

Akış Kontrol Mekanizmaları (Karar Yapıları)

Öğr. Gör. Erkan HÜRNALI

Yol Ayrımı



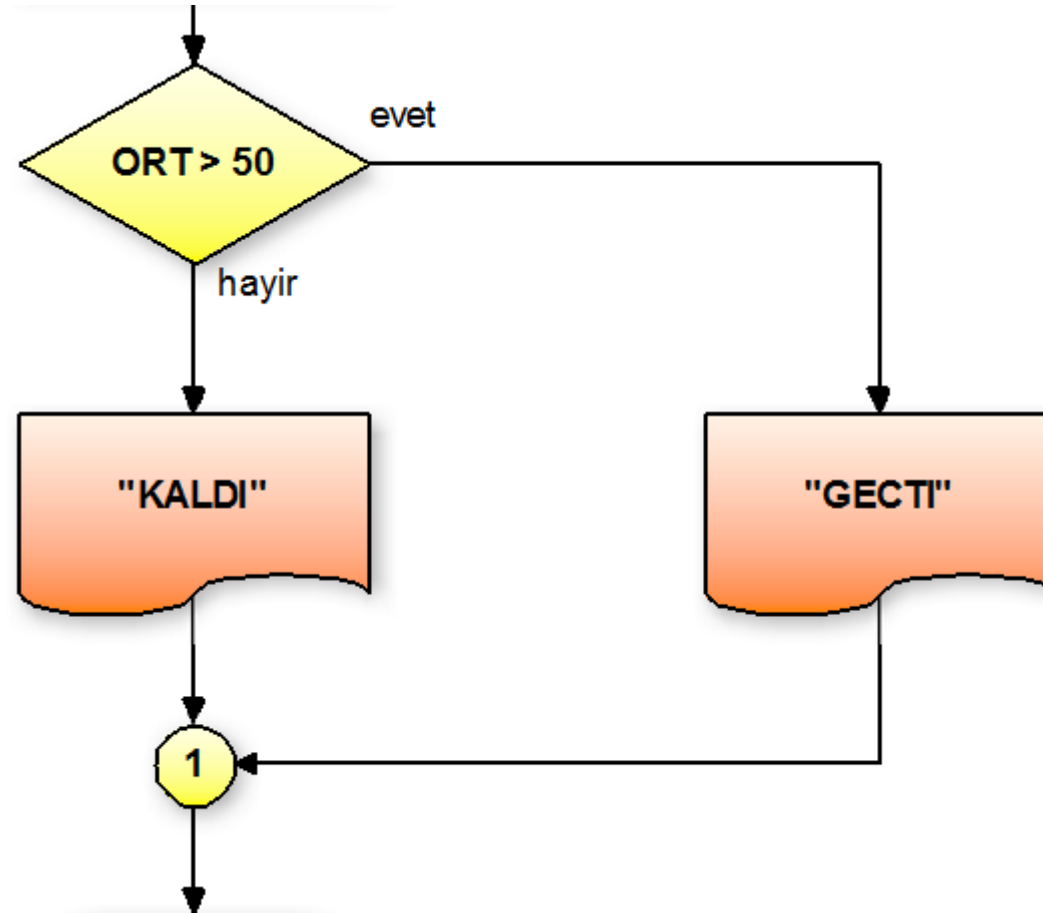
Öğreneceklerimiz

- Akış Kontrol Nedir?
- Neden İhtiyaç Duyarız?
- Karar Yapıları
 - if – Else
 - Switch
- Örnekler

Öğrendiklerimiz



Akış Kontrol Mekanizmaları



C#'da Akış Kontrol Mekanizmaları

```
if (true)
{

}
else
{

}
```

```
switch (meyve)
{
    case "elma": break;
    case "armut": break;
    case "muz": break;

    default: break;
}
```

if - else

if else deyimi sayesinde belli bir koşul sağlandığında söz konusu komutlar çalıştırılır, o belli koşullar sağlanmadığında çalıştırılmaz ya da başka komutlar çalıştırılır. Kullanılışı şu şekildedir:

```
if(koşul)
    komut1;
else
    komut2;
```

