

Bölüm 08

Program Akışının Yönlendirilmesi

blok,
if yönlendirmesi
switch yönlendirmesi
for döngüsü
while döngüsü
do ... while döngüsü

Bir program komutların yazıldığı sırada akar. Ama çoğunlukla, bu akışı yönlendirmek gerekir. Bu iş için denetim yapılarını kullanırız. Bunlar başlıca üç gruba ayrılabilir: **Bloklar**, **Yönlendiriciler** ve **Döngüler**. Bu yapılardan birisine girince, program o yapı içinde istenen her işi yapar. Bilgisayarın karmaşık işleri yapmasını sağlayan bu yapılar altı tanedir.

Bu yapılardan her birisi tek bir varlık olarak düşünülür, ama gerçekte her birisi birden çok deyim içeren birer yapıdır. Bunların her birisini örneklerle açıklayacağız.

Blok

Bir arada yürütülmesi istenen deyimleri içeren bir yapıdır. Bir blok içine giren program, aksi söylenmedikçe, o blok içindeki bütün deyimleri çalıştırır. Blok yapısının sözdizimi şöyledir:

```
{  
    deyimler  
}
```

{ } bloku içindeki deyimler istenildiği sayıda C# deyimlerinden oluşur. Hiç deyim içermeyen bloklara boş blok denir. İç-içe bloklar olabilir. Deyimler, basit C# deyimleri olabileceği gibi, yapısal deyimler de olabilir.

Örnekler

```
{
    System.Console.WriteLine("Ankara başkenttir. ");
    System.Console.WriteLine(x);
}
```

```
{ // Bu blok x ile y değişkenlerinin yerlerini değiştirir
    int yedek; // blok içinde kullanılacak geçici değişken
    yedek = x; // x değerini yedek adresine al
    x = y; // y değerini x adresine al
    y = yedek; // yedek değerini y nin adresine al
}
```

Blok içinde bildirimi yapılan bir değişken, o blok için bir yerel değişkendir. Bir bloğun yerel değişkenine blok dışından erişilemez. Blok içinden dış bloka erişilebilir. Başka bir deyişle, blok içindeki bir operatör ya da metot, dış bloktaki değişkenleri ve metotları çağırabilir. Bloğun işi bitince, yerel değişkene ayrılan adres yok olur. Bu nedenle, bir değişkenin erişilebilirlik bölgesi (scope) o değişkenin tanımlandığı bloktur.

Yönlendiriciler

Program akışını, belli mantıksal koşullara göre istenen yöne saptıran denetim yapılarıdır. Bunlar da kendi içinde üçe ayrılır: if, if-else ve case yapıları.

if deyimleri

Program akışını, mantıksal koşullara (boolean) göre istenen yöne saptıran denetim yapılarıdır. Bu yapının üç türü vardır: tek seçenekli if, if-else ve çok seçenekli if.

1. Tek seçenekli if
2. if-else seçeneği
3. çoklu-durum seçeneği

Şimdi bunların her birisini örneklerle açıklayacağız.

Tek Seçenekli if

Bazı deyimlerin işlenmesini, ancak belirli koşulların sağlanması durumunda isteyebiliriz. Bu durumda if yönlendirmesini kullanırız. Bu yönlendirmenin sözdizimi yapısı aşağıdaki iki durumdan birisi gibidir.

- a. Eğer if denetiminden sonra bir tek deyim işleyecekse,

```
if ( mantıksal_deyim )
    deyim ;
```

biçimindedir.

- b. Eğer if denetiminden sonra birden çok deyim işleyecekse, onlar bir blok ({ }) içine alınır.

```
if ( mantıksal_deyim )
{
    deyimler;
}
```

Eğer mantıksal_deyim true değerini alıyorsa deyim(ler) işlenir ve program if yapısından sonraki deyimde geçer. Eğer mantıksal_deyim false değerini alıyorsa deyim(ler) işlenmeden atlanır ve program if yapısından sonraki deyimde işlemeye başlar.

