

T.C.
MİLLÎ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

BİLİŞİM TEKNOLOJİLERİ

ETKİLEŞİMLİ WEB UYGULAMALARI - 2

ANKARA 2008

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	iii
GİRİŞ	1
ÖĞRENME FAALİYETİ - 1	3
1. PROGRAMLAMA DİLİ	3
1.1. VB.NET Programlama Dili.....	3
1.2. Değişkenler (Variables)	4
1.2.1. Tam Sayı Veri Tipleri.....	6
1.2.2. Ondalık Sayı Veri Tipleri	6
1.2.3. String Veri Tipi.....	6
1.2.4. Char Veri Tipi.....	7
1.2.5. Date Veri Tipi.....	7
1.2.6. Boolean Veri Tipi	7
1.3. Değişken Tanımlama (Değişken Bildirimi)	7
1.4. Değişken İsimlendirmede Dikkat Edilecek Noktalar	9
1.5. Değişkenlere Değer Atama	13
1.6. Değişken Tip Dönüşümü	18
1.7. Dizi (Array) Değişkenler	19
1.8. Sabitler (Constants).....	21
1.9. İşleçler (Operatörler).....	21
UYGULAMA FAALİYETİ	24
ÖLÇME VE DEĞERLENDİRME	26
ÖĞRENME FAALİYETİ - 2	27
2. KONTROL DEYİMLERİ	27
2.1. Kontrol Deyimleri	27
2.1.1. If..Then Deyimi	27
2.1.2. If..Then..Else Deyimi.....	35
2.1.3. Else if Deyimi	36
2.1.4. Select Case Deyimi.....	38
UYGULAMA FAALİYETİ	41
ÖLÇME VE DEĞERLENDİRME	42
ÖĞRENME FAALİYETİ - 3	43
3. DÖNGÜ DEYİMLERİ.....	43
3.1. Döngü Deyimleri	43
3.1.1. For..Next Deyimi	44
3.1.2. For Each Deyimi.....	47
3.1.3. While..End While Deyimi	48
3.1.4. Do While..Loop Deyimi	50
3.1.5. Do Until..Loop Deyimi.....	51
UYGULAMA FAALİYETİ	67
ÖLÇME VE DEĞERLENDİRME	68
ÖĞRENME FAALİYETİ - 4	69
4. ALT PROGRAMLAR	69
4.1. Alt Programlar (Subroutines).....	70
4.2. Fonksiyonlar (Functions)	73
UYGULAMA FAALİYETİ	75
ÖLÇME VE DEĞERLENDİRME	76

MODÜL DEĞERLENDİRME	77
CEVAP ANAHTARLARI.....	79
KAYNAKÇA	80
SÖZLÜK	81

AÇIKLAMALAR

KOD	481BK0091
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Web Programcılığı
MODÜLÜN ADI	Etkileşimli Web Uygulamaları - 2
MODÜLÜN TANIMI	Etkileşimli web uygulamaları geliştirirken kullanılan programlama dilinin komut yapısını anlatan öğrenim materyalidir.
SÜRE	40/32
ÖN KOŞUL	Etkileşimli Web Uygulamaları 1 modülünü bitirmiş olmak.
YETERLİK	Programlama dilinin komut yapısını tanıyarak uygulama içinde kullanmak.
MODÜLÜN AMACI	Genel Amaç Gerekli ortam sağlandığında, etkileşimli web uygulamaları geliştirirken kullanılan programlama dilinin komut yapısını tanıyarak uygulama içinde kullanabileceksiniz. Amaçlar 1. Program içinde değişken işlemlerini gerçekleştirebileceksiniz. 2. Kontrol deyimlerini kullanabileceksiniz. 3. Döngü işlemlerini gerçekleştirebileceksiniz. 4. Alt program ve fonksiyonları kullanabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Web programlama yazılımlarını çalıştırabilecek yeterlikte donanıma sahip bilgisayar.
ÖLÇME VE DEĞERLENDİRME	<ul style="list-style-type: none">➤ Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz.➤ Modül sonunda uygulanacak ölçme araçları ile modül uygulamalarında kazandığınız bilgi ve beceriler ölçülerek değerlendirilecektir.

GİRİŞ

Sevgili Öğrenci,

Etkileşimli Web Uygulamaları 1 modülünde web uygulamaları geliştirmek için hangi programları bilgisayarınıza kurmanız gerektiğini ve bir web uygulamasının yapısını öğrendiniz. Bu modülde ise sizlere web uygulamaları hazırlarken kullanacağınız programlama dilinin komut yapısı tanıtılacaktır.

Bu modülde mümkün olduğunca kısa ve öz örnekler verilmeye çalışılmıştır. Örnekleri dikkatlice uygulayıp, sonuçlarını inceleyiniz. Verilen örneklere benzer uygulamalar üretmeye çalışınız. Örnekler üzerinde deneme amaçlı yaptığınız değişikliklerle ve benzer uygulamalar yapmaya çalışmakla modülü iyice pekiştirmiş olacaksınız. Verilen örneklerde yapılan işlemler gerçek uygulamalarda çok daha farklı yöntemlerle, ayrıntılı, güncel ve güvenilir sonuç verecek şekilde yapılıyor olabilir. Bunu da ileride sizler keşfedeceksiniz.

Modülü işlerken konuyu daha iyi anlamak için kendinize sorular sormak ve bu sorulara cevap aramak araştırmacı kişiliğinizi geliştirecektir.

ÖĞRENME FAALİYETİ-1

AMAÇ

Program içinde değişken işlemlerini gerçekleştirebileceksiniz.

ARAŞTIRMA

- Yaşamımızda kullandığımız kimi değerler sabitken, kimi değerler değişkendir. Örneğin, bilimde kullandığımız Pi sayısının değeri 3.14 olarak veya devletimizin adı Türkiye Cumhuriyeti olarak sabit değerlerdir. Fakat hava sıcaklığı bugün 23° iken, yarın 25° olabilir, yani değişebilir. Sizde gündelik yaşantımızdan 5'er tane sabit ve değişken değerlere örnek veriniz. Bulduğunuz örnekleri derste sınıfla paylaşınız. Araştırma sonuçlarınızı öğretmene teslim edecek veya sınıfta sunacak şekilde hazırlayınız.

1. PROGRAMLAMA DİLİ

ASP.NET uygulamaları geliştirmek için birçok programlama dili (C++, C#, Java, Perl, Visual Basic.NET vs.) kullanılabilir. Hangi programlama diliyle uygulamalarınızı yazarsanız yazın, uygulamayı kullanacakların büyük ihtimalle bu seçimden haberi olmayacak, hatta ilgilenmeyecektir. Uygulama kullanıcı için önemli olan, uygulamanın istediği sonucu güvenli, doğru, hızlı bir şekilde üretip üretmediğidir. Etkileşimli Web Uygulamaları modüllerinde programlama dili olarak Visual Basic.NET (VB.NET) kullanılacaktır.

1.1. VB.NET Programlama Dili

Piyasada kullanılan Visual Basic programlama dili .NET platformuyla uyumlu olacak şekilde geliştirilerek VB.NET programlama dili oluşturulmuştur. VB.NET için NET ortamının getirdiği yenilikleri de içeren Visual Basic'in yeni sürümüdür diyebiliriz. Dolayısıyla yapı ve mantık açısından Visual Basic'e benzer. Etkileşimli Web Uygulamaları modüllerinde VB.NET, örnek ASP.NET uygulamaları geliştirmek için gerekli olacak düzeyde anlatılacaktır. İleri düzey ASP.NET uygulamaları geliştirirken ihtiyaç duyulmuş VB.NET'te gerekli araştırmaları yapabilirsiniz.

1.2. Değişkenler (Variables)

Bilgisayarlar, kullanıcılar tarafından kendilerine verilen bilgileri, verileri hafıza adı verilen birimlerinde tutarlar. Hafızada tutulan bilgilerin kimisi sabit değerken, kimi bilgilerin değerleri değişkendir. Bilgisayar hafızası Şekil 1.1'deki gibi düşünülebilir.

HAFIZA	
Hafıza alanı 1 (Hafıza hücresi)	
Hafıza alanı 2	
Hafıza alanı 3	
.....	

Şekil 1.1:Hafıza

Hafıza milyonlarca hücreden oluşur. Bilgisayarda tutulan tüm bilgiler bu hafıza hücrelerinde tutulur. Bilgisayarda bir hafıza hücresiyle diğerini karıştırmamak için hepsine ayrı bir adres verilir. Programcılar da bu hafıza alanlarına, içindeki bilgilerin özelliğine göre anlamlı isimler verirler (Çeşitli isimlendirme kurallarına uyarak). İşte bu anlamlı isimlere değişken denir.

Başka bir ifadeyle, değişken, bilgilerin saklandığı hafıza hücrelere verilen bir isimdir. Değişkenler hafızanın bir bölümünü ifade eder. Programda bir değişken belirlendiğinde (değişken atandığında), bilgisayar bu değişkenin değerinin tutulması için hafızada bir alan, hücre belirler. Değişkenin aldığı tüm değerler bu hücreye yazılır, program içinde o değişken her çağrıldığında bu hücredeki değer getirilir. Konuyu daha iyi anlamak için Şekil 1.2'yi inceleyelim.

	HAFIZADAKİ DEĞER	DEĞİŞKEN İSMİ
Hafıza alanı 1 (Hafıza hücresi)	Barış	Ad
Hafıza alanı 2	Erdem	Soyad
Hafıza alanı 3	Gül	AnneAd
Hafıza alanı 4	Mutlu	BabaAd
Hafıza alanı 5	Hakkari	DogumYeri
Hafıza alanı 6	1978	DogumTarihi
.....

Şekil 1.2:Değişkenler ve hafıza

Program yazarken çeşitli değişkenler kullanılır. Programda hangi değişkenlerin kullanılacağı değişken atama işlemiyle yapılır. Değişken atandıktan sonra ona bir ilk değer verilir, program akışı esnasında değişkenin değeri değiştirilebilir, varolan değeri silinebilir, bir değişken içindeki bir değer başka bir değişkene atanabilir.

Bir değişkenin yaşam döngüsünü aşağıdaki şekilde ifade edebiliriz.

- Değişken atanır.
- Değişkene ilk değeri verilir.

- Değişken çağrılır, gerektiğinde içeriği değiştirilir.

Bilgisayarda tutulacak değerlerin, verilerin özellikleri farklı olduğundan veriler veri tiplerine (türlerine) ayrılmıştır. Örneğin, sayısal değerleri tutmak için ayrı, metin bilgilerini tutmak için ayrı veri tipleri kullanılır. Tam sayılar için **integer** veri tipinin kullanılması uygunken, “Millî Eğitim Bakanlığı” ifadesi için **string** (dize, dizgi) veri tipinin, günün tarihini tutmak için **Date** veri tipinin kullanılması gerekir.

