İNTERNET TABANLI PROGRAMLAMA- 4.ders

HAZIR FONKSİYONLAR

STRING FONKSİYONLARI (KÜTÜPHANESİ)

Çoğu web sitesinde olduğu gibi kullanıcıdan kullanıcı adını ve şifresini aldıktan sonra bu değerleri veri tabanından elde edilen değerlerle karşılaştırmak bir string işlemidir veyahut kullanıcı adı ve şifresini, yazacağınız sorgu cümlesinin(**SQL**) uygun yerine yerleştirmek yine bir string işlemidir ve çok önemlidir.

Birileri çok önceden bir uygulama yapmış ve uygulama verilerinin bir kısmını, bir **text** dökümanında veya bir **word** dökümanında veya herhangi bir yerde text biçiminde tutmuş.Sizde yeni bir yazılım gerçekleştireceksiniz fakat yazılımıza eski bilgileri de aktarmayı ihmal etmeyeceksiniz.(Her ne kadar eski program bilgileri veri tabanında tutuluyor olsada bazı bilgilerin bir **text** dosyasında veya bir **excel** dosyasında tutulması muhtemeldir ve karşılaşılan bir şeydir).Sizde dosyalara bir bakıyorsunuz ki adamlar kendilerine uygun bir biçimde verileri dosyalara kaydetmişler.Bu durumda size düşen bu text dosyalarından aldığınız verileri ayrıştırıp anlamlı verilere dönüştürmek.Yine bu durumda da yapacağınız işlemler genel itibariyle string işlemleridir.

Sitemizin üyelerine belli aralıklarla mail gönderdiğimizi düşünelim.Tabi mail gönderirken düz yazı değil,bir çerçeve oluşturup,çerçevenin sağ üst kısmına logomuzu koyup ,içeriğinin belli yerlerine veri tabanından çektiğimiz verileri yerleştirdip güzel görünümlü bir yazıyı göndeririz.

Örneğin “Sayın **Hüseyin Akkuş**,Şifreniz:**265901** .....” gibi uzayıp giden bir mail içeriğini hazırlamak için koyu olarak yazılmış verilerin veri tabanından çekilip diğer sabit string ifadelerle birleştirilmesi gerekmektedir.Bu durumda ne tür bir string işlemi gereklidir acaba?

Velhasıl kelam,bu saydıklarımda ve sayılacak daha bir ton şeyde string işlemleri vazgeçilmezdir.

String.Compare() – String ifadeleri karşılaştırmak

```
string str1 = "Dürdane";
string str2 = "Fikriye";

int result = String.Compare(str1, str2);

if (result < 0)
   Console.WriteLine("{0} < {1}", str1, str2);
else if (result > 0)
   Console.WriteLine("{0} > {1}", str1, str2);
else
   Console.WriteLine("{0} = {1}", str1, str2);
```

Yukarıdaki kod parçasında 2 string değer, **String** sınıfının **Compare()** metodu kullanılarak karşılaştırılmaktadır.

String.Compare() metodun, 1. parametre,2.parametreden küçük ise(alfabetik sıralamaya göre) negatif,büyük ise pozitif eşit ise 0 değerini döndürmektedir.

2 string'in eşit olup olmadığını karşılaştırmak için benzer şekilde

```
if (str1 == str2)
   Console.WriteLine("{0} = {1}", str1, str2);
```

karşılaştırmasını kullanabiliriz,fakat hangisinin hangisinden küçük olduğunu bulmak için **<** ve **>** işaretlerini bu karşılaştırmada kullanamayız.

Aşağıdaki kod parçasına bir bakalım;

```
string str1 = "Dürdane";
string str2 = "dürdane";

int result = String.Compare(str1, str2);

if (result < 0)
   Console.WriteLine("{0} < {1}", str1, str2);
else if (result > 0)
   Console.WriteLine("{0} > {1}", str1, str2);
else
   Console.WriteLine("{0} = {1}", str1, str2);
```

2 string ifade de aynıdır fakat 1. String ifadenin ilk harfi büyük olduğundan dolayı,1.string ikincisinden büyük olacak ve Compare metodu 0'dan büyük bir değer döndürecektir.Bazen öyle durumlar olur ki bu 2 string ifadenin eşit olmasını isteyebiliriz yani büyük küçük harf duyarlı olmamasını isteyebiliriz.Bu durumda Compare metodunun 3 parametresi işimizi görecektir.Zira eğer bu parametre **true** olur ise büyük küçük harf olup olmadığı dikkate alınmayacaktır.Aynı kod parçasını şe şekilde işletelim;

```
string str1 = "Dürdane";
string str2 = "dürdane";

int result = String.Compare(str1, str2, true); // Buraya DİKKAT!...

if (result < 0)
   Console.WriteLine("{0} < {1}", str1, str2);
else if (result > 0)
   Console.WriteLine("{0} > {1}", str1, str2);
else
   Console.WriteLine("{0} = {1}", str1, str2);
```

Bu durumda ekran görüntüsü (Dürdane = dürdane) olacaktır.