IfYapısı01.cs

```
//if deyiminin kullanılışına örnektir
```

```

using System;

namespace DenetimYapıları
{
    class ifYapısı01
    {
        static void Main(string[] args)
        {
            string s;
            int n;

            Console.WriteLine("Bir tamsayı giriniz: ");
            s = Console.ReadLine();
            n = Int32.Parse(s);

            //if yönlendirmesi
            if (n > 0)
                Console.WriteLine("Girdiğiniz {0} sayısı pozitiftir.", n);
        }
    }
}

```

Çıktı

Bir tamsayı giriniz:

45

Girdiğiniz 45 sayısı pozitiftir.

Bu program koşarken, ekrana önce “Bir tamsayı giriniz: “ iletisi gelir. Bu iletie karşılık, pozitif bir tamsayı girerseniz, örneğin 45, ekrana “Girdiğiniz 45 sayısı pozitiftir.” iletisi gelir. Eğer 0 ya da negatif bir tamsayı girerseniz, program girdiğiniz sayıyı string olarak okur, ve int tipine dönüştürür. Ama sayı pozitif olmadığı için if blokuna girmez. If blokundan sonraki deyme geçer. Ama if blokundan sonra başka deyim olmadığı için başka bir iş yapmadan program sona erer.

Eğer kullanıcı int tipinden başka bir değer girerse, `n = int.Parse(s)` tip dönüşümü olamayacağı için, program koşma hatası (runtime error) ile karşılaşır aşağıdaki uyarıyı vererek kesilir:

Unhandled Exception: System.FormatException: Input string was not in a correct format...

Aşağıdaki program, if denetiminin bir mantıksal deymi denetlediğini ve deyim true değerini alıyorsa if yapısına girildiğini göstermektedir.

```

/*
if denetimi için mutlaka bir mantıksal deyim (boolean) gereklidir.
*/
using System;

namespace DenetimYapıları
{
    class GeçersizIf
    {
        public static void Main()
        {
            if (1)
                Console.WriteLine("The if statement executed");
        }
    }
}

```

Bu program derleme anında şu hata mesajını verir:

Constant value '1' cannot be converted to a 'bool'

Bu ileti bize şunu söylüyor: Kaynak programdaki `if(1)` satırında sözdizimi hatası vardır. `if` anahtar sözcüğünden sonra mutlaka bir boolean deyim gelmelidir. Oysa, programda `if` sözcüğünden sonra yazılan `(1)` ifadesi bir boolean değildir. '1' booleana dönüştürülemez. *[C dilinde program yazarlar, bu biçimin o dilde geçerli olduğunu anımsayacaklardır. C# dili C dili üzerine kurulmuş olmakla birlikte, onun bazı niteliklerini değiştirmiş, iyileştirmiştir. Dolayısıyla, C dilinde geçerli olan her şey C# dilinde geçerli değildir.]*

if-else yönlendirmesi

Bazı durumlarda önümüze iki seçenek çıkar. Belirli bir koşulun sağlanması durumunda seçeneklerden birinin, aksi halde ötekinin işlemesi istenebilir. Başka bir deyişle, bir koşulun sağlanıp sağlanmamasına bağlı olarak, iki seçenekten birisini mutlaka yaptırmak gerekir. Bu durumda, `if-else` yapısını kullanırız. Bu yapının sözdizimi şöyledir:

```
if ( mantıksal_deyim )
    deyim-1
else
    deyim-2
```

Eğer `deyim-1` ve `deyim-2` yerinde işlenecek birden çok deyim varsa, onlar bir blok içine alınabilir:

```
if ( mantıksal_deyim )
{
    deyimler-1
}
else
{
    deyimler-2
}
```

Eğer `mantıksal_deyim` `true` değerini alıyorsa `deyim(ler)-1` işlenir ve program `if` denetim yapısından sonraki deyimi işlemeye başlar. Eğer `mantıksal_deyim` `false` değerini alıyorsa `deyim(ler)-1` işlenmeden atlanır ve `deyim(ler)-2` işlenir. Sonra, program `if` denetim yapısından sonraki deyimi işlemeye başlar. Bu yapıda program ya `deyim(ler)-1` ya da `deyim(ler)-2` 'yi işler.