Değişken tanımlama işlemi sırasında değişkene bir isim verilirken, değişkenin hangi tipte veri tutacağı da belirtilir. Benzer veri tipleri arasında işlemler yapılabilir. Örneğin, “integer” veri tipindeki 2 değişken içindeki değerler toplanabilir, fakat içeriği “Ahmet” olan “string” veri tipli bir değişkenle, içeriği 20 olan “integer” veri tipli bir değişken toplanamaz. Sonuç olarak değişken bir değer, veri tutmak için kullanılan belli bir veri tipinde isimlendirilmiş hafıza alanıdır. Tablo 1.1’de VB.NET’teki bazı veri tipleri ve onlarla eşleşen .NET veri tipleri gösterilmiştir.

Veri Tipi	.NET Veri Tipi	Hafızada Kapladığı Alan	Değer Aralığı
Byte	System.Byte	1 Bayt	0 ile 255 arası
Short	System.Int16	2 Bayt	-32768 ile +32767 arası
Integer	System.Int32	4 Bayt	-2.147.483.648 ile +2.147.483.647 arası
Long	System.Int64	8 Bayt	-9,223,372,036,854,775,808 ile +9,223,372,036,854,775,807 arası
Single	System.Single	4 Bayt	-3.4028235*10 ³⁸ ile -1.401298*10 ⁻⁴⁵ arası (negatif değerler için) 1.401298*10 ⁻⁴⁵ ile 3.4028235*10 ³⁸ arası (pozitif değerler için)
Double	System.Double	8 Bayt	-1.79769313486231570*10 ³⁰⁸ ile -4.94065645841246544*10 ⁻³²⁴ arası (negatif değerler için) 4.94065645841246544*10 ⁻³²⁴ ile 1.79769313486231570*10 ³⁰⁸ arası (pozitif değerler için)
Char	System.Char	2 Bayt	0 ile 65535 (0 ile 65535 arasındaki sayısal değerlere karşılık gelen Unicode karakter)
String	System.String	Uygulandığı platforma bağlı	0 ile yaklaşık 2 milyar Unicode karakter
Date	System.DateTime	8 Bayt	1 Ocak 0001, 0:00:00 tarihinden 31 Aralık 9999 11:59:59 tarihine kadar
Boolean	System.Boolean	Uygulandığı platforma bağlı	True (Doğru) veya False (Yanlış)
Object	System.Object	32-bit platformda 4 bayt, 64-bit platformda 8 bayt	Object veri tipi içinde her tip tutulabilir.

Tablo 1.1: Veri tipleri

NOT : Veri tipi aynı zamanda deęişken tipi olarak da isimlendirilir.

Deęişkenler için veri tipleri seçerken dikkat edilecek en önemli nokta, tutulmak istenilen veriye en uygun veri tipini seçmektir. Programların hızlı çalışması açısından belleęi en az kullanılan veri türü seçilmelidir. Örneęin, 0-255 arasındaki tam sayı verileri tutmak için Byte (Bayt), Integer, Long (long) deęişkenleri kullanılabilir. Fakat bu veri türlerinden hafızada en az yer kaplayan Byte olduğundan Byte seçilmelidir.

1.2.1. Tam Sayı Veri Tipleri

Tam sayılar ondalık kısım içermeyen sayılardır. Örneęin, 10, 2450, -45, 245678 sayıları tam sayılardır. Gündelik hayattan tam sayı deęerler içeren durumlar aşağıda sıralanmıştır:

Sınıf mevcudu, okuldaki öğrenci sayısı, bir sınıftaki kız öğrenci sayısı, boy uzunluęu (cm olarak, örneęin 175 cm)

Byte, Short, Integer, Long tam sayı veri tipleridir.

1.2.2. Ondalık Sayı Veri Tipleri

6.8, 3.56, 6.9876 gibi sayılar ondalık sayıları oluşturmaktadır. Yaşantımızda birçok ondalık sayı içeren durumlar vardır. Örneęin;

Boy uzunluęu (m olarak, örneęin 1.75 m), maaş miktarı (930.25 YTL)

Ondalık sayıları yazarken kimi ülkeler “,”(virgöl), kimi ülkelerse “.” (nokta) sembolünü kullanırlar. Single ve Double ondalık sayı veri tipleridir.

1.2.3. String Veri Tipi

String, bir karakter dizisini, grubunu ifade eder. Her türlü karakter grubu bir string oluşturabilir. Örneęin, “Seda”, “Ev Adresiniz”, “235” ifadeleri.

“235” ifadesi tam sayı tipinde mi yoksa bir string tipinde mi diye düşünülebilir? Eęer bir sayı üzerinde matematiksel işlemler yapılmıyorsa o string tipinde tanımlanabilir. Örneęin, telefon numaraları tamamen sayısal ifadelerden oluşur fakat telefon numaraları üzerinde matematiksel işlemler yapmayız. Hiçbir zaman iki telefon numarasının toplanması, çıkarılması ihtiyacı duyulmamıştır. Okul öğrenci numaraları da birer sayı olmalarına rağmen, bu numaralar üzerinde de aritmetiksel işlemler yapılmaz. Bu nedenle telefon numarasını ve öğrenci numarasını tutacak deęişken string tipinde tanımlanabilir. Hangi tipte tanımlandığında hafızada az yer kaplıyorsa o tip seçilmelidir.

Ayrıca sayılardan oluşan bir string tipindeki deęişken ile tam sayı tipindeki bir deęişken arasında aritmetik bir işlem yapılamaz. Çünkü deęişkenlerin veri tipleri birbirinden farklıdır.

1.2.4. Char Veri Tipi

Char (Character) veri tipi, bir karakterlik veri tutulmasını sağlar. Bilgisayardaki her harf, rakam, sembol (a, C, d, 1, 7, ^, \$, [, ?) karakter olarak isimlendirilir.

Bilgisayar dünyasında kelimeler arasındaki boşluklar da karakter olarak değerlendirilir. Boşluk karakteri de diğer karakterler gibi bilgisayarın hafızasında yer kaplar. Char veri tipindeki bir değişkenin içeriğinde ara (boşluk) tuşuna basılarak üretilen boşluk karakteri tutulabilir.

Değişkenin boşluk değerine sahip olması, henüz değerinin olmamasından farklıdır. Eğer bir değişken değer almamışsa o değişkenin içeriği Null'dır. Null ifadesi değeri olmayan, değersiz anlamındadır. Özetle bir değişken henüz değer almamışsa, değersizdir (Null), boşluk değerini almışsa içeriğinde boşluk karakteri vardır, a değerini almışsa değişkenin içeriğinde a karakteri vardır.

1.2.5. Date Veri Tipi

Date veri tipindeki değişkenlerde tarih ve saat değerlerini tutabiliriz. Tarih ve zaman bilgilerinin değişik şekillerde saklanmasını mümkündür. Örneğin, tarih "28/03/1978" şeklinde tanımlanabileceği gibi "28 Mart 1978 Pazartesi" şeklinde de tanımlanabilir. Tarih, zaman bilgilerini string tipinde de tutmak mümkün olmasına rağmen "Date" tipinde tanımlamak işimizi kolaylaştırır. Örneğin Date tipi, iki tarih arasındaki zaman farkının kolaylıkla bulunmasını sağlar.

1.2.6. Boolean Veri Tipi

Gündelik yaşamda doğru-yanlış, açık-kapalı, evet-hayır, 1-0 gibi birbirinin karşıtı olan durumlar vardır. Boolean veri tipi bu tür durumlarda tercih edilir. Boolean veri tipi sadece True (Doğru) ve False (Yanlış) değerlerinden birini alabilir. Örneğin, "onay" isimli bir değişkenin değeri ya "True"dur, ya da "False"dur. Bu değişken iki değere aynı anda sahip olamaz, örneğe göre düşünersek bir işlem ya onaylanmıştır ya onaylanmamıştır.

Bu tipten tanımlanan değişkenlere True, False değerlerinin dışında sayısal değerler de atanabilir. Atanan sayı 0 ise False, değilse True kabul edilir. Örneğin,

onay = 0 ifadesi onay = False ifadesine eşittir.

onay = 1 veya onay = 99 ifadeleri onay = True ifadesine eşittir.

1.3. Değişken Tanımlama (Değişken Bildirimi)

Değişken tanımlama, belirli veri tipinde değerleri tutmak için bilgisayar hafızasında yer/alan tahsis etmektir. Değişken tanımlama işlemi VB.NET'te aşağıdaki şekilde yapılır.

dim (Değişken ismi) [as (Veri tipi)]

Örnekteki tanımlama biçiminde köşeli parantezler, parantez içindeki ifadenin seçime bağlı olduğunu göstermektedir. Örnek değişken tanımlamaları aşağıda verilmiştir.

```
dim ad
dim soyad
dim adres
```

“dim ad” değişken tanımlaması bilgisayara hafızasında “ad” isimli bir değişken alanı açmasını söyler. Bu tanımlamada “ad” değişkeni oluşturulmuş, fakat bu değişkenin hangi tip veriyi tutacağı belirtilmemiştir. Bu kullanım VB.NET’te hatalı bir kullanım değildir, fakat değişken tanımlanırken değişkenin veri tipinin de yazılması programcılık açısından daha yararlıdır.

Değişkenin tipiyle beraber tanımlanmasıyla ilgili örnekler aşağıda gösterilmiştir.

```
dim isim as string
String veri tipinde isim adında bir değişken tanımlanmıştır.
```

```
dim cinsiyet as char
Char veri tipinde cinsiyet adında bir değişken tanımlanmıştır.
```

```
dim miktar as double
Double veri tipinde miktar adında bir değişken tanımlanmıştır.
```

Uygulama: String tipinde evAdresi adında, string tipinde isAdresi adında, byte tipinde not adında değişkenleri tanımlayınız.

Aynı satırda birden fazla değişken tanımlanabilir. Örneğin;

```
Dim as evtel, ceptel, istel as string
```

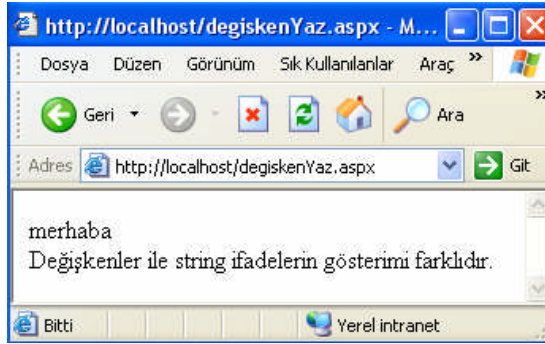
Böylelikle ev, cep ve iş telefonları tutmak için string tipinde 3 farklı değişken tanımlaması yapılmıştır.

Uygulama: Kişinin adını, soyadını, anne adını ve baba adını tutan dört değişkeni, bir satırda tanımlayınız. Değişkenlerin isimleri sırasıyla ad, soyad, anne_ad, baba_ad olsun.

