

## Karar Yapıları

Bilgisayarlar karar verme ve kontrol işlemleri gerçekleştirerek işlemlerini yaptığını belirtmiştik. Matematiksel işlemleri gerçekleştirdiği gibi bilgisayarda hangi birimlerden hangi sırada giriş yapılacak hangi sırada çıkış yapılacak, öncelikler nasıl olacak vb. işlemleri de gerçekleştirir. Bu işlemleri gerçekleştirmek için programlama dillerinde kullanılan çeşitli komutlar bulunmaktadır. Bu komutların kullanılması, programdaki bazı değişkenlerin değerlerinin kontrol edilerek programın akışının değiştirilmesiyle gerçekleştirilir.

Alternatifler arasından seçim yaparak program akışını değiştiren komutlar doğru yanlış karar yapıları ve çoklu seçim karar yapısıdır. Akış kontrol komutları karşılaştırma ve mantıksal operatörleri kullanarak bu işlemlerini gerçekleştirir. Bu haftaya kadar basit algoritmaların programlarını oluşturduk. Bu algoritmaların içerisinde belirttiğimiz gibi mantıksal karar verme işlemleri mevcut değildir. Bu haftadan itibaren içerisinde mantıksal kararların verildiği ve program akışının bir noktadan başka bir noktaya geçtiği algoritmaların programlarının hazırlanması anlatılacaktır. Mantıksal karar verme işlemi sonucunda programın akışını daha önceki adımlara gönderildiği yapılar olduğunu belirtmiştik, bu yapıları oluşturmak için programlama dillerinde döngü komutları kullanılır. (Karar komutları kullanılarak programın akışı daha önceki adımlara göndermek için etiket yapısı kullanılması gerekmektedir.)

Karar yapıları algoritmalar içerisinde eğer şöyle ise durumda vb. gibi ifadelerin olduğu ve akışın ileri yönlü değiştirildiği durumlarda kullanılır. “Eğer öğrencinin notu 50 den büyükse ekrana geçti yaz.” , “Yaş değeri 12 ise kilo hesabı yapma” gibi karar verme işlemlerini gerçekleştirmek için karar yapıları kullanılmalıdır.

Birkaç seçenekten birini seçmek veya komutları belli bir koşula bağlı olarak işlemek için karar yapıları kullanılır. Günümüzde kullanılan programlama dillerinde doğru yanlış karar yapıları ve çoklu seçim karar yapısı olmak üzere genel iki tane karar yapısı vardır.

## Doğru Yanlış Karar Yapısı

Bilgisayarlarda karar verme işleminde karşılaştırma işleminin sadece doğru veya yanlış olduğuna karar verilebilir. Programlama dillerinde bu doğru veya yanlış olma durumunu kontrol eden komutlar vardır. Bir mantıksal karşılaştırma işleminin doğru veya yanlış olmasını kontrol eden yapılara doğru yanlış karar yapısı denilmektedir. Programlama dillerinde çeşitli doğru yanlış karar komutları kullanılmaktadır.

## ALGORİTMA VE PROGRAMLAMA

Doğru yanlış karar yapısı bir koşulun doğruluğu veya yanlışlığına göre işlemleri gerçekleştiren bir karar yapısıdır. Bu komutların kullanılması, programdaki bazı değişkenlerin veya bazı değerlerin değerlerinin kontrol edilerek programın akışı değiştirebilir.

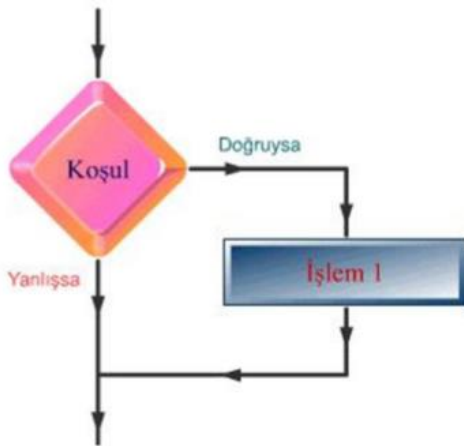
### Tek Alternatifli Karar Yapısı

Sadece mantıksal karşılaştırma doğru olduğunda işlemleri gerçekleştiren yanlış olduğunda herhangi bir işlem yapmadan bir sonraki adıma geçen karar yapısı kullanılabilir. Algoritmada “Eğer ....ise .....” gibi, akış diyagramı ise aşağıdaki şekildeki gibi olabilir.

C programlama dilinde bu karar yapısını gerçeklemek için if komutu kullanılır.

