GÖRÜNTÜ İŞLEME

DERS-3

AFYON KOCATEPE ÜNİVERSİTESİ

Görüntü İşleme (Temel Matlab)

İLİŞKİSEL OPERATÖRLER

- **8. KONTROL AKIŞ YAPILARI**
- MATLAB bir programlama dilidir. Bu nedenle diğer programla dillerindeki temel yapılar benzer şekilde kullanılmaktadır. if-else-end, switch, for, while, continue ve break yapılarını burada da göreceğiz.
- **İlişkisel ve mantıksal İşlemler**

- MATLAB diğer programlar gibi ilişkisel imleçleri kullanmaktadır.
-
- < Küçük
- <= Küçük ya da eşit
- > Büyük
- >= Büyük ya da eşit
- == Eşit
- ~= Eşit değil

Operatörler:

&	and operatörü
|	or operatörü
~	not operatörü

AFYON KOCATEPE ÜNİVERSİTESİ

SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

IF DEYIMI

if deyimi,Matlab dilinde **sartli dallanma(conditional branching)** adi verilen islemi gerçeklestiren bir deyimdir. Sartli dallanma, herhangi bir programlama dili için temel kontrol yapisidir. Sartli dallanma islemi sayesinde,bir program, kararlar alma imkanina kavusur; bir ifadenin sonucuna göre, bir komutlar dizisinin icra edilip edilmeyecegine karar verebilir. Ifadenin degeri, bir icradan digerine degisebilecegi için, bu özellik bir programa farkli verilere karsi farkli sekillerde davranma imkani saglar. Matlab dilinde sartli dallanma if ve else anahtar sözcükleri ile gerçeklestirilir. if deyiminin en basit sekli için yazilis biçimi asagidaki gibidir:

```
if ifade
    deyim1;
end
    deyim2;
```

AFYON KOCATEPE ÜNİVERSİTESİ

SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

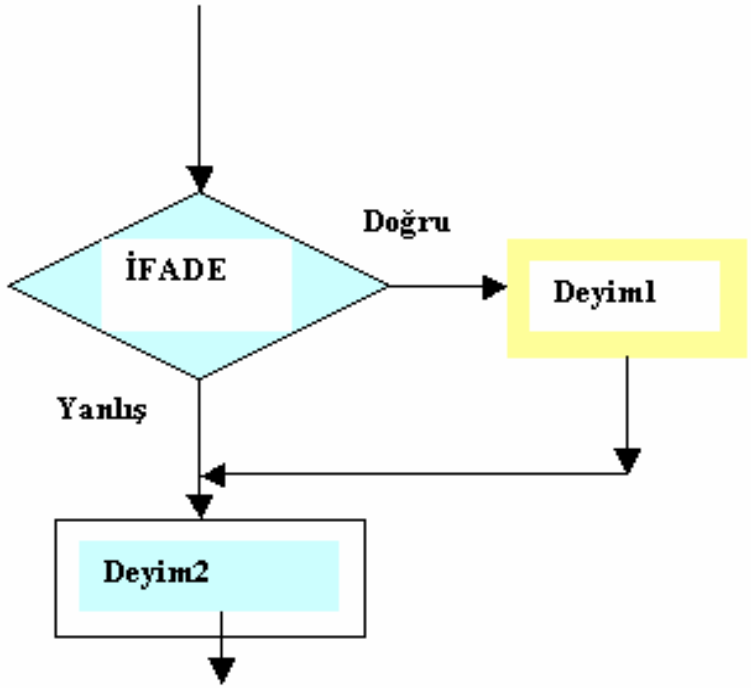
Burada ifadenin degeri dogru(true) ise deyim1 icra edilir sonra icra deyim2'ye geçer.Ifadenin degeri yanlis(false) ise bu durumda da dogrudan deyim2'ye geçilir. Ifadenin degeri yanlis ise deyim1 icra edilmeyecektir.Asagida if yapisini açiklayan bir akis diyagrami verilmistir.

if ifade

deyim1;

end

deyim2;