Bir değişkeni ekrana yazdırmak, değişkenin temsil ettiği hafıza bölgesindeki veriyi ekrana yazdırmaktır. Yani ekrana yazdırma komutuyla beraber bir değişken belirttiğimizde, ekrana değişkenin ismi değil içindeki bilgi yazılır. Bir string veriyi görüntülerken string verinin başına ve sonuna eklenen çift tırnak değişken yazdırılırken kullanılmaz. Bu konuyla ilgili aşağıdaki örneği inceleyiniz.

degiskenYaz.aspx

```
1 <%@ page language="VB" %>
2 <script runat="server">
3     sub page_load(obj as object, e as eventargs)
4         dim merhaba as string
5         merhaba="Değişkenler ile string ifadelerin gösterimi
6             farklıdır."
7         response.write("merhaba" & "<br>")
8     end sub
9 </script>
10 <html><body></body></html>
```



Resim 1.1: degiskenYaz.aspx ekran çıktısı

Response.write komutuyla önce **Merhaba** yazdırılmış, ardından **Merhaba** değişkenin içeriği yazdırılmıştır. Değişken içeriği olan "Değişkenler ile string ifadelerin gösterimi farklıdır." ifadesindeki çift tırnaklar sayfada gösterilmemiştir.

1.4. Değişken İsimlendirmede Dikkat Edilecek Noktalar

Program boyunca sürekli değişkenlerle işlem yapıldığından değişken isimlendirme konusuna dikkat etmek gerekir. Değişken isimleri verilirken değişkenlerin tutacakları bilgiye göre isim verilmesi, değişkene anlamlı isimler verilmesi programcıya büyük kolaylık sağlar. Örneğin, kullanıcıların adlarını tutmak amacıyla tanımlanacak bir değişkene **a** ismini vermektense **ad** ismini vermek daha anlamlıdır.

Programda kimi zaman onlarca değişken kullanılır, bu kadar çok değişken olunca, hangi değeri hangi değişkende tuttuğunu hatırlamak zorlaşmaktadır. Bu nedenle anlamlı değişken isimleri vermek uygulama geliştiricisi için vazgeçilmez öneme sahiptir.

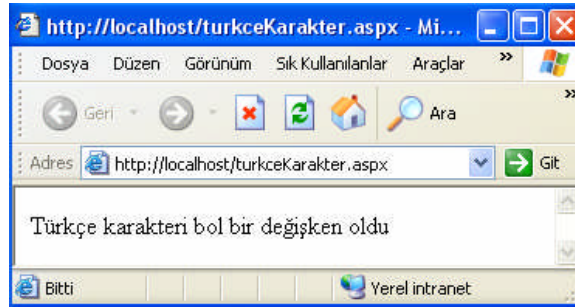
Daha sonra bakıldığında farklı kelimeleri çağrıştıracak nitelikte kısaltmalar kullanılmamalıdır. Standartlaşmış kısaltmaları kullanmak yararlıdır.

Değişken isimlendirme kurallarına uyulduğu sürece değişkene istenilen her isim verilebilir. Değişken isimlendirmeye ilgili değişken isimlendirme standartları vardır. Değişken isimlendirme standartları, değişkenleri program içinde rahat okumak ve takip etmek etmede kolaylık sağlar. Bu değişken isimlendirme standartlarından herhangi birisi seçilerek uygulanabilir.

VB.NET'te değişken isimlerinde Türkçe karakterler kullanılabilir. Fakat karakter uyum sorunlarından dolayı çoğu programcı değişken isimlerinde Türkçe karakter kullanmamayı tercih etmektedir. Değişken isminde Türkçe karakterin kullanıldığı aşağıdaki örneği inceleyelim.

turkceKarakter.aspx

```
1 <script language="VB" runat="server">
2     sub page_load(obj as object, e as eventargs)
3         dim ğüşİöçİ as string
4         ğüşİöçİ = "Türkçe karakteri bol bir değişken oldu"
5         response.write(ğüşİöçİ)
6     end sub
7 </script>
```



Resim 1.2: turkceKarakter.aspx ekran çıktısı

Örnekte **ğüşİöçİ** isimli değişken oluşturulmuştur. **Türkçe karakteri bol bir değişken oldu** ifadesi değişkene atanmıştır. (Değişkenlere değer atamaya ilgili ayrıntılı örnekler daha sonra verilecektir.) Ardından değişken sayfaya yazdırılmıştır.

Aşağıda değişken isimlendirme konusunda çeşitli sınırlamalar, kurallar sıralanmıştır.

Değişken ismi bir harf (A..Z, a..z) veya alt çizgi (_) ile başlamalıdır.

Değişken isimleri bir harf veya alt çizgiyle başlamazsa hatayla karşılarız. Değişkenin ilk karakteri olmadığı sürece değişken isimlerinde rakamlar kullanılabilir.

Doğru verilmiş değişken isimlerine örnekler : ad, soyad, adres, not, isim, kitabevi_ad, parola, ceptel, evtel, ogrenim, mesaj, renk, cinsiyet, adres2, not3, ulke

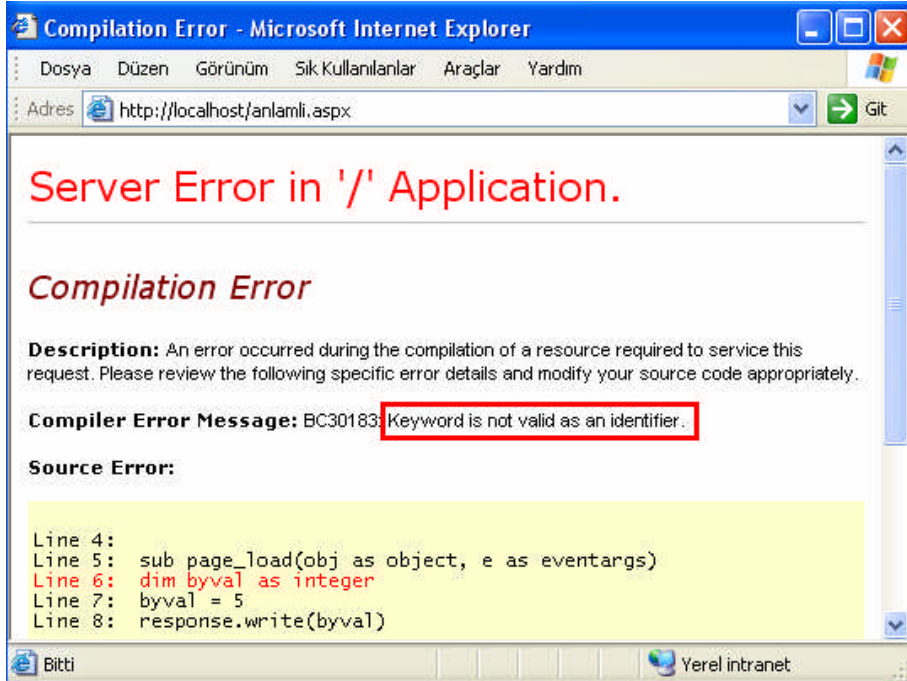
Yanlış verilmiş değişken isimlerine örnekler : 1.not, 2not (Değişken ismi rakamla başlayamaz.)

VB.NET için anlamı olan ifadeler değişken ismi olarak seçilmemelidir.

VB.NET’te özel anlamı, görevi olan kelimeler değişken ismi olarak kullanıldığında hata oluşur. Örneğin, string, cstr, single, cint, cbyte, byval ifadeleri VB.NET tarafından farklı amaçlarla kullanıldığından değişken ismi olarak kullanılamazlar.

anlamli.aspx

```
1 <script language="VB" runat="server">
2     sub page_load(obj as object, e as eventargs)
3         dim byval as integer
4         byval = 5
5         response.write(byval)
6     end sub
7 </script>
```



Resim 1.3: anlamli.aspx ekran çıktısı

Değişken ismi olarak seçilen “byval”, VB.NET’in kullandığı bir ifade olduğundan hata ortaya çıkmıştır.

Değişkenin isminde karakterler arasında, boşluk, bölü işaretleri gibi özel karakterler kullanılmamalıdır.

Kimi programcılar birden fazla kelimeyle oluşturacakları değişken isimlerinde, kelimeler arasında boşluk konulacak yerler için alt çizgi karakteri kullanırlar. Örneğin, yazar_ad, kullanıcı_adi, öğrenim_durum, uye_adi, posta_kodu,

Özellikle birden fazla kelimedenden oluşan değişkenleri isimlendirme de farklı yollar tercih edilir. Örneğin, kullanıcının kimlik numarasını tutmak amacıyla kullanılacak bir değişken aşağıdaki gibi farklı şekillerde isimlendirilebilmektedir.

- **kimliknu** : Numara kelimesi **nu** olarak kısaltılmış ve iki kelime birleşik yazılmıştır. Değişkende kullanılacak kelimeleri birleşik yazmak kimi zaman değişkenin yanlış yazılmasına, rahat okunamamasına neden olur. (Değişkenlerin okunurluğunu azaltır.) Örnekler : babaadi, anaadi.
- **KimlikNu** : Değişkeni oluşturan kelimelerin baş harfleri büyük yazılmıştır. Kimi programcıların değişkeni oluşturan tüm kelimelerin baş harflerini büyük harfle yazırlar. Örnekler : ArkaAlanRengi, OgrenciSayisi, OncekiSoyadi, DogumTarihi, MedeniHal.
- **kimlikNu** : İkinci kelimenin baş harfi büyük yazılmış. Kimi standartlarda değişkeni oluşturan harflerin hepsi küçük yazılır, sadece ilk kelime hariç diğer kelimelerin baş harfi büyük yazılır. Örnekler : kanGrubu, ciltNu, aileSıraNu, sıraNu, seriNu, devamsızGun.
- **kimlik_nu** : Kelimeler arasında boşluk bırakılmayacağından iki kelime arasında **alt çizgi** karakteri kullanılmış. Bazı yazılım geliştiriciler değişkeni oluşturan tüm karakterleri küçük harfle yazıp, kelimeler arasında alt çizgi kullanırlar. Örnekler : arshiv_nu, vergi_kimlik_nu, saglik_karne_nu, hizmet_suresi, staj_durumu, bitis_tarihi.
- **kimlik_numarasi** : Kısaltma kullanılmadan oluşturulmuş değişken ismi. Bu tür uzun değişken isimleri sık kullanıldığında zahmetlidir.
- **KIMLIK_NU, KIMLIK_NU** : Kimi uygulama geliştiriciler değişkeni oluşturan tüm harfleri büyük harflerle yazırlar. OKUL_KODU, BOLUM_KODU, PI_SAYISI.

Değişken ismi 255 karakterden uzun olmamalıdır.

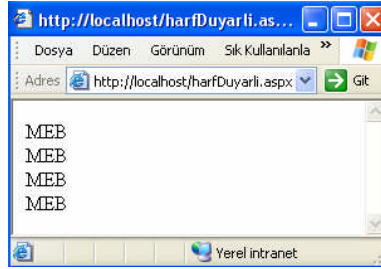
Çoğu zaman ihtiyaç duyulmasa da VB.NET değişken isminde 255 karaktere kadar izin verir. Fakat uygulama içerisinde sık sık kullanılan değişkenleri tekrar tekrar yazarken vakit kaybetmemek ve yazım hatalarından sakınmak için, değişken isimleri mümkün olduğunca kısa ve özlü olmalı, hangi tür bilgiyi içinde tutacaksa o bilgiyi çağrıştırır olmalıdır. Örneğin, kişilere ait T.C. kimlik numaraları tutmak için kullanacağımız bir değişken için isim olarak kimlik_numarasi ifadesini seçmektense kimlik_nu ifadesini seçmek daha uygun olacaktır.

