

Bölüm 28

Tarih İşleme ve Yazdırma Biçemleri

DateTime Yapısı
DateTime Yapısının Özgenleri (properties)
DateTime Yapısının Metotları
DateTime Yapısının Operatörleri
DateTime Yapısıyla İlgili Arayüzler (Interfaces)
ToString() Metodu ile Tarih Yazdırma
Yöresellik (locale)
String.Format() Metodu ile Tarih Biçemleme

Bilgisayar uygulamalarında karakterler ve sayılar gibi önem taşıyan başka bir kavram tarih ve zamandır. Yaşantımızın her anı bir şekilde tarih ve zamanla iç içedir. O nedenle, tarih ve zamanla ilgili pek çok işlem yaparız. Tarihleri karşılaştırırız, birbirinden çıkarırız, bir zamana başka bir zaman dilimini ekleriz veya çıkarırız. İleriye veya geriye doğru bir zamanın bileşenlerini bilmek isteriz. Bu istekler, haftanın hangi günü doğduğumuzu doğum tarihimizden çıkarmak gibi basit bir merak olabileceği gibi, bir iş adamının 180 gün sonra ödeyeceği yüklü bir bononun hangi tarihe rasladığını bilmek gibi finansal bir konu da olabilir. Ayrıca, farklı meridyenlerdeki zamanlar farklıdır. Onların birisindeki zamanı bir başkasındaki zamana dönüştürmek gerekir. Farklı ülke ve kültürlerde tarih ve zamanı yazma biçemi farklı olduğu gibi, bir ülkede bile tarih ve zamanı yazmak için birden çok biçem kullanılır. Dolayısıyla, bilgisayar programları bütün bu sorunların üstesinden gelmek zorundadır. C# dili tarih ve zamanla ilgili her işlemi yapabilir, her dil ve kültürdeki farklı biçemlerde tarih yazabilir.

Bu bölümde, bunların nasıl yapıldığını örneklerle anlatacağız.

DateTime Yapısı

.NET Framework Class kütüphanesinde `System` namespace içindeki `DateTime` yapısı (structure) tarih ve zaman ile ilgili bilgileri tutan ve tuttuğu bilgileri istenen biçimde işleyen metotlara sahiptir. Özgenler (property) dediğimiz bu öğeler yıl, ay, gün, saat, dakika, saniye, milisaniye, vb. cinsinden tarih ve zamanı belirlemeye yarar. Metotlar ise, tarih ve zaman ile ilgili her türlü işlemi yapar ve istenen biçimde standart çıkışa gönderir.

DateTime Yapısının Özgenleri (properties)

`Date`, `Day`, `DayOfWeek`, `DayOfYear`, `Hour`, `Kind`, `Millisecond`, `Minute`, `Month`, `Now`, `Second`, `Ticks`, `TimeOfDay`, `Today`, `UtcNow`, `Year`.

Adlarının ima ettiği gibi, bu özgenlerin her birisi bir **an (bir anlık zaman)**'ın belli bir bileşenini tutar. Örneğin, `Date` o anın tarihini, `Day` o anın ayın kaçınıcı günü olduğunu, `Month` o anın hangi ay olduğunu, `Now` şimdiki zamanı, `Today` bu günün tarihini, `Year` o anın hangi yıl olduğunu tutar. `Now` ve `Today`, elbette bilgisayarın gösterdiği tarih ve zamanı tutar. Ayrıca `Now` o anın bütün bileşenlerini tuttuğu için, `DateTime` yapısında olan her öğeye sahiptir.

DateTime Yapısının Metotları

`DateTime` yapısında, tarih ve zaman ile ilgili işlemleri yapan 50 ye yakın metot vardır. Bazılarının işlevlerini söylemekle yetineceğiz. Tarih ve zaman ile ilgili program yazarlar, `msdn` kütüphanesinden bu metotları alabilirler.

`AddDays` metodu bir tarihe belirli sayıda gün eklenince (veya çıkarılınca) yeni tarihin ne olacağını bulur. `Compare` metodu iki tarihi mukayese eder, hangisinin önce olduğunu bulur. `GetDateTimeFormats` metodu tarihi istenen biçimdeki metne dönüştürür. `IsLeapYear` metodu o yılın artık yıl olup olmadığını bulur. `Parse` metodu tarih ve zaman belirten bir stringi `DateTime` eşdeğerine dönüştürür. `ToShortDateString` metodu tarihi kısa biçimli stringe dönüştürür (`Parse` metodunun tersidir). `ToString` metodu tarihi uzun biçimli stringe dönüştürür (`Parse` metodunun tersidir).