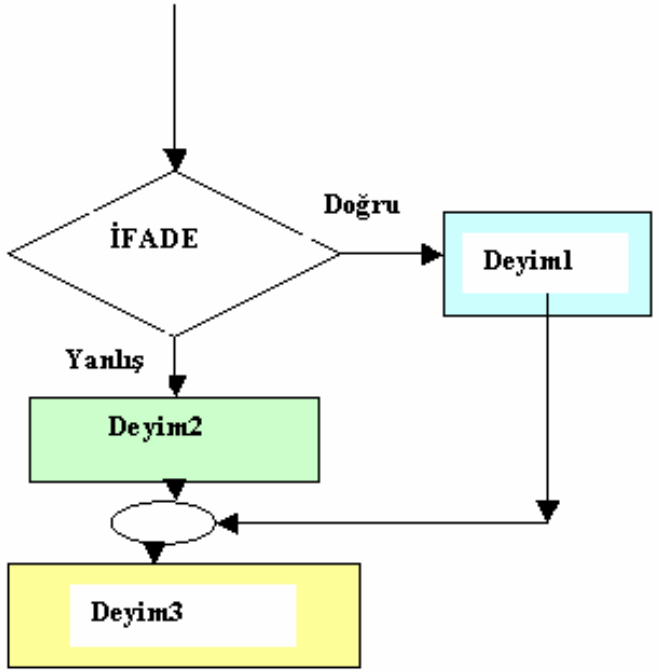


AFYON KOCATEPE ÜNİVERSİTESİ

SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

If deyimi else sözcügü ile birlikte kullanilirsa asagidaki yazilis biçimi kullanilir:

```
if ifade
deyim1 ;
else
deyim2;
end
deyim3;
...
```



if else deyimi için akış diyagramı(flow chart)

SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

- if-else-end
-
- En basit gösterim şekli;
-
- **if expression**
- **commands**
- **…**
- **end**
-
- Örneğin;
-

```
» A=[1 2; -3 6];
» if det(A)>0
    Ainv=inv(A);
    disp(Ainv)
  end
```

çıktısı aşağıdaki gibidir,çünkü determinant pozitif bir değerdir.

```
0.5000   -0.1667
0.2500    0.0833
```

AFYON KOCATEPE ÜNİVERSİTESİ

SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

r = round(-10 + (20+10).*rand(1,8))

r = [5 7 -3 4 19 6 6 -3]

Dizinin ortalaması pozitif ise pozitif dizi yazan ifadeyi yazalım.

```
if (mean(r)>0)
   fprintf('Pozitif Dizi')
End
```

Else yapısı ekleyelim.

r = round(-10 + (-1+10).*rand(1,8))

```
if (mean(r)>0)
   fprintf('Pozitif Dizi')
else
   fprintf('Negatif Dizi')
end
```

SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

- if-else-end yapısının en genel hali aşağıdaki gibidir.
-
- **if expression-1**
- **commands-1**
- **elseif expression-2**
- **commands-2**
- **.**
- **.**
- **.**
- **elseif expression-(n-1)**
- **commands-(n-1)**
- **else**
- **commands-n**
- **end**
-

SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

```
a=input('a değerini giriniz:');

    if a == 10
        % IF şartı doğruysa aşağıdakini yaz
        fprintf('Girilen değer= 10\n' );
    elseif( a == 20 )
        % ELSEIF şartı doğruysa aşağıdakini yaz
        fprintf('Girilen değer= 20\n' );
    elseif a == 30
        % ELSEIF şartı doğruysa aşağıdakini yaz
        fprintf('Girilen değer= 30\n' );
    else
        % Hiçbir şart doğru değil ise '
        fprintf('Hiçbir değer eşleşmedi\n');
    fprintf('Girilen değer: %d\n', a );
    end
```

AFYON KOCATEPE ÜNİVERSİTESİ

SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

switch DEYİMİ

Bir seçim söz konusu olduğunda bu yapı kullanılabilir. Genel gösterimi:

```
switch (selector)
    case label-1
        commands-1
    case label-2
            commands-2
        .
        .
        .
    case label-n
        commands-n
    otherwise
        commands-m
end
```

SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

- Örnek:
-
- **>> a=[4 5;2 3];**
- **>> switch (det(a))**
- **case 1**
- **b=a';**
- **disp(b)**
- **case 2**
- **b=a*a;**
- **disp(b)**
- **end**
-
- a matrisi determinantı 2 ye eşit olduğu için çıktısı aşağıdaki gibi olacaktır.
-
- 26 35
- 14 19

DÖNGÜLER

for döngüsü:

Bir grup bilginin birkaç defa değerlendirilmesi gerekiyorsa bu yapı kullanılır.

```
for x=array
    commands
end
```

Komutlar x'in bütün değerleri için işler. Örneğin;

```
» k=1;
» for num=[6 37 23 -1]
    disp([num2str(k), ' inci elementi ',
num2str(num)])
    k=k+1;
 end
```

çıktısı;

1 inci elementi 6
2 inci elementi 37
3 inci elementi 23
4 inci elementi -1

AFYON KOCATEPE ÜNİVERSİTESİ

DÖNGÜLER

Rastgele bir dizideki pozitif ve negatif sayıları yazdırınız.

```
r = round(-10 + (20+10).*rand(1,20)) % random negatif, pozitif sayılar
i=1;
for i=1:length(r)
   if (r(i)>0)
      fprintf('r(%d) pozitif. Değeri=%d\n',i,r(i));
   else
      fprintf('r(%d) negatif. Değeri=%d\n',i,r(i));
   end
end
```

DÖNGÜLER

Dizinin maksimumunu bulma örneği:

For ile çözüm.

```
rnd= round(-10 + (20+10).*rand(1,20));
maks=rnd(1);
j=1;
for i=1:20
    if (rnd(i)>maks)
        maks=rnd(i);
        j=j+1;
    end

end
fprintf('maksimum=%4.2f\n\n',maks);
fprintf('indis=%d\n',j)
```

AFYON KOCATEPE ÜNİVERSİTESİ

DÖNGÜLER

while döngüleri

Genel gösterimi;

while expression
commands
....
end

şeklindedir.

DÖNGÜLER

Dizinin maksimumunu bulma örneği:

```
rnd= round(-10 + (20+10).*rand(1,20));
maks=rnd(1);
j=1;i=1;
while (i<21)
   if (rnd(i)>maks)
      maks=rnd(i);
      j=j+1;
   end
i=i+1;
end
fprintf('maksimum=%4.2f\n\n',maks);
fprintf('indis=%d\n',j)
```

While ile çözüm.

AFYON KOCATEPE ÜNİVERSİTESİ

DÖNGÜLER

break komutu

break yapısı döngünün dışına ulaşmak için kullanılır.

```
» y=zeros(1,2);
» x=[2 3];
» while x>=0
    y=y+x;
    x=x-1;
    if y>=3
      break
    end
    disp(y)
  end
```

çıktısı;

2 3

(3, 5) vektörü görünmedi çünkü y>=3 ifadesi doğru bir ifade.

AFYON KOCATEPE ÜNİVERSİTESİ

DÖNGÜLER

continue

Bu yapı kontrol durumunu geçip sıradaki iterasyona ulaşmak için for ve while döngülerinde kullanılır.

```
>> y=zeros(1,2);
>> x=[2 3];
>> while x>=0
     y=y+x;
     x=x-1;
     if y<=3
       continue
     end
     disp(y)
   end
```

çıktısı;

3 5

3 6

(2, 3) vektörü görüntülenmedi. y<=3 ifadesi doğru olduğu için sıradaki iterasyonu atladı.