Aşağıdaki programları satır satır inceleyiniz. Her satırın işlevini algılayınız.

Program

```
using System;

namespace DenetimYapıları
{
    class IfElseYapısı
    {
        static void Main(string[] args)
        {
            string s;
            int i;

            Console.WriteLine("Bir tamsayı giriniz: ");
            s = Console.ReadLine();
            i = Int32.Parse(s);

            if (i > 0)
                Console.WriteLine("Girdiğiniz {0} sayısı pozitiftir ", i);
            else
```

```

        Console.WriteLine("Girdiğiniz {0} sayısı pozitif değildir ", i);
    }
}

```

Aşağıdaki program verilen bir yılın artık yıl olup olmadığını bulur.

Program

```

//Metotların iç-içe kullanımına örnektir.
using System;

namespace DenetimYapıları
{
    class ArtıkYıl
    {
        public static void Main()
        {
            int yıl;
            Console.WriteLine("Hangi yıl? (yyyy) :");
            yıl = Int32.Parse(Console.ReadLine());
            if ((yıl % 4 == 0 && yıl % 100 != 0) || yıl % 400 == 0)
                Console.WriteLine(" {0} yılı artık yıldır. Şubat ayı 29
çeker. ", yıl);
            else
                Console.WriteLine(" {0} yılı artık değildir. Şubat ayı 28
çeker. ", yıl);
        }
    }
}

```

Çıktılar

Programı

```

Hangi yıl? (yyyy) :
2007
2007 yılı artık yıl değildir. Şubat ayı 28 çeker.

```

```

Hangi yıl? (yyyy) :
2008
2008 yılı artık yıldır. Şubat ayı 29 çeker.

```

c. Çoklu-durum seçeneği

Bazen ikiden çok seçenek ortaya çıkabilir. Bu durumlarda else-if denen çoklu-durum yapısını kullanırız:

```

if (Koşul_1)
    Deyim_1;
else if (Koşul_2)

```

```

        Deyim_2;
    else if (Koşul_3)
        Deyim_3;
    ...
    else
        Deyim_n;

```

Else-ifYapısı01.cs

```

using System;

namespace DenetimYapıları
{
    class ElseIfYapısı01
    {
        static void Main(string[] args)
        {
            string s;
            int i;

            Console.WriteLine("Bugün ısı kaç derecedir? ");
            s = Console.ReadLine();
            i = Int32.Parse(s);

            if (i < 10)
                Console.WriteLine("Bugün hava soğuktur. ");
            else
                if (i < 20)
                    Console.WriteLine("Bugün hava serindir. ");
                else
                    Console.WriteLine("Bugün hava sıcaktır. ");
        }
    }
}

```

Else-ifYapısı02.cs

```

using System;

namespace DenetimYapıları
{
    class ElseIfYapısı02
    {
        static void Main(string[] args)
        {
            string s;
            int i;

            Console.WriteLine("Bir tamsayı giriniz ");
            s = Console.ReadLine();
            i = Int32.Parse(s);

            if (i %3 == 0)
                Console.WriteLine("{0} tamsayısı 3 ile tam bölünür. " , i
);

```

```

        else
            if (i %2 == 0)
                Console.WriteLine("{0} tamsayısı çift sayıdır. ",
i );
            else
                Console.WriteLine("{0} tamsayısı tek sayıdır, ama 3
ile tam bölünmez. ", i );
        }
    }
}

```

Aşağıdaki programlar else if yapılarının istenildiği kadar iç-içe konulabileceğini göstermektedir.