DateTime Yapısının Operatörleri

Tarih ve zaman ile ilgili işlemleri yapmaya yarayan sekiz operatör şunlardır: `Addition`, `Equality`, `GreaterThan`, `GreaterThanOrEqual`, `Inequality`, `LessThan`, `LessThanOrEqual`, `Subtraction`. Bunların ne yaptıkları adlarından belli olduğu için, açıklamalarına girmeyecek, ancak gerekli olduklarında kullanacağız.

DateTime Yapısıyla İlgili Arayüzler (Interfaces)

Tarih ve zaman ile ilgili çok sayıda arayüz vardır. Onları burada açıklamaya gerek görmüyoruz.

ToString() Metodu ile Tarih Yazdırma

Hemen her sınıfta var olan `ToString()` metodu `DateTime` yapısının öğelerini standart çıkışa belirtilen biçimde yazar. Bu metodun 'overloaded' olmuş dört farklı şeklini kullanacağız:

```
public string ToString();

public string ToString(IFormatProvider);

public string ToString(string);

public string ToString(string, IFormatProvider);
```

Parametresiz olan birincisi tarihi biçemlemeden yazar. İkincisi `IformatProvider` arayüzünü (interface) parametre olarak alır. Bu arayüz `NumberFormatInfo`, `DateTimeFormatInfo` ve `CultureInfo` sınıfları tarafından çıktığı biçemlemek için kullanılır. Üçüncüsü string tipi parametre alır. Dördüncüsü bir string tipi ve bir `IformatProvider` olmak üzere iki parametre alır. Belirtgen adını alan string tipi parametreler aşağıdaki tabloda gösterilmiştir.

Belirtke	<code>DateTimeFormatInfo</code> özelliği	Pattern değeri (TR için)
t	<code>ShortTimePattern</code>	h:mm tt
d	<code>ShortDatePattern</code>	M/d/yyyy
T	<code>LongTimePattern</code>	h:mm:ss tt
D	<code>LongDatePattern</code>	dddd, MMMM dd, yyyy
f	(combination of D and t)	dddd, MMMM dd, yyyy h:mm tt
F	<code>FullDateTimePattern</code>	dddd, MMMM dd, yyyy h:mm:ss tt
g	(combination of d and t)	M/d/yyyy h:mm tt
G	(combination of d and T)	M/d/yyyy h:mm:ss tt
m, M	<code>MonthDayPattern</code>	MMMM dd
y, Y	<code>YearMonthPattern</code>	MMMM, yyyy
r, R	<code>RFC1123Pattern</code>	ddd, dd MMM yyyy HH':'mm':'ss 'GMT' (*)
s	<code>SortableDateTimePattern</code>	yyyy'-'MM'-'dd'T'HH':'mm':'ss (*)
u	<code>UniversalSortableDateTimePattern</code>	yyyy'-'MM'-'dd HH':'mm':'ss'Z' (*)
		(*) = ülkeden bağımsız

Today özgeninin tuttuğu tarih, `ToString()` metodu tarafından konsola "gün-ay-yıl" biçeminde, "ay-gün-yıl" biçeminde veya istenen başka bir biçemde yazdırılabilir.

Tarih01.cs

```
using System;

namespace Methods
{
    class Tarih01
    {
```

```

        static void Main()
        {
            Console.WriteLine(DateTime.Today.ToString("dd MMMM yy"));
        }
    }
}

```

Çıktı

19 Temmuz 08

Uyarı

```
DateTime.Today.ToString("dd MMMM yy")
```

deyimi yerine

```
DateTime.Now.ToString("dd MMMM yy")
```

deyimini yazarsanız, aynı çıktıyı elde edersiniz. Çünkü Today ve Now özgenlerinin her ikisi de bugün'ün tarih bileşenine sahiptir. Ama Now özgeni daha çok bileşen tutar. Gerçekte o şimdiki zamanı DateTime'in belirlediği bütün bileşenlerine ayırabilir. Örneğin, saat, dakika, saniye, milisaniye bileşenleri Now tarafından tutulur, ama Today tarafından tutulmaz.