Değişken isimlendirmede bir harfin büyük hâli ile küçük hâli farklı olarak değerlendirilmez.

VB.NET “büyük küçük harfe duyarlı” (case sensitive) bir dil değildir. Yani AD, ad, Ad, aD değişken isimlerinin hepsi aynı değişkeni temsil eder. Bu durumu daha iyi anlayabilmek için aşağıdaki örneğimizi inceleyelim.

harfDuyarli.aspx

```
1 <script language="VB" runat="server">
2     sub page_load(obj as object, e as eventargs)
3         dim ad as string
4         ad = "MEB"
5         response.write(ad &"<br>")
6         response.write(AD &"<br>")
7         response.write(aD &"<br>")
8         response.write(Ad &"<br>")
9     end sub
10 </script>
```



Resim 1.4: harfDuyarli.aspx ekran çıktısı

Örnekte 4 defa “ad” değişkeni ekrana yazdırılmıştır. Fakat “ad” değişkeni, harflerinin küçük büyük halleri kullanılarak her seferinde farklı şekilde (ad, AD, aD, Ad) yazılmıştır. Visual Basic bu değişken yazımlarının hepsini aynı kabul etmiş ve “ad” değişkeninin içeriği olan **MEB** ifadesini ekrana yazdırmıştır.

1.5. Değişkenlere Değer Atama

Değişken tanımlaması yapılarak değişken oluşturulduktan sonra değişkene değer atanabilir. Değişken atama işlemi aşağıdaki şekilde yapılır.

Değişken = Değişkenin tutması istenilen değer

Kullanıcının ismini tutmak amacıyla bir değişken tanımlaması yapalım ve bu değişkene “Burak” ismini verelim.

```
dim ad as string
ad = "Burak"
```

Birinci komut satırında “ad” isimli, “string” tipinde bir değişken tanımlanmıştır. İkinci komut satırında “ad” değişkenine “Burak” değeri atanmıştır. “String” tipindeki değişkenlere atanan değerler (örnekte Burak ismi) çift tırnak içinde yazılırlar.

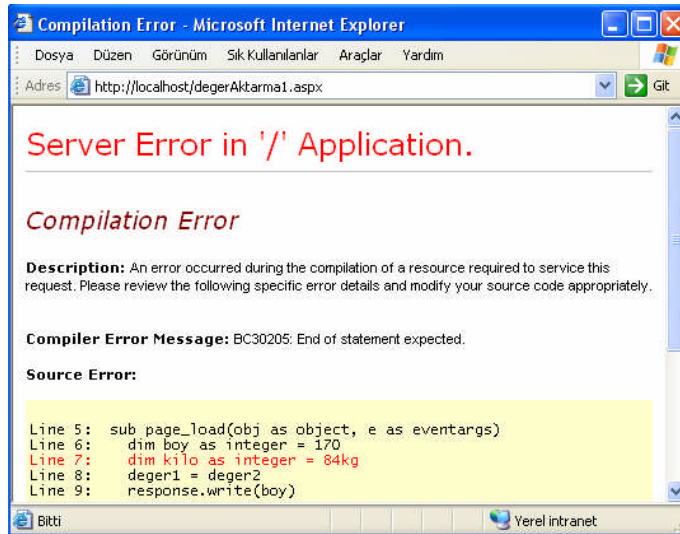
VB.NET değişkene değer atama işleminin değişken tanımlama satırında yapılmasına izin vermektedir. Örneğin, yukarıda iki satır kullanılarak yapılan işlem tek satırda aşağıdaki gibi yapılabilir.

```
dim ad as string = "Burak"
```

Bir değişken için seçtiğimiz veri tipinden farklı tipteki veriyi, o değişkene atamaya çalıştığımızda hatayla karşılaşırız. Nasıl bir hatayla karşılaşacağımızı görmek için aşağıdaki örnek sayfayı inceleyelim.

degerAktarma1.aspx

```
1 <script language="VB" runat="server">
2     sub page_load(obj as object, e as eventargs)
3         dim boy as integer = 170
4         dim kilo as integer = 84kg
5         deger1 = deger2
6         response.write(boy)
7         response.write(kilo)
8     end sub
9 </script>
```



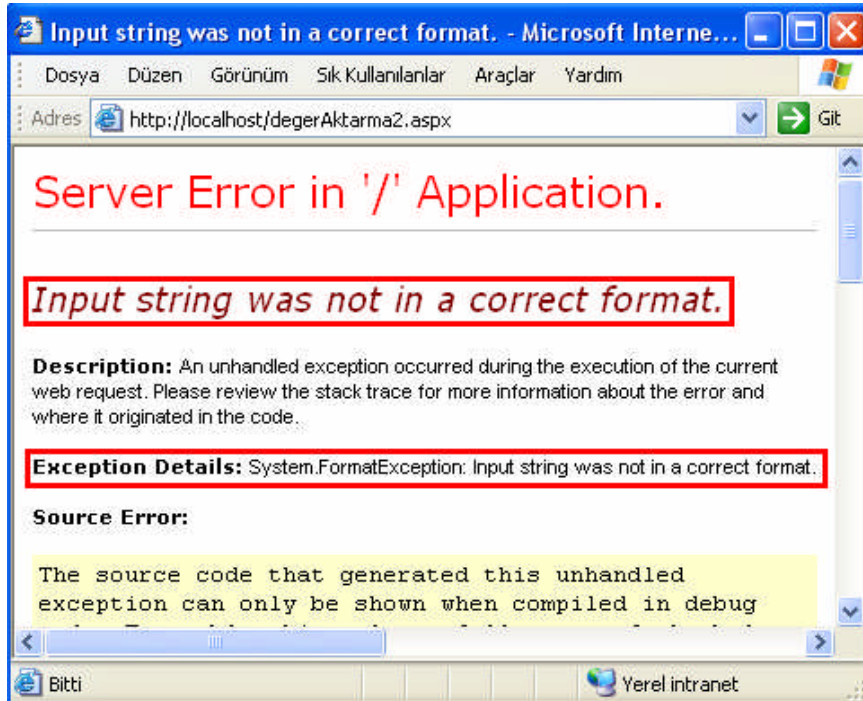
Resim 1.5: degerAktarma1.aspx ekran çıktısı

Örnekte “kilo” değişkeni integer tipinde tanımladığından sadece tam sayı değerler alabilir. Fakat “kilo” değişkenine değer olarak “84 kg” ifadesi aktarılmıştır. “84 kg” ifadesi sayısal bir ifade olmadığından sayfa hata vermiştir.

Değişkenler üzerinde işlemler yaptığımızda değişkenlerin aynı tipte olmasına dikkat etmeliyiz. Örneğin, iki sayısal değeri toplarken, değişkenler aynı sayı tipinde olmalıdır. Bu konuyla ilgili örneği degerAktarma2.aspx adıyla saklayınız.

degerAktarma2.aspx

```
1 <script language="VB" runat="server">
1     sub page_load(obj as object, e as eventargs)
2         dim masraf1 as double = 6.75
3         dim masraf2 as string = "3.25 YTL"
4         response.write(masraf1 + masraf2)
5     end sub
6 </script>
```



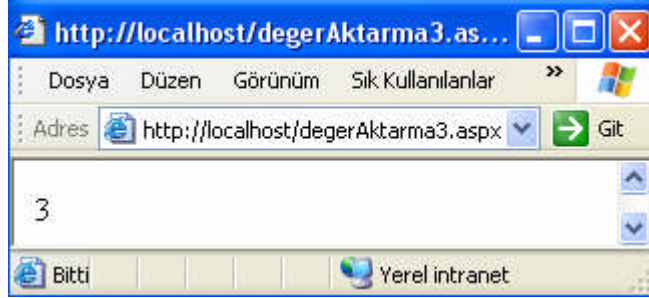
Resim 1.6: degerAktarma2.aspx ekran çıktısı

Örneğimizde double tipinde tanımlanan “masraf1” değişkenine 6.75 sayısı, string tipte tanımlanan “masraf2” değişkenine ise “3.25 YTL” ifadesi atanmıştır. Daha sonra masraf1 ile masraf2’in değerleri toplanarak ekrana yazdırılmaya çalışılmıştır. Fakat double ve string tipleri toplanamayacağından uygulama hata vermiştir.

Farklı tipteki deęişkenler arasında deęer aktarma sırasında kimi zaman deęer kaybı olabilir. Aşağıdaki örneęi deęerAktarma3.aspx ismiyle kaydediniz ve çalıştırınız.

deęerAktarma3.aspx

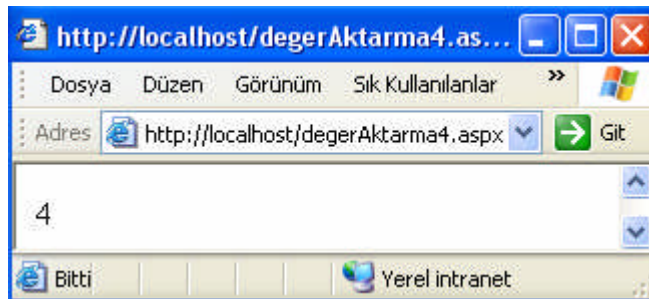
```
1 <script language="VB" runat="server">
2     sub page_load(obj as object, e as eventargs)
3         dim deęer1 as integer
4         dim deęer2 as double = 3.14
5         deęer1 = deęer2
6         response.write(deęer1)
7     end sub
8 </script>
```



Resim 1.7: deęerAktarma3.aspx ekran çıktısı

Örneęimizde integer tipinde “deęer1” deęişkeni ve “double” tipinde “deęer2” deęişkeni tanımlanmıştır. İkinci deęişken olan “deęer2”ye 3.14 deęeri tanımlama esnasında verilmiştir. “deęer2” deęişkeninin 3.14 olan deęeri “deęer1” deęişkenine aktarılmıştır. Ardından “deęer1” ekrana yazdırılmıştır. Aktarma işleminde “deęer1” deęişkeni tam sayı tipinde olduğundan sayıyı 3 olarak tutabilmiştir. Bu nedenle bir deęer kaybı yaşanmıştır.

Yukarıdaki uygulamayı “deęer2” deęişkeninin içerięini 3.78 olarak deęiştirip çalıştırdığımızda aşağıdaki görüntüyü elde ederiz. Sayfayı deęerAktarma4.aspx ismiyle kaydedin.