String.Format() – String biçimlendirmek

Bazen string ifadelerimizi formatlamak(biçimlendirmek) isteyebiliriz.Örneğin elimizde bir tarih varsa bunu anlamlı bir şekilde yazdırmak isteyebiliriz.Veya bir kordinat verilerini anlamlı bir biçimde yazdırmak isteyebiliriz.Şimdi aşağıdaki kod parçacıklarına bakalım;

```
int x = 3,
    y = 4;

// {0} yazan yere 3,
// {1} yazan yere 4 değeri gelecek ve
// coord değeri "3,4" olacaktır.
string coord = String.Format("{0},{1}", x, y);

// 2 string ifade ("Koordinat" ve "3,4") toplanıyor(birleştiriliyor).
Console.WriteLine("Koordinat:" + coord);

DateTime date = new DateTime(2008, 8, 23);
string dateText = String.Format("{0:d}", date);
Console.WriteLine(dateText);
dateText = String.Format("{0:D}", date);
Console.WriteLine(dateText);
```

İlk örnekte gerekli açıklama yapılmıştır.2 örnek için ise öncelikle **DateTime** türünde bir değişken tanımlanmıştır.Ardından **String** sınıfının **Format** fonksiyonu kullanılarak bu tarihe 2 çeşit biçim verilmiştir.İsterseniz yukarıdaki kod parçasının ekran çıktısına bir bakalım;

```
23.08.2008
23 Ağustos 2008 Cumartesi
```

Bu tarz biçimlendirmelerin bazılarını listeyelim;

{0:d}

23.08.2008

{0:D}

23 Ağustos 2008 Cumartesi

{0:f}

23 Ağustos 2008 Cumartesi 13:20

{0:F}

23 Ağustos 2008 Cumartesi 13:20:05

{0:t}

13:20

{0:T}

13:20:05

{0:y}

Ağustos 2008

Bir string ifadenin sol veya sağ yanına boşluk karakteri doldurmak istiyor iseniz yine String.Format() metodunu kullanabilirsiniz.Aşağıdaki örneğe göz atalım;

```
string str;


str = String.Format("-{0,15}-", "ASP.NET");
Console.WriteLine(str);


str = String.Format("-{0,-15}-", "ASP.NET");
Console.WriteLine(str);
```

Bu durumda birincisinde ASP.NET yazısının sol tarafına 15-7=8 tane boşluk karakteri eklenecek, ikincisinde ise sağ tarafına eklenecektir.

Şimdi ekran çıktısına bakalım;

```
-       ASP.NET-
-ASP.NET       -
```

Contains() - EndsWith() – StartsWith()

Contains metodu,bir string ifadenin diğer bir string ifade de içinde geçip geçmediğini bulur,geçiyor ise **true** geçmiyor ise **false** döndürür.

Örneğin,

```
string str1 = "Dürdane";
if(str1.Contains("dane") == true)
```

gibi bir karşılaştırma doğrudur.”Dürdane” kelimesi, “dane” kelimesini içermektedir ve yukarıdaki **if** koşulu true değerini döndürecektir.

Benzer şekilde **EndsWith()** ve **StartsWith()** metodlarıda aldıkları stringi,karşılaştırma yaptıkları stringin başında mı sonunda mı olduğu bilgisini döndürür.

```
string str1 = "Dürdane";
if (str1.StartsWith("Dür") == true)
{
   // "Dürdane" kelimesi "Dür" kelimesiyle başlamaktadır.
   // if bloğu işletilecektir.
```

```
}

if (str1.EndsWith("dane") == true)
{
   // "Dürdane" kelimesi "dane" kelimesiyle sonlanmaktadır.
   // if bloğu işletilecektir.
}
```

Split() – String'i dizi halinde parçalamak

Bir string içerisindeki kelimeleri bazı karakterleri kullanarak ayırmak istiyor isek bu durumda **Split()** metodunu kullanmamız gerekir**."2,3"** koordinat bilgisinden **2** ve **3** sayılarını elde etmek istiyor isek Split() metodu harika metoddur.

Aşağıdaki kod parçacığında x,y,z değerleri string ifadeden parçalanıp elde edilmektedir.

```
// Koordinat bilgisi string olarak tutuluyor.
string coord = "2,3,5";

// Bu komut sonrasında 3 string ifadeden oluşan bir dizi elde edilecektir.
// xyz[0] = "2"
// xyz[1] = "3"
// xyz[2] = "5"
string[] xyz = coord.Split(',');

int x = int.Parse(xyz[0]);
int y = int.Parse(xyz[1]);
int z = int.Parse(xyz[2]);

// Ekran Çıktısı : 2,3,5
Console.WriteLine("{0},{1},{2}", x, y, z);
```

SubString() – String içindeki alt stringleri elde etmek

Bir string ifadenin içinde,4.karakterden başlayıp 10 karakter elde etmek istiyor isek bu durumda kullanacağımız metod **SubString()** metodudur.