Bu günün tarihi yerine başka bir tarih kullanabiliriz. Önce DateTime sınıfından işaretçisi dt olan bir nesne yaratacak ve ona 29-10-1923 değerini vereceğiz. Bu işi yapan deyim şudur:

```
DateTime dt = new DateTime(1923, 10, 29 );
```

dt nin işaret ettiği nesneye (1923,10,29) şeklinde atadığımız tarih o nesne içinde DateTime tipi tarihe dönüşmüştür. Onun üzerinde artık tarih ile ilgili her işlemi yapabilir ve istediğimiz string biçimine dönüştürebiliriz. Örneğin, bu tarihi ToString metodu ile "29 Ekim 1923" biçimine çevirelim.

```
dt.ToString("dd MMMM yy ")
```

Bu söylediklerimizi bir araya getirirsek, aşağıdaki programı elde ederiz.

Tarih02.cs

```

using System;

namespace Methods
{
    class Tarih02
    {
        static void Main()
        {
            DateTime dt = new DateTime(1923, 10, 29);
            Console.WriteLine(dt.ToString("dd MMMM yy "));
        }
    }
}

```

Çıktı

29 Ekim 1923

Yöresellik (locale)

ToString() metodu tarih yazarken string tipi parametre olarak yazılan "dd MMMM yy" biçimine uydu. Ay adını Türkçe yazdı: Temmuz . Peki ama, Türkçe bilmeyen birisi için bu çıktı uygun mu?

.NET bütün ülke ve kültürlerin kullandıkları farklı biçemlerde çıktı verme yeteneğine sahiptir. Kullandığımız Windows işletim sistemi Türkçe'ye ayarlı olduğu için, .NET tarih ve sayı çıktılarını bizim kullandığımız biçeme dönüştürerek veriyor. Windows başka bir dile ayarlı ise, çıktılar o dilin *syntax*'ına uyar. Bu oldukça iyi bir yetenektir. Ama, işletim sistemimiz hangi dilde olursa olsun, çıktının belirli bir yöresel biçimde yazılmasını isteyebiliriz. .NET bunu da yapar.

Bu söylediğimizi yaptırmak için, uygulamamızın çalıştığı thread' in CurrentCulture özgenini istediğimiz ülke ve kültüre ayarlamamız yetecektir. Örneğin, yukarıdaki tarih çıktısını Almanca yazdırmak için

```
Thread.CurrentThread.CurrentCulture = new CultureInfo("de-DE");
```

deyimini programımıza eklememiz yetecektir.

Tarih03.cs

```
using System;
using System.Globalization;
using System.Threading;

namespace Methods
{
    class Tarih03
    {
        static void Main()
        {
            Thread.CurrentThread.CurrentCulture = new CultureInfo("de-DE");
            Console.WriteLine(DateTime.Today.ToString("MMMM dd yyyy"));
        }
    }
}
```

Çıktı

Juli 19 2008

Tarihi İspanya'nın Bask bölgesinde kullanılan biçemde yazdırmak isterseniz, programda aşağıdaki değişikliği yapınız.

```
CultureInfo("eu-ES")
```

çiktının

uztailla 19 2008

olduğunu göreceksiniz.

Aşağıdaki program, aynı işi program koşarken kültür thred'ini değiştirerek tekrarlamaktadır.

Tarih04.cs

```
using System;
using System.Globalization;
using System.Threading;

namespace Yöresellik
{
    class Tarih04
    {
        static void Main(string[] args)
        {
            DateTime d = new DateTime(1923, 10, 29);

            // etkin kültüre (türkçe)
```

```

        Console.WriteLine(d.ToLongDateString());

        // IformatProvider kullanarak Avusturya Almancasına geçiş
        Console.WriteLine(d.ToString("D", new CultureInfo("de-AT")));

        // etkin kültür thread'ini (Türkçe) kullan
        CultureInfo cInf = Thread.CurrentThread.CurrentCulture;
        Console.WriteLine(cInf.ToString() + ": " + d.ToString("D"));

        // IformatProvider kullanarak İspanyol kültür thread'ine
geçiş
        cInf = new CultureInfo("es-ES");
        Thread.CurrentThread.CurrentCulture = cInf;
        Console.WriteLine(cInf.ToString() + ": " + d.ToString("D"));
    }
}

```

Çıktı

29 Ekim 1923 Pazartesi
 Montag, 29. Oktober 1923
 tr-TR: 29 Ekim 1923 Pazartesi
 es-ES: lunes, 29 de octubre de 1923

Bu çıktıyı açıklayalım.

```
DateTime d = new DateTime(1923, 10, 29);
```

deyimi DateTime sınıfının d ile işaret (referans) edilen bir nesnesini yaratmıştır. Bu nesne “29-10-1923” tarihini tutan DateTime tipindedir. (Class'ın soyut bir veri tipi olduğunu anımsayınız.)

