensions',. . .
struct('rad', 4, 'height', 2), 'weight', 5);
```

==========εναλλακτικά==========

```
>> cyls(3).code = 'c';
>> cyls(3).dimensions.rad = 3;
>> cyls(3).dimensions.height =  6;
>> cyls(3).weight = 9;
>> cyls(1).code = 'x';
>> cyls(1).dimensions.rad = 3;
>> cyls(1).dimensions.height = 6;
>> cyls(1).weight = 7;
>> cyls(2).code = 'a';
>> cyls(2).dimensions.rad = 4;
>> cyls(2).dimensions.height = 2;
>> cyls(2).weight =  5;
```

Για τους κυλίνδρους ένας επιθυμητός υπολογισμός μπορεί να είναι ο όγκος τους π * r^2 * h. Η παρακάτω συνάρτηση printcylvols τυπώνει τον όγκο του κάθε κυλίνδρου και τον κωδικό για λόγους ταυτοποίησης. Ο όγκος υπολογίζεται από μια subfunction

```
function printcylvols(cyls)

for i = 1:length(cyls)

    vol = cylvol(cyls(i).dimensions);

    fprintf('Cylinder %c has a volume of %.1f in^3\n', . . .

    cyls(i).code, vol);

end

end

function cvol = cylvol(dims)

cvol = pi * dims.rad ^ 2 * dims.height;

end
```

==========

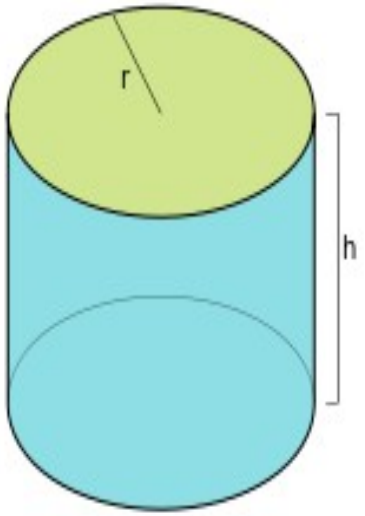
```
 >>printcylvols(cyls)

Cylinder x has a volume of 169.6 in^3

Cylinder a has a volume of 100.5 in^3

Cylinder c has a volume of 169.6 in^3
```

- Τροποποιήστε κατάλληλα το subfunction cylvol για να υπολογίζει και την επιφάνεια του κάθε κυλίνδρου (με το εμβαδό των βάσης και κορυφής) και να την τυπώνει δίπλα στον όγκο του.



Εμβαδό : 2 π r h + 2 π r^2

Διανύσματα cells

Ένα cell είναι το πιο γενικό αντικείμενο που περιέχει δεδομένα στο Matlab. Μπορεί να περιέχει αριθμούς σε πίνακες, strings, δομές, ή ακόμα και άλλα cells.

- Καταχώρηση δεδομένων σε cells

```
>>c(1,1) = {rand(3)};

>>c(1,2) = {char( ' Nikos ' , ' Nikoy ' ) };

>>c(2,1) = {13};

>>c(2,2) = { foititis }

>> c

c =

    [3x3 double]    [2x7 char  ]

    [        13]    [1x2 struct]

>>c(1,1)

>>c{1,1}
```

```
>>c{1,1} = rand(3);

>>c{1,2} = char( ' Nikos ' , ' Nikoy ' ) ;

>>c{2,1} = 13;

>>c{2,2} =  foititis
```

- Μπορούμε να χρησιμοποιήσουμε αγκύλες για να εισάγουμε ένα cell array σε μια πρόταση

```
>>b = { [1:5], rand(2); foititis, char(' Nikos ' , ' Amy ') }
```

- Η εντολή cell μας επιτρέπει να καταμερίσουμε ένα κενό cell array

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

a =

[ ]    [ ]

[ ]    [ ]

[ ]    [ ]
```

μετά μπορούμε να καταχωρίσουμε τα δεδομένα στα cell

```
>>a(2,2) = { magic(3) }
```

Πρόσβαση των δεδομένων σε cell array

Μπορούμε να έχουμε πρόσβαση στα δεδομένα ενός cell με αγκύλες

```
>>r = c{1,1}

r =

0.4447      0.9218      0.4057

0.6154      0.7382      0.9355

0.7919      0.1763      0.9169
```

και σε ένα υποσύνολο με παρενθέσεις

```
>>rnum = c{1,1}(2,3)

rnum =

0.9355
```

Χρησιμοποιώντας τα cell array

Οι συναρτήσεις varargin και varargout που επιτρέπουν σε μια συνρτηση να έχει μεταβλητό αριθμό ορισμάτων είναι cell array.

- Η συνάρτηση testvar έχει μεταβλητό αριθμό ορισμάτων οι οποίες διπλασιάζουν την τιμή τους

```
function [varargout] = testvar(varargin)

for i = 1:length(varargin)

    x(i) = varargin{ i };

end

for i = 1:nargout

    varargout{i} = 2*x( i );

end
```

Όταν μια συνάρτηση καλείται με την varargin το Matlab αυτόματα πακετάρει τα δεδομένα σε ένα cell array. Μετά τα δεδομένα μεταφέρονται στην συνάρτηση με τις αγκύλες.

- >>cellplot(c) >>celldisp(c)

Με την εντολή cellfunc μπορούμε να εφαρμόσουμε μια συνάρτηση σε ένα cell array. Για παράδειγμα

```
>>C = {1:10, [2; 4; 6], [ ] };

>> averages = cellfun(@mean, C)

averages =

    5.5000    4.0000       NaN
```

Για να βρούμε τις διαστάσεις κάθε πίνακα στο C

```
>>[nrows, ncols] = cellfun(@size, C)

nrows =

     1     3     0

ncols =

    10     1     0
```

Δημιουργήστε ένα cell array που περιέχει strings και κάνει σύντμηση των string σε τρεις χαρακτήρες. Επειδή τα αποτελέσματα δεν είναι αριθμοί να θέσετε την επιλογή UniformOutput σε False

```
>>days = {'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'};

>>abbrev = cellfun(@(x) x(1:3), days, ' UniformOutput ', false)

abbrev =

    'Mon'    'Tue'    'Wed'    'Thu'    'Fri'
```