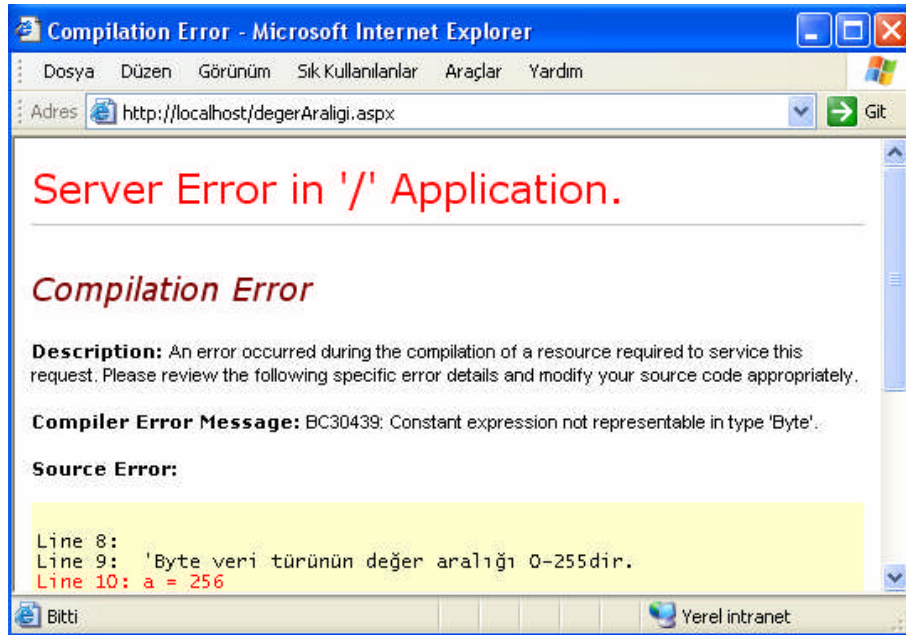
Resim 1.8: deęerAktarma4.aspx ekran çıktısı

Bu seferde 3.78 değeri yuvarlanarak “deger1” değişkeninin içeriği “4” olmuştur. Bu durumda da tutmak istediğimiz değerden farklı bir değer ortaya çıkmıştır.

Eğer bir değişkene tanımlı olduğu veri türünün değer aralığının dışında bir değer atanırsa hata oluşur. Aşağıdaki degerAraligi.aspx sayfası çalıştırıldığında a değişkenine girilen değer, Byte veri tipinin değer aralığının dışında olduğundan hata verecektir.

degerAraligi.aspx

```
1 <script language="VB" runat="server">
2     dim a as byte
3     sub page_load(obj as object, e as eventargs)
4         'Byte veri türünün değer aralığı 0-255'dir.
5         a = 256
6         response.write(a)
7     end sub
8 </script>
```



Resim 1.9: degerAraligi.aspx ekran çıktısı

Uygulama: degerAraligi.aspx sayfasındaki a = 256 ifadesini a = 255 şeklinde değiştirerek sayfayı tekrar çalıştırınız. Sonucu yorumlayınız.

Uygulama: degerAraligi.aspx sayfasındaki a = 256 ifadesini a = -1 şeklinde değiştirerek sayfayı tekrar çalıştırınız. Sonucu yorumlayınız.

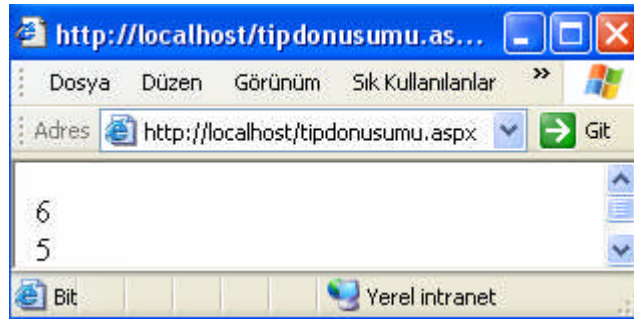
1.6. Değişken Tip Dönüşümü

Kimi zaman farklı tipteki değişkenler arasında işlem yapma gereği ortaya çıkar. Bu durumda, farklı tipteki değişkenlerin aynı tipe dönüştürülmeleri gerekir. Değişkenler, aynı tipe çevrildikten sonra üzerinde işlem yapılır.

Değişkenlerin tiplerini çevirmek için VB.NET'te bulunan fonksiyonlar aşağıda sıralanmıştır. Cbool, Cbyte, Cchar, Cdate, Cdec, Cint, Clng, Cobj, Cshort, Cstr. Tip dönüşümü fonksiyonunun adı fonksiyonun döndürdüğü değer in veri türünü belirler. Örneğin, Cint fonksiyonunun döndüreceği değer integer veri türü, Cstr fonksiyonunun döndüreceği değer string veri türüdür.

tipDonusumu.aspx

```
1 <script language="VB" runat="server">
2     sub page_load(obj as object, e as eventargs)
3         Dim d As Single = 5.5
4         Dim sayi1,sayi2,sayi3 As Integer
5         sayi1 = Cint(d)
6         sayi2 = Int(d)
7         response.write(sayi1 & "<br>")
8         response.write(sayi2)
9     end sub
10 </script>
```



Resim 1.10: tipDonusumu.aspx ekran çıktısı

Sayfada 5.5 değeri Cint fonksiyonu kullanılarak yuvarlama yöntemiyle 6 tam sayısına dönüştürülmüştür. Ardından bu değer integer tipindeki sayi1 değişkenine aktarılmıştır. Int fonksiyonu Cint fonksiyonundan farklı olarak yuvarlama yapmamış, ondalık kısmı atarak sayıyı 5'e dönüştürmüştür.

1.7. Dizi (Array) Değişkenler

Hafızada birden fazla alan tutup, her birine bir numara vererek aynı değişken ismiyle çağırmak amacıyla diziler kullanılır. Dizideki tüm elemanların veri tipi aynıdır.

Kullanım şekli :

Dim (Dizi ismi) (Dizinin eleman sayısı) as (Veri türü)

Örnek :

Dim ad(4) as String

Örnekte **ad** adında 5 elemanlı bir dizi tanımlanmıştır. Diziyi oluşturan elemanları ifade etmek için dizi isminin yanında parantez içerisinde bir sayı belirtilir. Bu sayıya indis denir ve sıfırdan başlar. **ad(0)** ifadesi dizinin 1.elemanını ve alt sınırını, **ad(4)** ifadesi dizinin 5.elemanını ve üst sınırını temsil eder.

ad(0)	Dizinin 1.elemanı
ad(1)	Dizinin 2.elemanı
ad(2)	Dizinin 3.elemanı
ad(3)	Dizinin 4.elemanı
ad(4)	Dizinin 5.elemanı

Tablo 1.2:Ad dizisi

Dikkat ederseniz 5 elemanlı bir dizi oluştururken sıfır indisli elemanı dikkate alarak diziyi ad (4) şeklinde tanımladık. Şimdi de dizinin elemanlarına değer atayalım.

ad(0) = "Mehmet"

ad(1) = "Buse"

ad(2) = "Başak"

ad(3) = "Burak"

ad(4) = "Eser"

Uygulama: 18 elemanlı string veri türünde, 1.lig futbol takımlarının isimlerinin tutulduğu **futbolTakim** isminde bir dizi tanımlayınız. Ardından birkaç elemanına değer atayınız.

İstenirse dizi tanımlanırken dizi elemanlarına değerleri aktarılabilir.

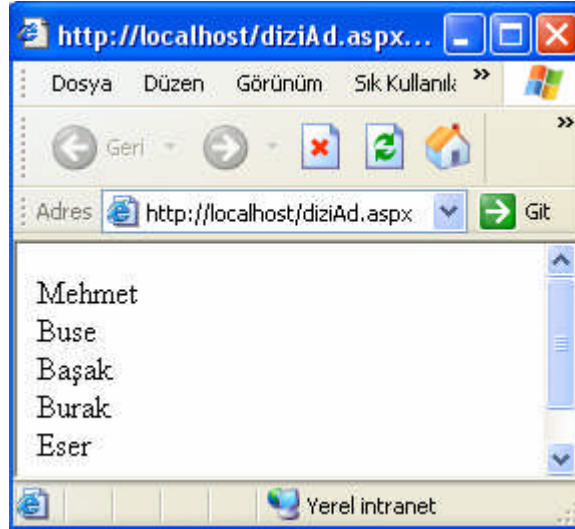
Dim ad () as string = {"Mehmet", "Buse", "Başak", "Burak", "Eser"}

Uygulama: 5 elemanlı string veri tipinde, renk isimlerinin tutulduğu **renkler** isminde dizi tanımlayarak, dizi elemanlarının değerlerini atama sırasında veriniz.

Dizi değişkeni olarak "ad" dizisinin tanımlandığı, elemanlarına değer atandığı ve elemanlarının sayfada görüntülediği diziAd.aspx adlı örneği inceleyiniz.

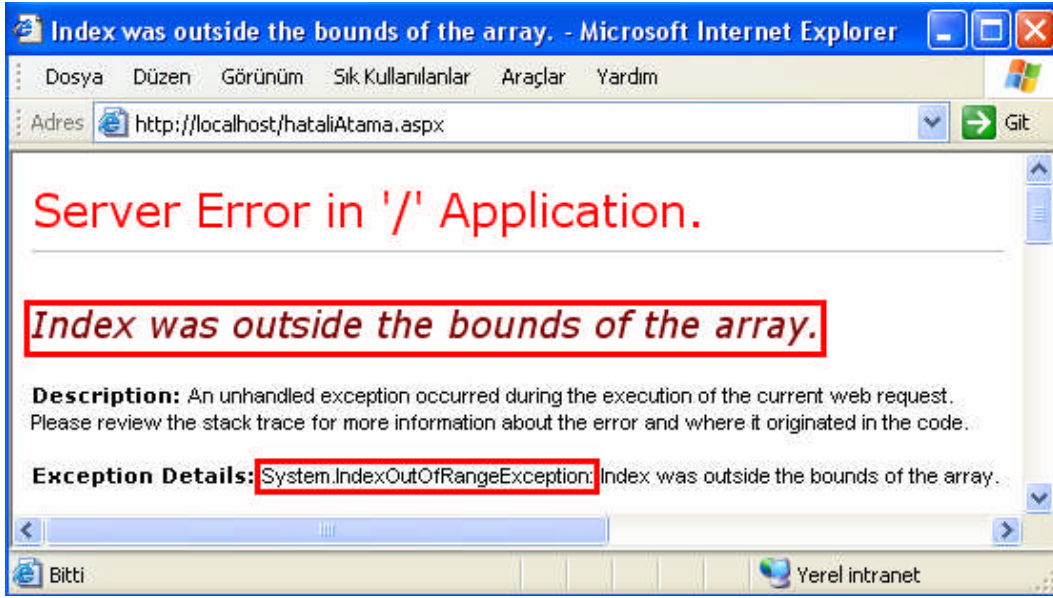
diziAd.aspx

```
1 <script language="VB" runat="server">
2     sub Page_Load(sender as object, e as eventargs)
3         dim ad(4) as string
4         ad(0) = "Mehmet"
5         ad(1) = "Buse"
6         ad(2) = "Başak"
7         ad(3) = "Burak"
8         ad(4) = "Eser"
9         response.write(ad(0) & "<br>")
10        response.write(ad(1) & "<br>")
11        response.write(ad(2) & "<br>")
12        response.write(ad(3) & "<br>")
13        response.write(ad(4) & "<br>")
14    end sub
15 </script>
```



Resim 1.11: diziAd.aspx ekran çıktısı

diziAd.aspx sayfasının ad(4) = "Eser" satırının altına ad (5) = "Efe Onur" satırını ekleyip sayfayı hataliAtama.aspx adıyla kaydediniz. Dizide ad (5) elemanı olmadığından sayfa **Index was outside the bounds of the array** (İndis dizi sınırları dışındadır) hatasını verecektir.