İlk satırı yazan

```
Console.WriteLine(d.ToLongDateString());
```

deyimidir. ToLongDateString() metodu, d nin işaret ettiği nesnenin tuttuğu tarihi, çıktının ilk satırında olduğu gibi yazdırır.

İkinci satır çıktısında, kültür thread'i Avusturya Almancasına geçmiştir. Bunu yapan

```
d.ToString("D", new CultureInfo("de-AT"))
```

deyimidir. ToString() metodunun ikinci parametresi olan

```
new CultureInfo("de-AT")
```

deyimi, CultureInfo sınıfının bir nesnesini yaratmakta ve onu Avusturya Almancası threadine ayarlamaktadır. Bu thread, tarihi o yörenin kullandığı biçimde bir string'e dönüştürür. Artık, WriteLine() metodu o stringi konsola göndermekle yükümlüdür.

Üçüncü satır çıktısından önce

```
CultureInfo cInf = Thread.CurrentThread.CurrentCulture;
```

deyimi, threadi etkin olana, yani Türkçe thread'e tekrar döndürmektedir. Dolayısıyla üçüncü satır Türkçe biçimindedir.

Dördüncü satırda ise İspanyolca thread'e geçilmektedir.

Yukarıdaki programda sabit bir tarih kullandık. İstersek, sabit bir tarih yerine bu günün tarihini koyabiliriz. Bunun için Main() metodunun gövdesindeki ilk satır yerine aşağıdaki iki deyimi koymak yetecektir.

```

DateTime d = new DateTime();
d = DateTime.Today;

```

Bunu yapınca, çıktı şuna benzeyecektir. Tabii, çıktındaki tarih, programın çalıştırıldığı tarih olacaktır.

Çıktı

21 Temmuz 2008 Pazartesi
Montag, 21. Juli 2008
tr-TR: 21 Temmuz 2008 Pazartesi
es-ES: lunes, 21 de julio de 2008

Aşağıdaki program, Türkçe tarih biçimleri yazdırma yöntemlerini göstermektedir.

Tarih06.cs

```
using System;

namespace BiçemliÇıktılar
{
    class Tarih06
    {
        static void Main()
        {
            Console.WriteLine(DateTime.Now);
            Console.WriteLine(DateTime.Now.ToString());
            Console.WriteLine(DateTime.Now.ToShortTimeString());
            //Console.WriteLine(DateTime.Now.ToString());
            Console.WriteLine(DateTime.Now.ToLongTimeString());
            Console.WriteLine(DateTime.Now.ToLongDateString());
            Console.WriteLine(DateTime.Now.ToString("d"));
            Console.WriteLine(DateTime.Now.ToString("D"));
            Console.WriteLine(DateTime.Now.ToString("f"));
            Console.WriteLine(DateTime.Now.ToString("F"));
            Console.WriteLine(DateTime.Now.ToString("g"));
            Console.WriteLine(DateTime.Now.ToString("D"));
            Console.WriteLine(DateTime.Now.ToString("m"));
            Console.WriteLine(DateTime.Now.ToString("r"));
            Console.WriteLine(DateTime.Now.ToString("s"));
            Console.WriteLine(DateTime.Now.ToString("t"));
            Console.WriteLine(DateTime.Now.ToString("T"));
            Console.WriteLine(DateTime.Now.ToString("u"));
            Console.WriteLine(DateTime.Now.ToString("U"));
            Console.WriteLine(DateTime.Now.ToString("y"));
            Console.WriteLine(DateTime.Now.ToString("dddd, MMMM dd
yyyy"));
            //Console.WriteLine(DateTime.Now.ToString("ddd, MMM d"
"yy"));
            Console.WriteLine(DateTime.Now.ToString("dddd, MMM dd"));
            Console.WriteLine(DateTime.Now.ToString("M/yy"));
            Console.WriteLine(DateTime.Now.ToString("dd-MM-yy"));
        }
    }
}
```