Else-ifYapısı03.cs

```

using System;

namespace DenetimYapıları
{
    class ElseIfYapısı03
    {
        static void Main(string[] args)
        {
            string s;
            int i;

            Console.WriteLine("Not ortalamanızı giriniz : ");
            s = Console.ReadLine();
            i = Int32.Parse(s);

            if (i > 90)
                Console.WriteLine("Puanınız {0} ise notunuz A olur. ", i );
            else
                if (i > 75)
                    Console.WriteLine("Puanınız {0} ise notunuz B olur. ", i );
                else
                    if (i > 60)
                        Console.WriteLine("Puanınız {0} ise notunuz C olur. ", i );
                    else
                        if (i > 50)
                            Console.WriteLine("Puanınız {0} ise notunuz D olur. ", i );
                        else
                            Console.WriteLine("Puanınız {0} ise notunuz F olur. ", i );
        }
    }
}

```

Else-ifYapısı04.cs

```

using System;

namespace DenetimYapıları
{
    public class ElseIfTest04
    {
        public static void Main()
        {
            Console.Write("Bir tuşa basınız: ");
            char c = (char)Console.Read();

            if (Char.IsUpper(c))

```

```

        Console.WriteLine("Büyük harf girdiniz.");
    else if (Char.IsLower(c))
        Console.WriteLine("Küçük harf girdiniz.");
    else if (Char.IsDigit(c))
        Console.WriteLine("Bir rakam girdiniz.");
    else
        Console.WriteLine("Alfasayısal olmayan bir karakter
girdiniz.");
    }
}
}

```

Programın farklı koşullarında, girilen karaktere bağlı olarak çıktılar şöyle olabilir.

Bir tuşa basınız: K
Büyük harf girdiniz.

Bir tuşa basınız: b
Küçük harf girdiniz.

Bir tuşa basınız: 7
Bir rakam girdiniz.

Bir tuşa basınız: %
Alfasayısal olmayan bir karakter girdiniz.

switch denetim yapısı

Program akışı bir yerde çok sayıda seçenekle karşılaşırsa, iç-içe else if denetimi yerine switch denetimi daha kolay olur. Sözdizimi şöyledir:

Sözdizimi (syntax)

```

switch (seçici)
{
    case seçki-1:
        deyim-1;
        sıçrama-1;

    case seçki-2:
        deyim-2;
        sıçrama-2
    ...
    [default:
        deyim
        sıçrama]
}

```

Bu yapıda case ifadelerinin sayısı için bir kısıt yoktur, gerektiği kadar case ifadesi konulabilir. Bu sözdiziminde geçen terimlerin açıklanması:

seçici

Seçici bir değişken veya bir ifade olabilir. Seçici değişken olduğunda `sbyte`, `byte`, `short`, `ushort`, `int`, `uint`, `long`, `ulong`, `char`, `string` ya da `enum` türlerinden herhangi birisinden bir literal (sabit değer) almalıdır. Seçici bir ifade ise bu türlerden literal bir değer veren formüldür. Burada formül, matematiksel formüllerde olduğu gibi, bir sonuç veren deyim olarak algılanmalıdır.

seçki

Seçici değişkenin veya seçici ifadenin aldığı bir değerdir. Bu değer tamsayı, `char` ya da `string` türünden bir literal (sabit değer) olmalıdır.

deyim

Seçici-değerine bağlı olarak yönlendirilen bir seçenekte yapılacak iş(ler)i belirleyen deyim veya deyimlerden oluşan bir bloktur.

sıçrama

Bir seçeneğe yönlendirilen program, o seçenekteki işleri yapınca ya `switch` yapısının dışına çıkar ya da başka bir seçeneğe yönlendirilebilir. `Switch` yapısından çıkmak için

```
break;
```

deyimi kullanılır. `break` deyimi, akışı `switch` yapısından çıkarır ve `switch` yapısından sonraki ilk deyime gönderir.

Bazan bir seçenekteki işler bitince, akışı `switch` yapısından sonraki ilk deyime göndermek yerine başka bir seçeneğe göndermek gerekebilir. Bunun için

```
goto
```

deyimi kullanılır.

default

İsteğe bağlı bir seçenektir. Seçicinin değeri `case` ile belirlenen hiçbir seçki ile uyuşmadığı zaman, yapılmasını istediğimiz işler için gerekli deyim(ler)i bu seçeneğe yazarız. Hiçbir `case` yönlendirmesi gerçekleşemediğinde `default` seçenek çalışır. Seçicinin değeri mutlaka `case`'lerdeki seçkilerden birisine eşit olacağı garanti edilen durumlarda, `default` seçeneğini yazmanın gereği yoktur.