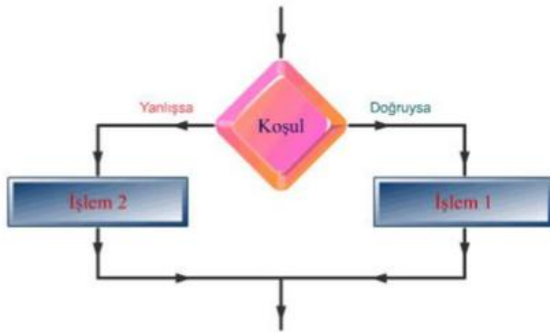
Kullanım şekli	Açıklama
if(koşul) işlem;	Eğer karşılaştırma işleminden sonra karar doğru olduğunda tek işlem gerçekleştirilecekse
if(koşul) { ..... işlemler; ..... }	Eğer karşılaştırma işleminden sonra karar doğru olduğunda birden fazla işlem gerçekleştirilecekse bu durumda işlemler {} “süslü parantezler” arasında yazılır.

### İki Alternatifli Karar Yapısı



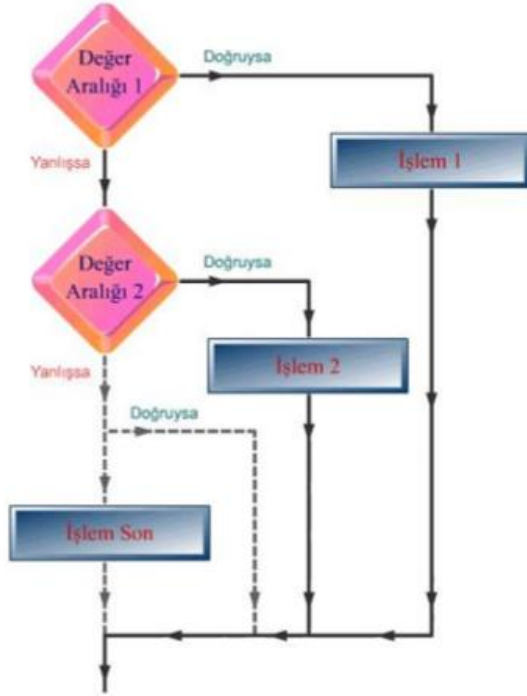
## ALGORİTMA VE PROGRAMLAMA

Eğer mantıksal karşılaştırma doğru olduğunda bazı işlemler gerçekleştirilecek yanlış olduğunda başka işlemler gerçekleştirilecek ise doğru yanlış karar yapısı karar yapısı kullanılabilir. Algoritmada “Eğer ....ise ..... değil ise....” gibi, akış diyagramı ise aşağıdaki şekildeki gibi olabilir. C programlama dilinde bu karar yapısını gerçeklemek için if..else komutu kullanılır.



Kullanım şekli		Açıklama
	<pre> if(koşul)     işlem1; else     işlem2;</pre>	Eğer mantıksal karşılaştırma işleminden sonra tek komut çalıştırılacaksa, mantıksal karşılaştırma doğru ise işlem1 yanlış olduğunda işlem2 gerçekleştirilir.
	<pre> if(koşul) { ..... 1.işlemler; ..... } else { ..... 2.işlemler; ..... }</pre>	Eğer karşılaştırma işleminden sonra birden fazla komut çalıştırılacaksa, mantıksal karşılaştırma doğru ise 1.işlemler yanlış olduğunda 2.işlemler gerçekleştirilir ve bu işlemler {} "süslü parantezler" arasında yazılır.

## Çok Alternatifli Karar Yapısı



Bazı durumlarda karşılaştırma işlemi yapıldıktan sonra doğru veya yanlış olduğunda tekrar karşılaştırmalar gerçekleştirilebilir. Bu yapıya içiçe if yapısı veya çoklu seçim karar yapısı da denilir. Akış diyagramı ise aşağıdaki şekildeki gibi olabilir.

C programlama dilinde bu karar yapısını gerçeklemek için if..else if komutu kullanılır.

Kullanım şekli	Açıklama
<pre> if (koşul_1) işlem1 else if (koşul_2) işlem2 ..... else if (koşul_n) işlemln else diğer_işlem; </pre>	<p>Birinci mantıksal karşılaştırma doğru ise işlem_1 yapılır, değil ise ikinci mantıksal karşılaştırmaya bakılır, ikinci karşılaştırma doğru ise işlem_2 yapılır, bu şekilde devam ederek son mantıksal karşılaştırmaya kadar devam edilir. Hiçbir karşılaştırma doğru değil ise diğer işlem gerçekleştirilir. Eğer karşılaştırma işlemlerinden sonra birden fazla komut çalıştırılacaksa, bu işlemler {} "süslü parantezler" arasında yazılır.</p>