Çıktı

23.07.2008 00:37:25
23.07.2008 00:37:25
00:37

00:37:25
23 Temmuz 2008 Çarşamba
23.07.2008
23 Temmuz 2008 Çarşamba
23 Temmuz 2008 Çarşamba 00:37
23 Temmuz 2008 Çarşamba 00:37:25
23.07.2008 00:37
23 Temmuz 2008 Çarşamba
23 Temmuz
Wed, 23 Jul 2008 00:37:25 GMT
2008-07-23T00:37:25
00:37
00:37:25
2008-07-23 00:37:25Z
22 Temmuz 2008 Salı 21:37:25
Temmuz 2008
Çarşamba, Temmuz 23 2008
Çarşamba, Tem 23
7.08
23-07-08

String.Format() Metodu ile Tarih Biçemleme

Buraya kadar ToString() metodu ile tarihi string olarak biçemlemeyi öğrendik. Bu kesimde String sınıfının Format() metodunu kullanarak benzer işleri yapacağız.

Tarih07.cs

```
using System;

namespace TarihVeZaman
{
    class Tarih07
    {
        static void Main()
        {
            DateTime dt = new DateTime(2008, 7, 13, 23, 35, 03, 456);

            Console.WriteLine(String.Format("{0:y yy yyy yyyy}", dt));
            Console.WriteLine(String.Format("{0:M MM MMM MMMM}", dt));
            Console.WriteLine(String.Format("{0:d dd ddd dddd}", dt));
            Console.WriteLine(String.Format("{0:h hh H HH}", dt));
            Console.WriteLine(String.Format("{0:m mm}", dt));
            Console.WriteLine(String.Format("{0:s ss}", dt));
            Console.WriteLine(String.Format("{0:f ff fff ffff}", dt));
            Console.WriteLine(String.Format("{0:F FF FFF FFFF}", dt));
            Console.WriteLine(String.Format("{0:t tt}", dt));
            Console.WriteLine(String.Format("{0:z zz zzz}", dt));
        }
    }
}
```


Bu programın çıktısı ve açıklamaları aşağıdaki tabloda verilmiştir. Tabloyu programla satır satır karşılaştırarak istenen biçimin nasıl elde edildiğini görünüz.

Biçemleme	Çıktı	Adı
String.Format("{0:y yy yyy yyyy}", dt);	8 08 008 2008	yıl
String.Format("{0:M MM MMM MMMM}", dt);	7 07 Tem Temmuz	ay
String.Format("{0:d dd ddd dddd}", dt);	13 13 Paz Pazar	gün
String.Format("{0:h hh H HH}", dt);	11 11 23 23	Saat 12/24
String.Format("{0:m mm}", dt);	35 35	dakika
String.Format("{0:s ss}", dt);	3 03	saniye
String.Format("{0:f ff fff ffff}", dt);	4 45 456 4560	sn/1000
String.Format("{0:F FF FFF FFFF}", dt);	4 45 456 456	Sn/1000, sıfırsız
String.Format("{0:t tt}", dt);		A.M. veya P.M.
String.Format("{0:z zz zzz}", dt);	+3 +03 +03:00	Zaman dilimi

String.Format() metodu ile tarih biçemlerken, aşağıda belirtilen standart biçem belirtgenlerini kullanırız.

Tarih08.cs

```
using System;

namespace TarihVeZaman
{
    class TarihYazma02
    {
        static void Main()
        {
            DateTime dt = new DateTime(2008, 7, 13, 23, 35, 03, 456);

            Console.WriteLine(String.Format("{0:t}", dt));
            Console.WriteLine(String.Format("{0:d}", dt));
            Console.WriteLine(String.Format("{0:T}", dt));
            Console.WriteLine(String.Format("{0:D}", dt));
            Console.WriteLine(String.Format("{0:f}", dt));
            Console.WriteLine(String.Format("{0:F}", dt));
            Console.WriteLine(String.Format("{0:g}", dt));
            Console.WriteLine(String.Format("{0:G}", dt));
            Console.WriteLine(String.Format("{0:m}", dt));
            Console.WriteLine(String.Format("{0:y}", dt));
        }
    }
}
```

```

        Console.WriteLine(String.Format("{0:r}", dt));
        Console.WriteLine(String.Format("{0:s}", dt));
        Console.WriteLine(String.Format("{0:u}", dt));
    }
}