`switch` yapısının nasıl kullanıldığını aşağıdaki örnekte göreceğiz.

Switch01.cs

```
/*
switch-case yapısında seçici olarak string tipi değişken
kullanılmasına örnek.
*/
using System;

namespace DenetimYapıları
{
    class Switch01
    {
        [STAThread]
        static void Main(string[] args)
        {
            string ad, mesaj ;

            Console.WriteLine("Adınız nedir? ");
            ad = Console.ReadLine();

            switch (ad)
            {
                case "Nihat":
                    mesaj = "En iyi golcümüz, değil mi?";
                    break;
            }
        }
    }
}
```

```

        case "Arda":
            mesaj = "İstikbalde umut veren bir oyuncu!";
            break;
        case "Servet":
            mesaj = "En iyi savunma oyuncumuz!";
            break;
        case "Rüştü":
            mesaj = "Her iyi kaleci gibi iyi hatalar yapıyor.";
            break;
        default:
            mesaj = "Volkan'a gelince, kafası yerine ellerini
kullandığında harika bir kaleci!";
            break;
    } // switch sonu

    Console.WriteLine();
    Console.WriteLine(mesaj);
    Console.WriteLine();
} // Main sonu
} // class "switch01 sonu
} // namespace sonu

```

İpucu

Bunun için byte, short, int, long ya da char tiplerinden bir seçici değişken kullanılır. Float ve double tipinden seçici kullanılamaz. Seçici değişkenin alacağı sabit değerlere göre program akışı istenen seçeneğe sapar.

Burada herhangi bir deyim-k birden çok deyim içeriyorsa, onlar blok içine alınır ve tek deyimmiş gibi işlenir.

Switch02.cs

```

/*
switch-case yapısında seçici olarak int tipi değişken
kullanılmasına örnek.
*/
using System;

namespace DenetimYapıları
{
    class Switch02
    {
        public static void Main()
        {
            int ay;
            string s;
            Console.WriteLine("Kaçınca ay ? ");
            s = Console.ReadLine();
            ay = Int32.Parse(s);

            switch (ay)
            {
                case 1: Console.WriteLine("Ocak"); break;
                case 2: Console.WriteLine("Şubat"); break;
                case 3: Console.WriteLine("Mart"); break;
                case 4: Console.WriteLine("Nisan"); break;
                case 5: Console.WriteLine("Mayıs"); break;
                case 6: Console.WriteLine("Haziran"); break;
                case 7: Console.WriteLine("Temmuz"); break;
            }
        }
    }
}

```

```

        case 8: Console.WriteLine("Ağustos"); break;
        case 9: Console.WriteLine("Eylül"); break;
        case 10: Console.WriteLine("Ekim"); break;
        case 11: Console.WriteLine("Kasım"); break;
        case 12: Console.WriteLine("Aralık"); break;
    }
}
}
}

```

İpucu

Bu programda 17-inci satırdaki `ay = Int32.Parse(s);` atama deyimi yerine `ay = int.Parse(s);` deyimi konulabilir. Deneyerek görünüz. Bunun nedeni, C# dilindeki `int` tipinin .NET Framework'taki `Int32` tipi ile aynı olmasıdır. Visual Studio, C# dilinin veri tipleri yerine .NET Framework'taki karşılıklarının kullanılmasına izin verir.

Switch03.cs

```

/*
switch-case yapısında seçici olarak char tipi değişken
kullanılmasına örnek.
*/
using System;

namespace DenetimYapıları
{
    class Switch03
    {
        public static void Main()
        {
            char puan;
            string s;
            Console.WriteLine("Karne notunuz nedir?");
            s = Console.ReadLine();
            puan = char.Parse(s);

            switch (puan)
            {
                case 'A': Console.WriteLine("Pekiyi");
                    break;

                case 'B': Console.WriteLine("İyi");
                    break;

                case 'C': Console.WriteLine("Orta");
                    break;

                case 'D': Console.WriteLine("Hmmm...");
                    break;

                case 'F': Console.WriteLine("Daha iyisini
başarabilirsin!");
                    break;

                default: Console.WriteLine("Başarı notun ne?");
                    break;
            }
        }
    }
}