## Switch-Case

if deyimi Doğru ya da Yanlış olmak üzere iki seçeneğe sahiptir. Oysa switch deyimi belli bir ifadenin çeşitli sayısal değerlerine karşı farklı işlemlerin yapılması için kullanılmaktadır. Genel biçimi aşağıdaki gibidir:

```
switch (ifade) {  
    case <sabit ifadesi1>: [break];  
    case <sabit ifadesi2>: [break];  
    case <sabit ifadesi3>: [break];  
    [default:      ;]  
}
```

switch, case ve default birer anahtar sözcüktür. Derleyici switch parantezinin içerisindeki ifadenin sayısal değerini hesaplar. Eğer bu sayısal değere eşit olan bir case ifadesi bulursa programın akışını oraya yönlendirir. Eğer böyle bir ifade yoksa programın akışı default ile belirtilen kısma geçer. Örneğin:

```
switch(n) {  
case 1: printf("Pazartesi\n");  
case 2: printf("Salı\n");  
case 3: printf("Çarşamba\n");  
case 4: printf("Perşembe\n");  
case 5: printf("Cuma \n");  
default: printf("Hiçbiri\n");  
}
```

burada n=1 ise programın akışını case 1: kısmına yöneltir. Akış buradan sırasıyla aşağı inerek devam edecektir. Dolayısıyla ekranda:

Pazartesi

Salı

Çarşamba

Perşembe

Cuma

Hiçbiri

yazılarını görürsünüz. Döngülerden çıkmak için kullanılan break anahtar sözcüğü switch içerisinde kullanılabılır. Bu durumda bir case ifadesi break ile sonlandırılırsa

## ALGORİTMA VE PROGRAMLAMA

programın akışı diğer case ifadelerine geçmez. Yukarıdaki örnekteki her bir case in sonuna break koyarsak örneğimiz şu şekilde değişecektir.

```
switch(n) {
case 1: printf("Pazartesi\n");break;
case 2: printf("Salı\n"); break;
case 3: printf("Çarşamba\n"); break;
case 4: printf("Perşembe\n"); break;
case 5: printf("Cuma\n"); break;
default: printf("Hiçbiri\n"); break;
}
```

bu program bir öncekinin aksine tüm ifadeleri değil sadece n değerinin karşılığı olan günü yazacaktır. Yani n değeri 3 ise Çarşamba, 1 ise pazartesi, 12345 dışında bir değerse hiçbirini yazacaktır.

Örnek: Klavyeden girilen iki sayıya uygulanabilecek dört işlemi soran seçime göre işlemi yapıp

ekrana yazdıran programın c kodunu yazınız.

```
#include<stdio.h>
```

```
int main()
{
    int secim;
    float x,y, sonuc;
    printf("İki sayı girin: ");
    scanf("%f %f",&x,&y);

    puts("*** Menu ***");
    puts("[1] Toplama");
    puts("[2] Cikarma");
    puts("[3] Carpma");
    puts("[4] Bolme");
    printf("Seciminiz: ");
    scanf("%d",&secim);

    switch( secim )
    {
        case 1:
            sonuc = x + y;
            printf("Toplam = %f\n",sonuc);
            break;

        case 2:
            sonuc = x-y;
            printf("Fark = %f\n",sonuc);
            break;

        case 3:
            sonuc = x * y;
            printf("Carpim = %f\n",sonuc);
            break;

        case 4:
            sonuc = x/y;
            printf("Oran = %f\n",sonuc);
            break;

        default:
            puts("Yanlis secim !\a");
    }

    return 0;
}
```

## ? Karşılaştırma Operatörü

Bu operatör, if-else karşılaştırma deyiminin yaptığı işi sınırlı olarak yapan bir operatördür.

Genel yazım biçimi:

(koşul) ? deyim1 : deyim2

şeklindedir.

İlk önce koşul sınanır. Eğer koşul olumluysa deyim1 aksi takdirde deyim2 değerlendirilir. deyim1 ve deyim2 de atama işlemi yapılamaz. Ancak koşul deyiminde atama işlemi yapılabilir. deyim1 ve deyim2 yerine fonksiyon da kullanılabilir.

```
#include<stdio.h>

int main(void)
{
    float x, y, z;

    printf("x : ");
    scanf("%f",&x);
    printf("y : ");
    scanf("%f",&y);

    if(y) /* y, 0'dan farklı mı? */
        z = ( y > x ) ? x/y : x*y; /* y>x ise z = x/y, değilse z = x*y */
    else
        z = 0.0;

    printf("z = %f\n",z);

    return 0;
}
```