```

Çıktı

```

23:35
13.07.2008
23:35:03
13 Temmuz 2008 Pazar
13 Temmuz 2008 Pazar 23:35
13 Temmuz 2008 Pazar 23:35:03
13.07.2008 23:35
13.07.2008 23:35:03
13 Temmuz
Temmuz 2008
Sun, 13 Jul 2008 23:35:03 GMT
2008-07-13T23:35:03
2008-07-13 23:35:03Z

```

Standart Biçem Belirtgenleri

Kod	Çıktısı	.NET adı
String.Format("{0:t}", dt);	23:35	ShortTime
String.Format("{0:d}", dt);	13.07.2008	ShortDate
String.Format("{0:T}", dt);	23:35:03	LongTime
String.Format("{0:D}", dt);	13 Temmuz 2008 Pazar	LongDate
String.Format("{0:f}", dt);	13 Temmuz 2008 Pazar 23:35	LongDate+ShortTime
String.Format("{0:F}", dt);	13 Temmuz 2008 Pazar 23:35:03	FullDateTime
String.Format("{0:g}", dt);	13.07.2008 23:35	ShortDate+ShortTime
String.Format("{0:G}", dt);	13.07.2008 23:35:03	ShortDate+LongTime
String.Format("{0:m}", dt);	13 Temmuz	MonthDay
String.Format("{0:y}", dt);	Temmuz 2008	YearMonth

String.Format("{0:r}", dt);	Sun, 13 Jul 2008 23:35:03 GMT	RFC1123
String.Format("{0:s}", dt);	2008-07-13T23:35:03	SortableDateTime
String.Format("{0:u}", dt);	2008-07-13 23:35:03Z	UniversalSortableDateTime

Tarih09.cs

```
using System;

namespace TarihVeZaman
{
    class TarihYazma02
    {
        static void Main()
        {
            DateTime dt = new DateTime(2008, 7, 13, 23, 35, 03, 456);

            Console.WriteLine(String.Format("{0:M/d/yyyy}", dt));
            Console.WriteLine(String.Format("{0:MM/dd/yyyy}", dt));
            Console.WriteLine(String.Format("{0:ddd, MMM d, yyyy}", dt));
            Console.WriteLine(String.Format("{0:dddd, MMMM d, yyyy}",
dt));
            Console.WriteLine(String.Format("{0:MM/dd/yy}", dt));
            Console.WriteLine(String.Format("{0:MM/dd/yyyy}", dt));
            Console.WriteLine(String.Format("{0:d/M/yyyy HH:mm:ss}",
dt));
            Console.WriteLine(String.Format("{0:d/M/yyyy HH:mm:ss}",
dt));
        }
    }
}
```

Çıktı

7.13.2008
07.13.2008
Paz, Tem 13, 2008
Pazar, Temmuz 13, 2008
07.13.08
07.13.2008
13.7.2008 23:35:03
13.7.2008 23:35:03

sıfır öntakısız gün ve ay yazma

String.Format("{0:M/d/yyyy}", dt);	7.13.2008
String.Format("{0:MM/dd/yyyy}", dt);	07.13.2008

gün ve ay adlarını yazma

<code>String.Format("{0:ddd, MMM d, yyyy}", dt);</code>	Paz, Tem 13, 2008
<code>String.Format("{0:dddd, MMMM d, yyyy}", dt);</code>	Pazar, Temmuz 13, 2008

2 veya 4 haneli yıl yazma

<code>String.Format("{0:MM/dd/yy}", dt);</code>	07.13.08
<code>String.Format("{0:MM/dd/yyyy}", dt);</code>	07.13.2008

gün-ay ayracı ["/" dan "." ya geçer]

<code>String.Format("{0:d/M/yyyy HH:mm:ss}", dt);</code>	13.7.2008 23:35:03	- english (en-US)
<code>String.Format("{0:d/M/yyyy HH:mm:ss}", dt);</code>	13.7.2008 23:35:03	german (de-DE)

Aşağıdaki programı çözümleyiniz. Kodlarla çıktıları karşılaştırınız. Bazı kültürler için çıktıyı elde etmek için System.Globalization etkin kılınmalıdır. Aynı kodlardan farklı İşletim Sistemlerinde farklı çıktılar oluşur.