Görüntü İşleme (Temel Matlab)

For ve İf örnekleri

$X^3 - 4.2x^2 + 3.3x - 4$ fonksiyonunun köklerini bulunuz.

```
tic
for x=-5:0.0001:5
  t=x.^3-4.2*x.^2+3.3.*x -4;
%   if(t==0)
  if (t>=-0.001 && t<=0.001)
      fprintf('Kök=%2.5f',t);
      fprintf(' değer=%2.5f\n',x);
%       disp(t);
  end
end
plot(x,t,'LineWidth',3)
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
toc
```

Görüntü İşleme (Temel Matlab)

$f(x) = 2x - \cos x$

Fonksiyonun köklerini döngüler kullanarak bulunuz.

```
tic
clc
clear
for x=-5:0.0001:5
  t=2*x-cos(x);

  if (t>=-0.0001 && t<=0.0001)
      fprintf('Kök=%2.5f',x);
      fprintf(' değer=%2.5f\n',t);
  end
  end

toc
```

Görüntü İşleme (Temel Matlab)

1’den 999’a kadar olan sayılardan asal sayı olanların toplamını bulunuz.

1. yöntem:

```
total = 0;
for k = 1:999
if(isprime(k))
total = total + k;
end
end
disp(total)
```

2. yöntem

```
total = 0;
for k = primes(999)
total = total + k;
end
disp(total)
```

AFYON KOCATEPE ÜNİVERSİTESİ

Görüntü İşleme (Temel Matlab)

Aşağıdaki fonksiyonun -5:+5 aralığında grafiğini 10 farklı renk ile çizdiriniz.

$$y = x^2 - 1$$

```
x=(-5:5);
y=x.^2-1;
for i=1:10
    r1=rand(1);
    r2=rand(1);
    r3=rand(1);
p=plot(x,y);
p.Color=[r1 r2 r3];
p.LineWidth=2;
pause(1);
end
```

Görüntü İşleme (Temel Matlab)

1’den 200’e kadar olan sayılardan kendisi ve 2 fazlası asal olan sayıları bulunuz.

```
for x = 1:2:200
if(isprime(x) & isprime(x+2))
fprintf('%.0f ve %.0f ikisi de asal sayıdır\n',x,x+2)
end
end
```

Görüntü İşleme (Temel Matlab)

1’den 20’e kadar olan sayıları fibonacci dizisi olarak sıralayan programı yazınız.

```
clear
clc
tic
f = zeros(1,20);
f(1) = 1;
f(2) = 1;
for j = 3:20
f(j) = f(j-1) + f(j-2);
end
toc
```

Görüntü İşleme (Temel Matlab)

Ekrandan girilen bir sayının faktöriyelini bulunuz. Eğer sayı negatif ise uyarı verecek.

```
sayi=input('Bir sayı giriniz: ');
fact=1;
if sayi<0
fprintf('\nNegatif sayı girdiniz!\n\n');
else
    for i=1:sayi
        fact=fact*i;
    end
fact
end
```

Görüntü İşleme (Temel Matlab)

İç içe geçmiş döngüler (Nested Loops)

```
for x = 1:3
  for y = 1:2
  fprintf('x= %.0f and y= %.0f\n',x,y)
  end
end
```

```
x= 1 and y= 1
x= 1 and y= 2
x= 2 and y= 1
x= 2 and y= 2
x= 3 and y= 1
x= 3 and y= 2
```

Görüntü İşleme (Temel Matlab)

Bir matrisi satır vektöre dönüştüren döngüyü oluşturunuz.

```
[m n] = size(A);
k=1;
for i=1:m
 for j=1:n
  v(k) = A(i,j);
  k = k+1;
 end
end
```

Görüntü İşleme (Temel Matlab)

Bir matrisi satır vektöre dönüştüren döngüyü oluşturunuz.

```
M=magic(6);
[r c] = size(M);
V = [ ];
for col = 1:c
for row = 1:r
V(end+1) = M(row,col);
end
end
disp(V)
```

Fonksiyon yapıları

M-Fonksiyonlar kullanılırken dikkat edilecek hususlar:

1- Kullanıcılar kendi fonksiyonlarını yazmak için m-fonksiyonlarını kullanabilirler.

2- Function alt programı ve ana program şeklinde iki program yazılarak bu iki program ayrı ayrı kaydedilir.

3- Alt programdaki ***fonksiyon_adı,*** m-dosyasına verilen isimle aynı olmalıdır.

4- Ana programdan alt program, function adı kullanılarak çağrılır.