```

```
}  
}
```

Switch04.cs

```
using System;  
  
namespace DenetimYapıları  
{  
    class Switch04  
    {  
        static void Main(string[] args)  
        {  
            string s;  
            int i;  
  
            Console.WriteLine("Not ortalamanızı giriniz : ");  
            s = Console.ReadLine();  
            i = Int32.Parse(s);  
  
            switch (i / 10)  
            {  
                case 9:  
                    Console.WriteLine("Notunuz A dir. "); break;  
                case 8:  
                    Console.WriteLine("Notunuz B+ dir. "); break;  
                case 7:  
                    Console.WriteLine("Notunuz B- dir. "); break;  
                case 6:  
                    Console.WriteLine("Notunuz C dir. "); break;  
                case 5:  
                    Console.WriteLine("Notunuz D dir. "); break;  
                default:  
                    Console.WriteLine("Notunuz F dir. "); break;  
            }  
        }  
    }  
}
```

Aşağıdaki program bir otomatik çikolata makinasının çalışmasını sağlamaktadır. Makinaya 3 boy çikolata konulmaktadır: Küçük, Orta, Büyük. Bunların fiyatları, sırasıyla 250, 500 ve 750 YTL dir. Switch-case yapısını kullanan programda sıçrama goto deyimi ile yapılmaktadır. Programı inceleyiniz ve goto deyimlerinin oynadığı rolü algılayınız.

GoTo01.cs

```
/*  
switch-case yapısında goto sıçramasının kullanılışına örnek.  
*/  
using System;  
  
namespace DenetimYapıları  
{  
    class GotoYapısı
```

```

{
    public static void Main()
    {
        Console.WriteLine("Hangi çikolata : 1=Küçük 2=Orta 3=Büyük");
        Console.Write("Lütfen seçiniz: ");
        string s = Console.ReadLine();
        int n = int.Parse(s);
        int bedel = 0;
        switch (n)
        {
            case 1:
                bedel += 250;
                break;
            case 2:
                bedel += 250;
                goto case 1;
            case 3:
                bedel += 500;
                goto case 1;
            default:
                Console.WriteLine("Geçersiz seçim yaptınız. Lütfen 1, 2, 3 sayılarından birisini seçiniz.");
                break;
        }
        if (bedel != 0)
            Console.WriteLine("Lütfen {0} YTL atınız.", bedel);
        Console.WriteLine("Afiyet olsun! Gene bekleriz!");
    }
}

```

Çıktılar

Program 1, 2, 3 seçenekleri için ayrı ayrı koşturulursa, sırasıyla, aşağıdaki çıktılar görülecektir. Deneyiniz.

Hangi çikolata : 1=Küçük 2=Orta 3=Büyük
 Lütfen seçiniz: 1
 Lütfen 250 YTL atınız.
 Afiyet olsun! Gene bekleriz!

Hangi çikolata : 1=Küçük 2=Orta 3=Büyük
 Lütfen seçiniz: 2
 Lütfen 500 YTL atınız.
 Afiyet olsun! Gene bekleriz!

Hangi çikolata : 1=Küçük 2=Orta 3=Büyük
 Lütfen seçiniz: 3
 Lütfen 750 YTL atınız.
 Afiyet olsun! Gene bekleriz!

Switch05.cs

```
//bir case seçeneğinden default seçeneğine sıçranamaz.
```

```

using System;

namespace DenetimYapıları
{
    class Switch05
    {
        public static void Main()
        {
            string day;
            Console.WriteLine("enter the day :");
            day = Console.ReadLine();
            switch (day)
            {
                case "Mon":
                    Console.WriteLine("day is Mon: go to work");
                    break;
                case "Monday":
                    Console.WriteLine("day is Monday: go to work");
                    break;
                default:
                    Console.WriteLine("default");
            }
        }
    }
}

```

Switch.06.cs

```

using System;
class SwitchTest1
{
    public static void Main()
    {
        int i = 1;
        switch (i)
        {
            case 1:
                Console.WriteLine("one");
                break;
            default:
                Console.WriteLine("default");
        }
    }
}

```