Tarih10.cs

```
using System;
//using System.Text;
//using System.Globalization;

namespace ex6
{
    class Program
    {
        static void Main(string[] args)
        {
            // Date biçimleri
            DateTime dt = DateTime.Now;
            // Standards
            // O or o yyyy'-MM'-dd'T'HH':'mm':'ss'.fffffffzz
            Console.WriteLine("{0:O}", dt);
            // R or r ddd, dd MMM yyyy HH':'mm':'ss 'GMT'
            Console.WriteLine("{0:R}", dt);
            // s yyyy'-MM'-dd'T'HH':'mm':'ss
            Console.WriteLine("{0:s}", dt);
            // u yyyy'-MM'-dd HH':'mm':'ss'Z'
            Console.WriteLine("{0:u}", dt);

            // date/time belirteçleri
            // short time
            Console.WriteLine("{0:t}", dt);
            // long time
```

```

        Console.WriteLine("{0:T}", dt);
        // short date
        Console.WriteLine("{0:d}", dt);
        // long date
        Console.WriteLine("{0:D}", dt);
        // long date / short time
        Console.WriteLine("{0:f}", dt);
        // long date / long time
        Console.WriteLine("{0:F}", dt);
        // short date / short time
        Console.WriteLine("{0:g}", dt);
        // short date / long time
        Console.WriteLine("{0:G}", dt);
        // Round Trip
        Console.WriteLine("{0:o}", dt);

        // Kültüre göre değişenler
        Console.WriteLine("{0:dd/mm/yyyy HH:MM:ss}", dt);
        Console.WriteLine("{0:mm/dd/yyyy HH:MM:ss}", dt);
        Console.WriteLine("{0:yyyy/mm/dd HH:MM:ss}", dt);

        Console.WriteLine("{0:dd MMM yyyy HH:MM:ss}", dt);
        Console.WriteLine("{0:MMM dd yyyy HH:MM:ss}", dt);
        Console.WriteLine("{0:yyyy MMM dd HH:MM:ss}", dt);
        Console.ReadKey();
    }
}

```

Çıktı

```

2008-08-29T18:19:46.3906250+03:00
Fri, 29 Aug 2008 18:19:46 GMT
2008-08-29T18:19:46
2008-08-29 18:19:46Z
18:19
18:19:46
29.08.2008
29 Ağustos 2008 Cuma
29 Ağustos 2008 Cuma 18:19
29 Ağustos 2008 Cuma 18:19:46
29.08.2008 18:19
29.08.2008 18:19:46
2008-08-29T18:19:46.3906250+03:00
29.19.2008 18:08:46
19.29.2008 18:08:46
2008.19.29 18:08:46
29 Ağu 2008 18:08:46
Ağu 29 2008 18:08:46
2008 Ağu 29 18:08:46

```

KAYNAKLAR

1. Andrew Troelsen: C# and the .NET Platform, Second Edition, Appress, 2007.
2. Borland C#BUILDER, Borland, 2003.
3. Bradley L. Jones: Sams Teach Yourself C# in 21 Days
4. Bruce Eckel: Thinking in C#, Prentice Hall, e-book.
5. Faraz Rasheed: Programmer's Heaven C# School, e-book.
6. <http://flint.cs.yale.edu/cs112/lecture.html>
7. <http://www.academicresourcecenter.net/curriculum/pfv.aspx?ID=6001>
8. <http://www.dotnetngene.kr/NewTech/Temp/Lectures.aspx?>
9. <http://www.ssw.uni-linz.ac.at/Teaching/Lectures/CSharp/Tutorial/>
10. J.Foxall – W.Haro-Chun: Sams Teach Yourself C# in 24 Hours.
11. Joel Murach: Murach's C# 2005, Murach.
12. Joseph Albahari - Ben Albahari: C# 3.0 in a Nutshell, Third Edition, O'Reilly, 2007.
13. msdn: Microsoft Developer Network, <http://msdn.microsoft.com/en-us/default.aspx>
14. Sefer Algan: Her Yönüyle C# (7.Basım), Pusula Yayıncılık, 2008.
15. V.Mukhi- S.Shanbhag- S.Mukhi: C# The Basics, e-book.