Resim 1.12: hataliAtama.aspx ekran çıktısı

1.8. Sabitler (Constants)

Uygulama boyunca değeri sabit kalan ifadeleri tutmak amacıyla sabitler tanımlanır. Pi sayısı, seçmen yaşı, vergi oranı sabit değerlere örnek olarak verilebilir. Sabitler aşağıdaki şekilde tanımlanır.

Const (Sabit ismi) as (Veri tipi)

Örnekler:

```
const pi as single = 3.14
const vergi as integer = 8
const secmen yas as single = 18
```

1.9. İşleçler (Operatörler)

İşlemleri gerçekleştirmek için kullanılan işaretler, sembollerdir. Örneğin, iki sayıyı toplamak için "+", bir sayıdan diğer sayıyı çıkartmak için "-" sembolünü kullanırız. VB.NET'te sıkça kullanılan işleçler Tablo 1.3'te gösterilmiştir.

İşlev (Fonksiyon)	İşleç (Operatör)
Toplama	+
Çıkarma	-
Çarpma	*
Bölme	/
Üs alma ($3^2 = 9$)	^
Eşittir	=
Eşit değildir (farklıdır)	< >
Küçüktür	<
Büyüktür	>
Küçük eşittir	<=
Büyük eşittir	>=
Mod alma ($9 \text{ mod } 3 = 0$)	Mod
Karakter toplama, ekleme ("TE" + "MA" = "TEMA")	+ veya & (ampersand)
Mantıksal (Lojik) olumsuz	Not
Mantıksal ve	And
Mantıksal veya	Or

Tablo 1.3: İşlev ve işleçler

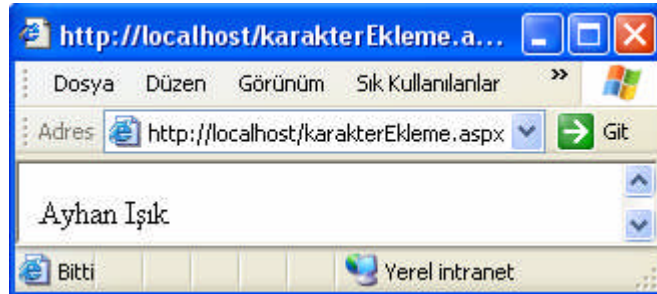
Aşağıda karakter toplama işleciyle ilgili bir örnek verilmiştir. Bu örneği karakterEkleme.aspx dosya adıyla kaydediniz.

karakterEkleme.aspx

```

1 <script language="VB" runat="server">
2     dim ad as string="Ayhan"
3     dim soyad as string="Işık"
4     sub page_load(obj as object, e as eventargs)
5         response.write(ad + " " + soyad)
6     end sub
7 </script>

```



Resim 1.13: karakterEkleme.aspx ekran çıktısı

Matematikte var olan işlem önceliđi program yazarken de geerlidir. Aritmetik işlemlerdeki öncelik sıralaması ařađıda belirtilmiřtir.

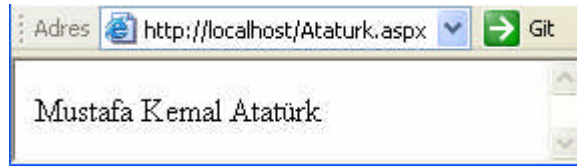
- 1.Üs alma
- 2.arpma ve bölme (Eřit önceliklidir.)
- 3.Toplama ve ıkarma (Eřit önceliklidir.)

Diđerlerinden önce yapılması istenen işlem parantez iine alınmalıdır. Örneđin, $5*6+3=33$ işleminde toplama işleminin önce yapılması isteniyorsa $5*(6+3)=45$ şeklinde parantez iine alınır.

Eřit öncelikli olan işlemlerde, hangi işlemin önce yapıldıđının önemi yoktur, yani işlem sonucunu etkilemez. Örneđin, $8+5-3$ işleminde önce 8 ile 5'in toplanıp, toplamdan 3'ün ıkarılması ile 5'den 3'ün ıkarılması ve sonuçla 8'in toplanması aynı sonucu verir.

UYGULAMA FAALİYETİ

- Aşağıdaki işlemleri gerçekleştiren **Ataturk.aspx** isimli ASP.NET sayfası oluşturunuz. Sayfanızın tarayıcıdaki görüntüsü aşağıdaki resme benzer olacaktır.
 - Sayfa yüklendiğinde ad ve soyad isimli iki değişken tanımlayıp ad değişkenine **Mustafa Kemal**, soyad değişkenine **Atatürk** değerlerini atayınız.
 - Bu iki metni birleştirip Label kontrolünde tarayıcıda görüntüleyiniz.



İşlem Basamakları	Öneriler
➤ Sayfada kullanılacak programlama dilini ifade eden bildirim yazınız.	
➤ Web formunu oluşturunuz.	➤ Formun sunucu kontrolü olduğunu belirten runat="server" ifadesini eklemeyi unutmayınız. Etiket kapatmayı unutmayınız.
➤ Etiket kontrolünü tanımlayınız.	➤
➤ Alt programı oluşturunuz.	➤ Program kodları <script> blokları arasına yazılmalıdır. Sayfa yüklendiğinde yapılacak işlemlerin yazıldığı alt programın isminin Page_Load olarak verilmesi gerektiğine dikkat ediniz. Alt programı sonlandırmayı unutmayınız.
➤ Değişkenleri tanımlayıp ilk değerlerini veriniz.	➤ Değişkenleri isterseniz alt programın dışında da tanımlayabilirsiniz. Tüm alt programların dışında tanımlanan değişkenler bütün alt programlar tarafından kullanılabilir. ➤ Belirli bir alt program içinde tanımlanan değişkenler sadece o alt program tarafından kullanılabilir ve diğer alt programlar tarafından tanınmazlar.
➤ Değişkenlerdeki metni birleştirerek etiket kontrolüne atayan komutu yazınız.	➤ İsterseniz değişkenlerdeki metni başka bir değişkene atayıp ardından bu değişkeni label kontrolüne atayabilirsiniz.

Verilmiş olan uygulama faaliyetini örnek olarak;

- Bir kitabevinin internet sitesini yapacağınızı ve sitede şu bilgileri bulduracağınızı farz edelim: Kitap adı, kitabın türü (roman, hikaye, şiir, deneme vs), yazar adı, fiyatı, basım tarihi, yayınevi adı. Bu bilgileri tutmak için kullanılacak uygun değişken isimleri ve veri türlerini yazınız.
- Bir çiçekçinin internet sitesi için kullanılacak uygun değişken isimleri ve veri türlerini yazınız.

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TEST (ÖLÇME SORULARI)

Aşağıdaki soruları dikkatlice okuyarak uygun cevap şıkkını işaretleyiniz.

1. Aşağıdakilerden hangisi doğru bir değişken ismidir?
A) l.ad
B) ogrenci ismi
C) UrunFiyat
D) 5_bolum
2. Bir sınavdan alınabilecek (0 ile 100 arasındaki) puanları tutmak için kullanılacak en uygun veri türü aşağıdakilerden hangisidir?
A) String
B) Char
C) Byte
D) Integer
3. Aşağıdaki bilgilerden hangisi yanlıştır?
A) Visual Basic.Net büyük küçük harfe duyarlı bir dil değildir.
B) Visual Basic.Net'te "AD" değişken ismi ile "ad" değişken ismi farklı değişken isimleri olarak algılanır.
C) Değişken, verileri tutmak amacıyla ayrılmış hafıza alanlarına verilen isimdir.
D) Char, tek karakterlik verileri tutmak için kullanılan veri tipidir.
4. Tarih, saat gibi verileri tutmak için kullanılan veri tipi nedir?
A) Double
B) Boolean
C) String
D) Date
5. Aşağıdakilerden hangisi bir kurumdaki personel sayısını tutmak için kullanılacak uygun veri türlerinden biri değildir?
A) Single
B) Integer
C) Short
D) Byte
6. Aşağıdaki bilgilerden hangisi doğrudur?
A) Her veri türünün bir değer aralığı vardır.
B) 2670 sayısı byte veri türünde tanımlanmış bir değişkende tutulabilir.
C) 5.23 sayısı integer veri türüyle tutulabilecek bir değerdir.
D) String, tam sayıları tutmak amacıyla kullanılan bir veri türüdür.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları geri dönerek tekrar inceleyiniz. Tüm sorulara doğru cevap verdiyseniz diğer modüle geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Kontrol deyimlerinin kullanabileceksiniz.

ARAŞTIRMA

- Gündelik hayatınızdan bir şarta bağlı olarak yaptığınız eylemleri düşününüz ve 5 adet örnek veriniz. Örneğin, kendimi yorgun hissedersen dinlenirim, yorgun hissetmezsem işime devam ederim. Burada şart yorgun hissetme durumudur. Araştırma sonuçlarınızı öğretmene teslim edecek veya sınıfta sunacak şekilde hazırlayınız.

2. KONTROL DEYİMLERİ

Gündelik hayatımızda içinde bulunduğumuz şartlara göre farklı kararlar veririz. Örneğin, evden dışarı çıkarken hava durumuna göre nasıl dışarı çıkacağımıza karar veririz. Havanın bulutlu veya güneşli olmasına göre dışarı çıkar, gerekli görürsek yanımıza şemsiye, mont alırız. Gündelik yaşamdaki bu mantık uygulama geliştirme sürecinde de geçerlidir.

2.1. Kontrol Deyimleri

Programcılıkta bir şarta bağlı olarak, şartın gerçekleşmesi durumunda bazı komutlar şartın gerçekleşmemesi durumunda başka komutlar çalıştırılır.

VB.NET'te kontrol deyimleri aşağıda sıralanmıştır:

- If..then
- If..then..else
- Else..if
- Select case

2.1.1. If..Then Deyimi

If komutunun kullanım şekli aşağıdaki gibidir:

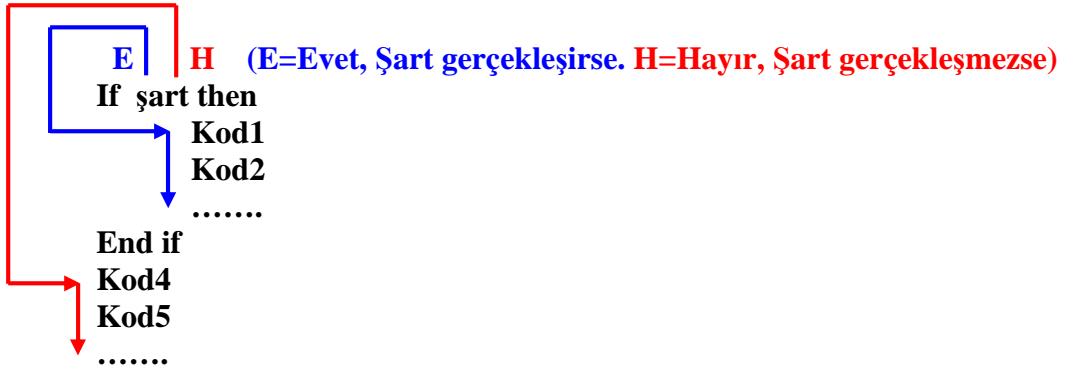
```
if şart then komut
```

Şayet if...then deyimiyle birden fazla komut kullanılacaksa aşağıdaki yazım biçimi kullanılır.

```
if şart then
komut1
komut2
.....
end if
```

Kullanım şeklindeki “if” kelimesi Türkçe’de “eğer”, “then” kelimesi “ise”, “end” kelimesi ise “sonlandırmak” anlamına gelir. Komutun çalışmasını genel olarak şu şekilde ifade edebiliriz: Eğer (if) şart gerçekleşirse ise (then) komutları çalıştır. Komutların çalıştırılmasından sonra if deyiminin çalışmasını da sonlandır (end if).