veya

```
if(koşul)
{
    komutlar1
}
else
{
    komutlar2
}
```

if - else

Yukarıdaki örneklerde eğer koşul sağlanırsa 1. komutlar, sağlanmazsa 2. komutlar çalıştırılır. if veya else'in altında birden fazla komut varsa bu komutları parantez içine almak gerekir. if veya else'in altında tek komut varsa bu komutları parantez içine almak gerekmez. Örnek bir program:

```
using System;
class AkisKontrolMekanizmalari
{
    static void Main()
    {
        int a=5, b=7;
        if(a<b)
            Console.Write("a b'den küçük");
        else
            Console.Write("a b'den küçük değil");
    }
}
```


if - else

Başka bir örnek program:

```
using System;
class AkisKontrolMekanizmalari
{
    static void Main()
    {
        int a=5, b=7;
        if(a<b)
        {
            Console.WriteLine("a b'den küçük");
            Console.Write(a);
        }
        else
        {
            Console.WriteLine("a b'den küçük değil");
            Console.Write(b);
        }
    }
}
```

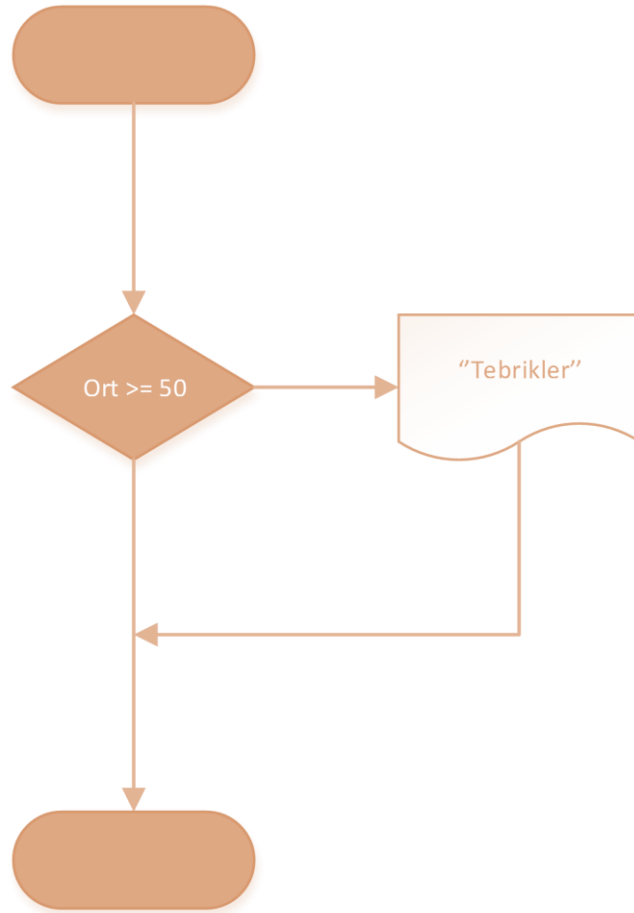
if - else

if else yapılarında else kısmının bulunması zorunlu değildir. Bu durumda sadece koşul sağlandığında bir şeyler yapılacak, koşul sağlanmadığında bir şeyler yapılmayacaktır. Örnek:

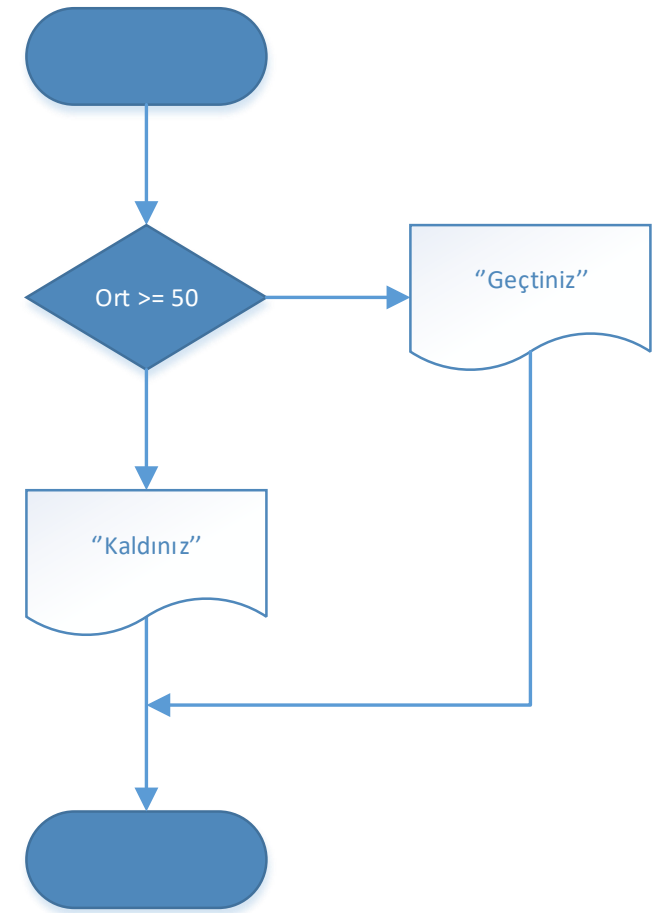
```
using System;
class AkisKontrolMekanizmalari
{
    static void Main()
    {
        int a=5, b=7;
        if(a<b)
            Console.WriteLine("a b'den küçük");
    }
}
```