Aşağıdaki kod parçalarını inceleyelim;

```
string text = "Visual Studio 2005";
Console.WriteLine(text.Substring(7,4));    // Ekran Çıktısı : "Stud"
Console.WriteLine(text.Substring(7));      // Ekran Çıktısı : "Studio 2005"
```

ToLower() – ToUpper() – ToLowerInvariant() - ToUpperInvariant()

Bir string ifadedeki bütün karakterleri küçük veya bütün karakterleri büyük yapmak istiyor iseniz bu fonksiyonlar işinizi görecektir.

Aşağıdaki kod parçasına ve ekran çıktısına bakalım;

```
string text = "Visual Studio 2005";
Console.WriteLine(text.ToLower());         // visual studio 2005
Console.WriteLine(text.ToLowerInvariant()); // visual studio 2005

Console.WriteLine(text.ToUpper());         // VİSUAL STUDİO 2005
Console.WriteLine(text.ToUpperInvariant()); // VISUAL STUDIO 2005
```

ToLower() ve ToUpper() metodları karakterleri olduğu gibi büyük veya küçük harfe çevirirken, ToLowerInvariant() ve ToUpperInvariant() metodları ise ilgili dile göre değişim göstermektedir. Uygulamayı gerçekleştirdiğim işletim sistemi ingilizce olduğundan dolayı küçük **'i'** karakterleri büyük harfe çevrildiğinde **'I'** haline dönüştürülüyor."**Invariant**" kullanılmayan metodlarda ise **'i'** harfleri olduğu gibi **'İ'** harfine dönüştürülüyor.

Trim() – TrimEnd() – TrimStart() – Boşlukları kaldıran fonksiyonlar

Trim metodları, string içindeki boşluklarla bir derdiniz var ise çok işinize yarayacaktır.

Aşağıdaki kod örneğini ve açıklamaları inceleyelim;

```
string text = "    Visual Studio 2005    ";

/*
* Trim()      : Text'in başındaki ve sonundaki boşlukları kaldırır
* TrimEnd()   : Text'in sonundaki boşlukları kaldırır.
* TrimStart() : Text'in başındaki boşlukları kaldırır.
*
*/
Console.WriteLine("-{0}-", text.Trim());       // -Visual Studio 2005-
Console.WriteLine("-{0}-", text.TrimEnd());    // -    Visual Studio 2005-
Console.WriteLine("-{0}-", text.TrimStart());  // -Visual Studio 2005    -
```

Replace() – Yer değiştirme fonksiyonu

Bir string içindeki bir değeri başka bir değerle değiştirmek istiyor iseniz Replace() metodunu kullanmanız gerekmektedir.

Aşağıdaki kod parçacığını inceleyelim;

```
string text = "Visual Studio 2005";

string text2 = text.Replace("sual", "SORU");
Console.WriteLine(text2);   // "Ekran Çıktısı : ViSORU Studio 2005"
```

StringBuilder() – String inşa eden sınıf

Birden fazla stringi birleştirmek istiyor iseniz **+** ile bunu yapabilirsiniz.(**str4 = str1+str2+str3**).Fakat bu yöntem performans açısından iyi değildir.Onun yerine **StringBuilder** sınıfını kullanmak yazılımınızı daha kaliteli hale getirecektir.

Aşağıdaki kod parçacığını inceleyelim;

```
StringBuilder builder = new StringBuilder();
// .Net dilleri dizi içinde tanımlanıyor
string[] diller = new string[] { "C#", "VB", "C++" };

builder.AppendLine(".Net Dilleri..");
builder.AppendLine();   // Boş bir satır ekleniyor.
for (int i = 0; i < diller.Length; i++)
{
   // Sırasıyla bütün diller yanyana ekleniyor.
   builder.Append(diller[i] + " ");
}
// 0.karakterden başlayarak "-->" ifadesi ekleniyor.
builder.Insert(0, "-->");
// Ekrana ToString() metodu ile yazdırılıyor.
Console.WriteLine(builder.ToString());
```

Makalemin en başında belirttiğim örneklerde birden fazla string'in birleştirilmesiyle ilgili örneklerin hepsi StringBuilder sınıfı kullanılarak yapılmalıdır.

Her ne kadar çözüm yolları çok basit olsa da incelediğimiz metodlar bir yazılımda sık sık kullanabileceğimiz metodlardır.Burada makalemi bitiyorum. Diğer makalelerimde görüşmek dileğiyle,hoşçakalın.