If komutunun kullanımında görülen şart için iki durum söz konusudur. Şart ya gerçekleşir ya da gerçekleşmez (Şartın belirttiği ifade ya doğrudur ya da yanlıştır). Belirtilen şartın gerçekleşmesi durumunda “then” kelimesi ile “end if” kelimeleri arasındaki komutlar çalıştırılır. Bu komutların çalıştırılmasından sonra “end if ”den sonraki komutlar çalışmaya devam eder. Eğer şart gerçekleşmezse, program “then” kelimesi ile “end if ” kelimeleri arasındaki komutları atlayarak “end if ”den sonraki komutları çalıştırmaya devam eder.



Şekil 2.1: If komutunun çalışma mantığı

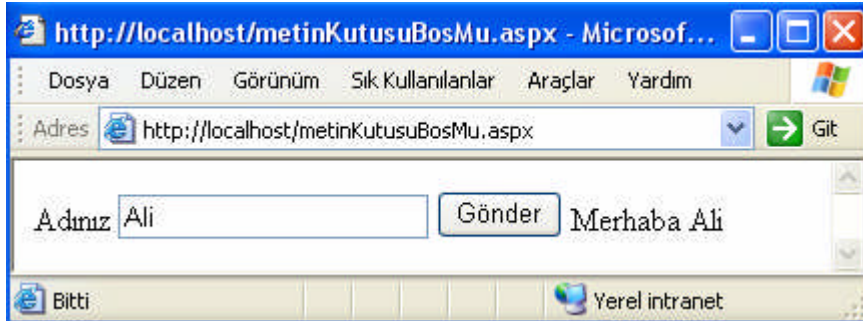
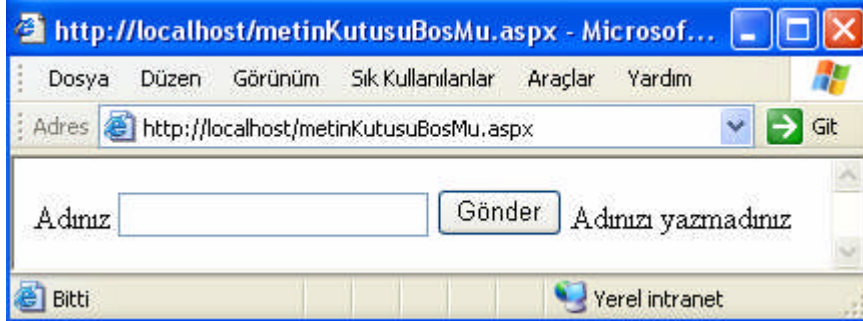
If komutunu daha anlaşılır kılmak için gündelik hayattan örnekler verelim.

```
if hava = bulutlu then şemsiyeni al
if bilgisayar = bozuk then tamirciye götür
if oda = havasız then pencereyi aç
if ortalama >= 4.50 then öğrenciye takdirname ver
if trafik ışığı = kırmızı then Dur
```

If komutunun kullanımıyla ilgili bir örnek uygulama yapalım.

metinKutusuBosMu.aspx

```
1 <script language="VB" runat="server">
2     sub gonderme(sender as object, e as EventArgs)
3         if textbox1.text="" then
4             label2.text="Adınızı yazmadınız"
5         end if
6         if textbox1.text<>"" then
7             label2.text="Merhaba " & textbox1.text
8         end if
9     end sub
10 </script>
11 <html><body>
12 <form runat="server">
13     <asp:label id="label1" text="Adınız" runat="server"/>
14     <asp:textbox id="textbox1" runat="server"/>
15     <asp:button id="button1" text="Gönder" onclick="gonderme"
16         runat="server"/>
17     <asp:label id="label2" runat="server"/>
18 </form></body></html>
```

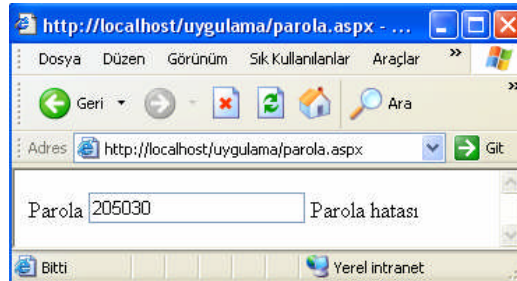
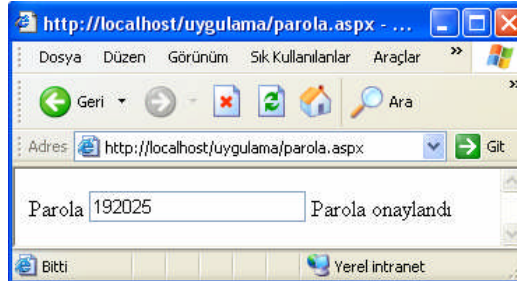


Resim 2.1: metinKutusuBosMu.aspx ekran çıktısı

Örnekte if deyimiyle metin kutusunun iinin boř olup olmadıęı kontrol edilerek uygun mesaj ekrana yazdırılmıřtır.

parola.aspx

```
1 <script language="VB" runat="server">
2     dim parola as string = "192025"
3     sub onaylama(sender as object, e as eventargs)
4         if textbox1.text = parola then
5             label2.text="Parola onaylandı"
6         end if
7         if textbox1.text <> parola then
8             label2.text = "Parola hatası"
9         end if
10    end sub
11 </script>
12 <html><body>
13 <form runat="server">
14     <asp:label id="label1" text="Parola" runat="server" />
15     <asp:textbox id="textbox1" ontextchanged="onaylama"
16         runat="server" />
17     <asp:label id="label2" runat="server" />
18 </form></body></html>
```



Resim 2.2: parola.aspx ekran ıktısı

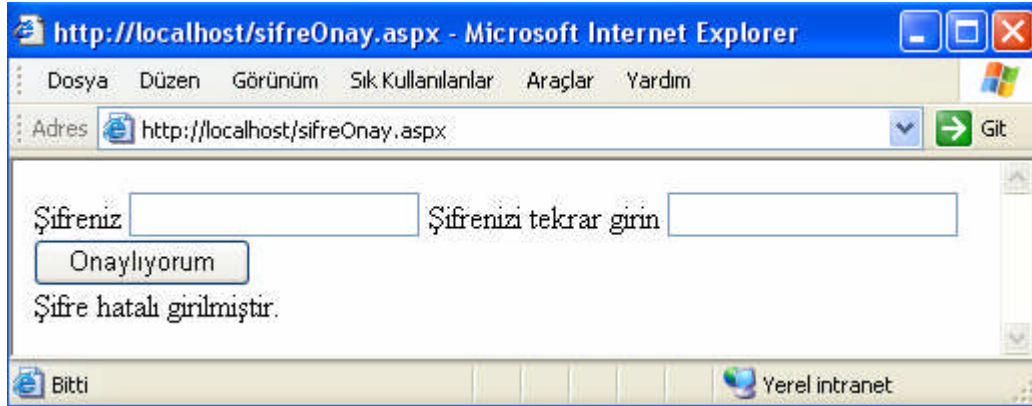
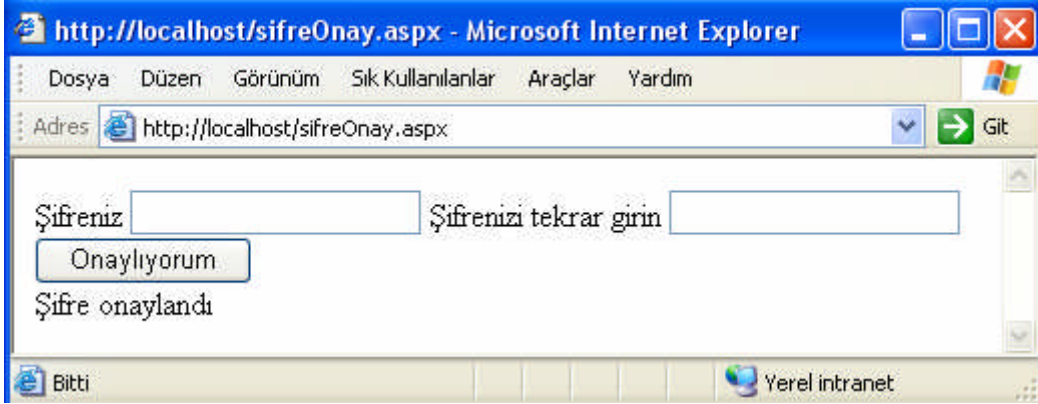
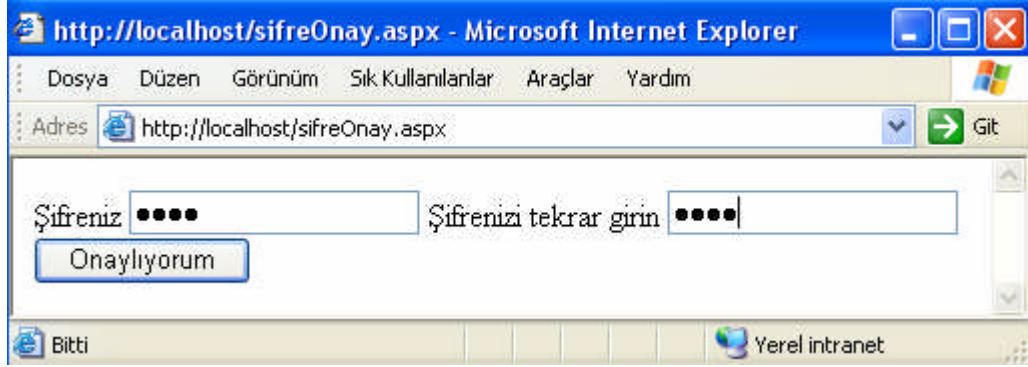
Bu uygulamada kullanıcıdan parolasını girmesi istenmiştir. Eğer parola doğruysa “Parola onaylandı” mesajı, parola yanlışsa “Parola hatası” mesajı görüntülenmiştir.

Yukarıdaki uygulamada metin kutusuna girilen değerle (textbox1.text) parola değişkeni içindeki değer karşılaştırılmış ve bu karşılaştırma sonucuna göre işlemler gerçekleştirilmiştir. Parola onaylama işlemi gerçekleştirildikten sonra metin kutusundaki değere artık ihtiyaç yoktur.