5- Alt programdan da ana programa geçiş yapılabilir fakat genelde tercih edileni tersidir.

6- Parametre aktarımı olması durumunda alt ve ana programda eşit sayıda parametre ve giriş değişkeni olmalıdır.

```
function cikis_ifadesi1, 2,..., n  =fonksiyon_adi (giris_ifadesi1, 2, ...n)
```

Fonksiyon yapıları

Örnek: İki nokta arasındaki uzaklığı bulan programı m-fonksiyon (alt program) kullanarak yazınız.

x1=1.noktanın x koordinati; x2=2.noktanın x koordinati
y1=1.noktanın y koordinati; y2=2.noktanın y koordinati

FUNCTION ALT PROGRAMI (uzak.m):

```
function uzaklik =uzak(x1,y1,x2,y2)
uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);
```

Bu function alt programı uzak.m olarak kaydedilir.

ANA PROGRAM:

```
ax=3; ay=4; bx=1; by=2;
uzaklik = uzak(ax,ay,bx,by); % uzak.m alt programını çağırıyor
fprintf('iki nokta arasindaki uzaklık=%f',uzaklik);
```

AFYON KOCATEPE ÜNİVERSİTESİ

Fonksiyon yapıları

Adım adım gerçekleştirilen işlemler:

- Ana program herhangi bir isimle kaydedilir ve koşturulur.
- Program, function adına (uzak) geldiği zaman alt program çağrılır

ve ax, ay, bx, by parametreleri sırasıyla x1, y1, x2, y2 giriş

değişkenlerine aktarılır.

- Function alt programında hesaplama gerçekleştirilir.
- Function'daki çıkış değişkeni olan uzaklik hem alt programda hem de ana programda hesaplanan sonuç değerinin aktarıldığı değişken olarak kullanılır.
- Alt programdan ana programa parametre aktarımı zorunlu değildir. İstenirse değişkenlerin değerleri alt programda da girilebilir ve sonuç alt programda yazdırılabilir.

Fonksiyon yapıları

Uygulama: Yukarıdaki örneği ana programdan alt programa parametre aktarımı yapmadan yeniden yazınız.

(Değişkenlerin girilmesi, sonucu hesaplama ve yazdırma işlemi alt programda yapılacaktır)

FUNCTION ALT PROGRAMI:

```
function uzaklik = uzak(x1,y1,x2,y2)
x1=3; y1=4; x2=1; y2=2;
uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);
fprintf('iki nokta arasindaki uzaklık=%f', uzaklik);
```

ANA PROGRAM:

```
uzaklik = uzak(ax,ay,bx,by); % uzak.m alt programını çağırıyor
```

Fonksiyon yapıları

Rastgele sayı üreten bir fonksiyon yazınız.

```
function r = RastgeleSayi( altSinir, ustSinir, satir,
sutun )
%UNTÝTLED Summary of this function goes here
%   Detailed explanation goes here
r = round(altSinir + (ustSinir-
altSinir).*rand(satir,sutun)); % random neagtif, pozitif
sayýlar

end
```

Fonksiyon yapıları

```
function [ maks, minimum ] = MaxMinBul( vektor )

maks=max(vektor);
minimum=min(vektor);
end
```

```
>> j=RastgeleSayi(-40,50,1,20);
>> [maksimum, minimum]=MaxMinBul(j)

maksimum =

   45

minimum =

  -29
```

Fonksiyon yapıları

```
function y = NormalizeEt01(x)
%b alt sýnýr, a ust sýnýr
[satir,sutun]=size(x);
i=1; j=1;
for i=1:satir
    for j=1:sutun
        y(i,j)=[(x(i,j)-min(x(:)))]/[max(x(:))-min(x(:))];
    end
end
```

Fonksiyon yapıları

Aşağıda verilen y fonksiyonunu hesaplayan bir matlab fonksiyonu yazınız.
X ve n değerlerinin girişleri dışarıdan verilecektir.

Fonksiyon
>> bToplam(x,n)
Şeklinde çalıştırılacaktır.

$$y = \sum_{k=1}^{n} \left(\frac{2}{x}\right)^{k}$$