MATEMATİK (Math) FONKSİYONLARI (KÜTÜPHANESİ)

Normal şartlarda kütüphanesi eklenmiş olarak gelir.

```
Math.E;          // e sayısını verir
Math.PI;         // pi sayısını verir
Math.Sin(b);     // b sayısının sin değerini alır
Math.Cos(b);     // b sayısının Cos değerini alır
Math.Tan(b);     // b sayısının Tan değerini alır
Math.Exp(b);     // e^b demektir
Math.Pow(b,c); // b^c demektir
```

Math.Sqrt(b); // Karekök değerini alır daha fazla kök için $a^{(2/3)}$, Math.Pow(a,(2/3))

Math.Ceiling(b); // Ondalık sayıyı üste yuvarlar, b=10.3 , 11 çıkar

Math.Floor(b); // Ondalık sayıyı aşağıya yuvarlar, b=10.3 , 10 çıkar

Math.Round(b); // En yakın tamsayıya yuvarlar, b=10.3 , 10 çıkar, b=10.7 den 11 olur. b=10.49864 sayısı , Dikkat b=10.5 sayısını 10 yuvarlar.

Math.Min(b,c); //b ve c sayısından en küçük sayıyı verir. b=3 , c=4 ise sonuç 3 çıkar

Math.Max(b,c); //iki sayıdan en büyük olanını döndürür.

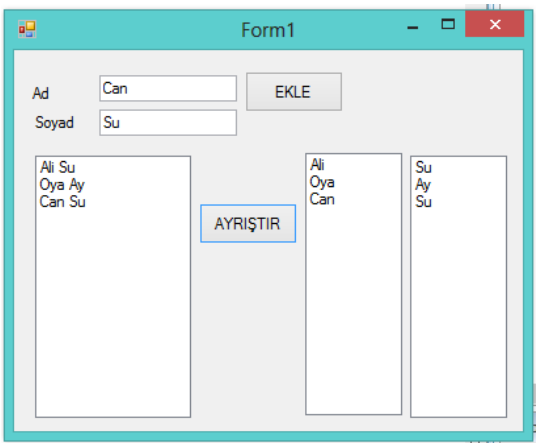
Math.Abs(b); // sayının mutlak değerini alır, yani tüm sayılar pozitif çıkar.

Math.Log10(b); // b sayısının 10 tabana göre logaritmasını alır. b=100 ise sonuç 2 çıkar $b=10^2$ => 2 çıkar.

Math.Log(b); // b sayısının In'ini almaktadır. e tabanına göre logaritmasını alır.

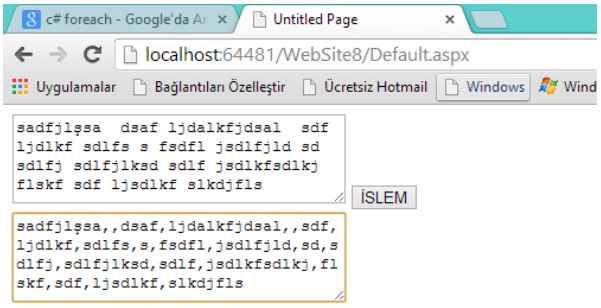
Math.Log(b,c); //c tabanında b sayısının logaritmasını alır. Örneğin b=8 ve c=2 ise sonuç 3 tür.

Örnek




```
private void button1_Click(object sender, EventArgs e)
{
      string Ad, Soyad;
      Ad = textBox1.Text;
      Soyad = textBox2.Text;

      listBox1.Items.Add(Ad + " " + Soyad);

}

private void button2_Click(object sender, EventArgs e)
{
      foreach (string Eleman in listBox1.Items)
      {
            string[] Dizi =Eleman.Split(' ');

            listBox2.Items.Add(Dizi[0]);
            listBox3.Items.Add(Dizi[1]);

      }
}
```

Örnek




```
protected void Button1_Click(object sender, EventArgs e)
{
      string metin1 = TextBox1.Text;

      string [] Dizi = metin1.Split(new Char[] { ' ' });

      foreach (string kelime in Dizi)
      {
            TextBox2.Text = TextBox2.Text + "," + kelime;
      }
}
```

Örnek

```
protected void Button1_Click(object sender, EventArgs e)
{
      double b = 10.49864;
      int c = Convert.ToInt32(Math.Round(b));

      Response.Write(c);

}

Math.Round(b,c);  // b sayısının virgülden sonra C haneye kadar yuvarlatır.
                  b=10.234567 TL sayısını şu şekilde yapmalıyız. b=Math.Round(b,2);
                  şeklinde yazılmalıdır.
protected void Button1_Click(object sender, EventArgs e)
{
      double b = 10.4382323;
      double c = Math.Round(b,2);

      Response.Write(c);
}
```