if - else

```
if (true)  
{  
}  
}
```



```
if (true)  
{  
}  
else  
{  
}  
}
```



switch

switch deyimi bazı if else deyimlerinin yaptığı işi daha az kodla yapar. Genellikle bazı karmaşık if else bloklarını kurmaktansa switch'i kullanmak programın anlaşılabilirliğini artırır. Ancak tabii ki basit if else bloklarında bu komutun kullanılması gereksizdir. Kuruluşu:

```
switch(ifade)
{
    case sabit1:
        komut1;
        break;
    case sabit2:
        komut2;
        break;
    default:
        komut3;
        break;
}
```

switch

- İfadenin ürettiği değer hangi case sabitinde varsa o "case"deki komutlar işletilir. Eğer ifadenin ürettiği değer hiçbir case sabitinde yoksa default casedeki komutlar işletilir.
- Aynı birden fazla case sabiti olamaz. Örneğin:

```
using System;
class AkisKontrolMekanizmalari
{
    static void Main()
    {
        int a=4;
        switch(a)
        {
            case 4:
                Console.WriteLine("deneme1");
                break;
            case 4:
                Console.WriteLine("deneme2");
                break;
            case 5:
                Console.WriteLine("deneme3");
                break;
            default:
                Console.WriteLine("deneme4");
                break;
        }
    }
}
```

switch

Bu program hatalıdır.

- C#'ta herhangi bir case'e ait komutların break; satırı ile sonlandırılması gerekmektedir. Eğer break; satırı ile sonlandırılmazsa programımız hata verir. Örneğin aşağıdaki program hata vermez:

```
using System;
class AkisKontrolMekanizmalari
{
    static void Main()
    {
        int a=4;
        switch(a)
        {
            case 4:
                string b=Console.ReadLine();
                Console.Write(b);
                break;
            default:
                Console.Write("deneme4");
                break;
        }
    }
}
```

switch

- Eğer programımızın bir case'deyken farklı bir case'e gitmesini istiyorsak goto anahtar sözcüğünü kullanırız. Örnek:

```
using System;
class AkisKontrolMekanizmalari
{
    static void Main()
    {
        int a=5;
        switch(a)
        {
            case 4:
                string b=Console.ReadLine();
                Console.Write(b);
                break;
            case 5:
                Console.Write("Şimdi case 4'e gideceksiniz.");
                goto case 4;
            default:
                Console.Write("deneme4");
                break;
        }
    }
}
```

switch

- goto satırı kullanılmışsa break; satırının kullanılmasına gerek yoktur.
- Eğer farklı case sabitlerinin aynı komutları çalıştırmasını istiyorsak şöyle bir program yazılabilir:

```
using System;
class AkisKontrolMekanizmalari
{
    static void Main()
    {
        int a=5;
        switch(a)
        {
            case 4:
            case 5:
                string b=Console.ReadLine();
                Console.Write(b);
                break;
            default:
                Console.Write("deneme4");
                break;
        }
    }
}
```


switch

Bu programda a deęiřkeni 4 olsa da 5 olsa da aynı komutlar alıřtırılacaktır.

- case anahtar sözcüęünün yanındaki ifade mutlaka ya sabit ya da sabitlerden oluşan bir ifade olmalıdır.
- default durumunu istedięiniz yere yazabilirsiniz, aynı řekilde istedięiniz case'leri de istedięiniz yere yazabilirsiniz. Yani case'lerin sırası önemli deęildir.
- Bir switch bloęunda default durumu bulunmak zorunda deęildir.
- switch'in parantez içindeki ifadesi bir deęiřken olabileceęi gibi, bir sabit ya da ifade de olabilir.

Uygulama Ödevi