İnternette eposta hesabı alırken veya bir siteye üye olurken kullanıcı adı belirledikten sonra bir şifre girmemiz istenir. Şifrenin gizliliği açısından şifreyi oluşturan karakterler * veya • karakterleriyle gösterilir. Şifre belirlerken yazdığımız karakterleri göremediğimizden, şifrenin doğru yazıldığından emin olmak için şifreyi tekrar yazmamız istenir. Eğer ilk girdiğimiz şifreyle ikinci girdiğimiz şifre aynıysa, şifreyi doğru olarak girmişizdir. Fakat ikinci girdiğimiz şifre birincisinden farklıysa o zaman şifreyi yanlış girdiğimizden şifremiz onaylanmaz. Bu durumla ilgili aşağıdaki örneğimizi inceleyelim.

sifreOnay.aspx

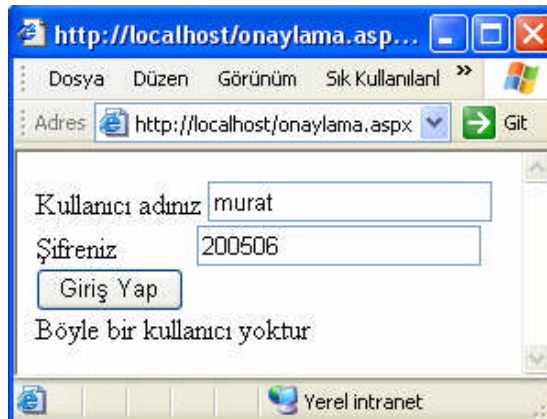
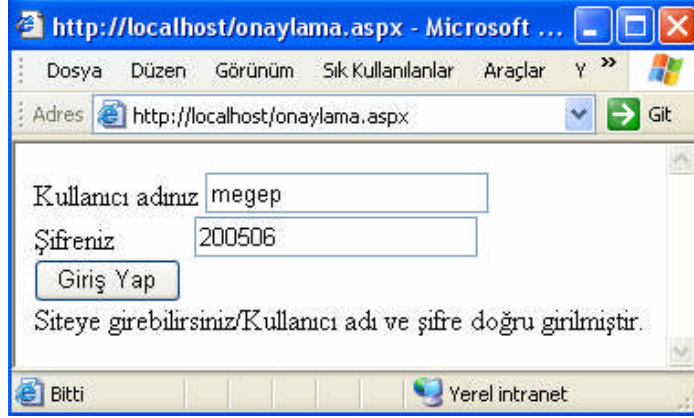
```
1 <script language="VB" runat="server">
2     sub onaylama(sender as object, e as eventargs)
3         if textbox1.text = textbox2.text then
4             label3.text="Şifre onaylandı"
5         end if
6         if textbox1.text <> textbox2.text then
7             label3.text = "Şifre hatalı girilmiştir."
8         end if
9     end sub
10 </script>
11 <html><body>
12 <form runat="server">
13     <asp:label id="label1" text="Şifreniz" runat="server"/>
14     <asp:textbox id="textbox1" textmode="password"
15         runat="server"/>
16     <asp:label id="label2" text="Şifrenizi tekrar girin"
17         runat="server"/>
18     <asp:textbox id="textbox2" textmode="password"
19         runat="server"/><br>
20     <asp:button id="buton" text="Onaylıyorum" onclick="onaylama"
21         runat="server"/><br>
22     <asp:label id="label3" runat="server"/>
23 </form></body></html>
```



Resim 2.3: sifreOnay.aspx ekran çıktısı

Kimi zaman bir şartın gerçekleşmesinden sonra ikinci şartın gerçekleşip gerçekleşmediği araştırılmalıdır. Bu durumda iç içe if deyimi kullanılır.

Kullanıcı adı ve şifre doğru girildiğinde “Siteye girebilirsiniz/Kullanıcı adı ve şifre doğru girilmiştir.” mesajı görüntülenmiştir. Şifre hatalı girildiğinde “Şifre hatalı girilmiştir” mesajı görüntülenmiştir. Kullanıcı adı hatalı girildiğinde ise “Böyle bir kullanıcı yoktur” mesajı görüntülenmiştir.



Resim 2.4: onaylama.aspx ekran çıktıları

2.1.2. If..Then..Else Deyimi

If deyiminin farklı bir kullanımıdır. Else kelimesi Türkçe’de “aksi hâlde, aksi takdirde, değilse” anlamlarına gelir. Bu deyimle, belirtilen şartın gerçekleşmemesi durumunda çalıştırılacak komutlar ifade edilir. Bu deyim kullanım şekli aşağıdaki gibidir:

```
if şart then
    şartın gerçekleşmesi durumunda çalıştırılacak komutlar
else
    şartın gerçekleşmemesi durumunda çalıştırılacak komutlar
end if
```

If then else komutun çalışması şu şekilde ifade edilebilir: Eğer şart gerçekleşirse if ile else arasındaki komutları çalıştır, şart gerçekleşmezse else ile end if arasındaki komutları çalıştır.

If then else komutuyla ilgili gündelik hayattan örnekler vermeye çalışalım.

if hava = bulutlu then şemsiyeni al else şemsiyeni alma end if	if bilgisayar = bozuk then servise götür. else kullanmaya devam et end if	if bilgisayar = açık then kullanmaya başla else bilgisayarı aç end if
--	---	---

Tablo 2.1: If then else komutuyla ilgili gündelik hayattan örnekler

Daha önce oluşturduğumuz parola.aspx dosyasını **if then else** deyimiyle tekrar yazıp, **parolaElse.aspx** ismiyle kaydedelim. Sayfa çalıştırıldığında sonucun değişmediğini görürüz.

parolaElse.aspx

```
1 <script language="VB" runat="server">
2     dim parola as string = "192025"
3     sub onaylama(sender as object, e as eventargs)
4         if textbox1.text = parola then
5             label2.text="Parola onaylandı"
6         else
7             label2.text = "Parola hatası"
8         end if
9     end sub
10 </script>
11 <html><body>
12 <form runat="server">
```

```

13 <asp:label id="label1" text="Parola" runat="server" />
14 <asp:textbox id="textbox1" ontextchanged="onaylama"
    runat="server" />
15 <asp:label id="label2" runat="server" />
16 </form></body></html>

```

Uygulama: sifreOnay.aspx dosyasındaki **if** deyimiyle oluşturulmuş şart kontrolünü **if then else** deyimiyle yapıp sayfayı **sifreOnayElse.aspx** ismiyle kaydediniz.

2.1.3. Else if Deyimi

If deyiminin çok şartlı durumlarda kullanılan bir türüdür. Aşağıda bu deyimin kullanım şekli belirtilmiştir.

```

if şart1 then
    şart1'in gerçekleşmesi durumunda çalıştırılacak komutlar
else if şart2 then
    şart2'in gerçekleşmesi durumunda çalıştırılacak komutlar
else if şart3 then
    şart3'in gerçekleşmesi durumunda çalıştırılacak komutlar
....
....
else
    Önceki şartların gerçekleşmemesi durumunda çalıştırılacak komutlar
end if

```

Else if deyiminin kullanımıyla ilgili gündelik hayattan örnekler verelim.

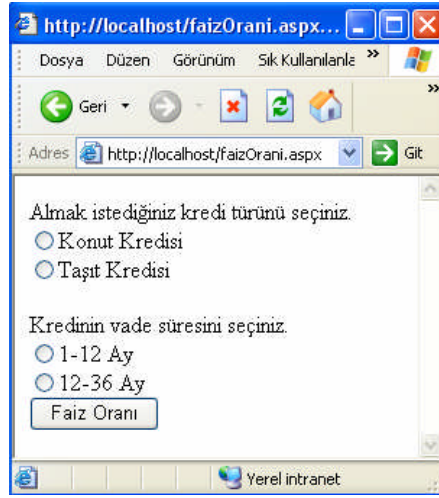
<pre> if sınav sonucu < 45 then not = 1 else if sınav sonucu < 55 then not = 2 else if sınav sonucu < 70 then not = 3 else if sınav sonucu < 85 then not = 4 else not = 5 end if </pre>	<pre> if öğrenci 1.tercihini kazanabiliyor then öğrenciyi 1.tercihine yerleştir else if öğrenci 2.tercihini kazanabiliyor then öğrenciyi 2.tercihine yerleştir else if öğrenci 24.tercihini kazanabiliyor then öğrenciyi 24.tercihine yerleştir else öğrenci herhangi bir bölüme yerleştirilemedi. end if </pre>
---	--

Tablo 2.2: Else if deyiminin kullanımıyla ilgili gündelik hayattan örnekler

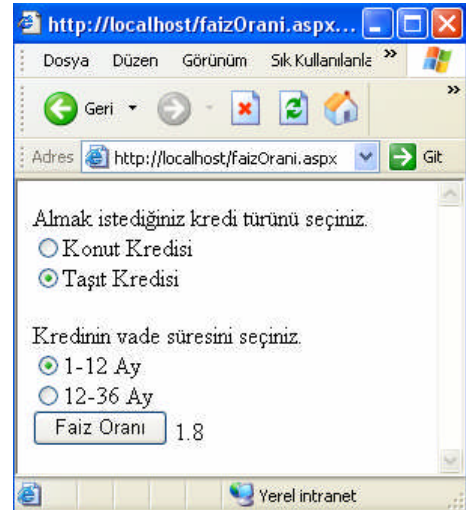
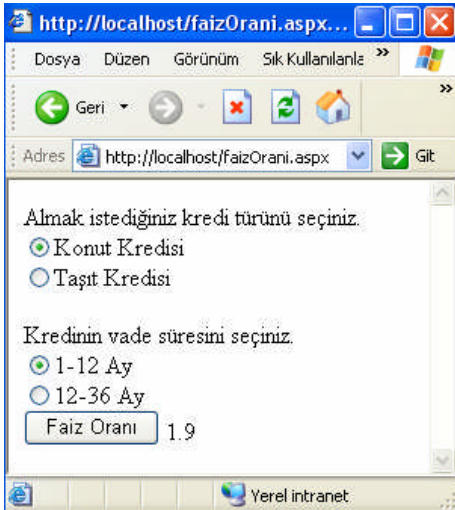
Bu deyimimin kullanımıyla ilgili **faizOrani.aspx** örneğini inceleyelim.

faizOrani.aspx

```
1 <script language="VB" runat="server">
2 dim faizOran as single
3 sub oran(sender as object, e as eventargs)
4 if kredil.checked and vadel.checked then
5 faizOran = 1.90
6 else if kredil.checked and vade2.checked
7 faizOran = 2.00
8 else if kredi2.checked and vadel.checked
9 faizOran = 1.80
10 else if kredi2.checked and vade2.checked
11 faizOran = 1.70
12 end if
13 label3.text = faizOran
14 end sub
15 </script>
16 <html><body>
17 <form runat="server">
18 <asp:label id="labell1" text="Almak istediğiniz kredi türünü
    seçiniz." runat="server"/><br>
19 <asp:radiobutton id="kredil" text="Konut Kredisi"
    groupname="krediler" runat="server"/><br>
20 <asp:radiobutton id="kredi2" text="Taşıt Kredisi"
    groupname="krediler" runat="server"/><p>
21 <asp:label id="label2" text="Kredinin vade süresini seçiniz."
    runat="server"/><br>
22 <asp:radiobutton id="vadel" text="1-12 Ay" groupname="vade"
    runat="server"/><br>
23 <asp:radiobutton id="vade2" text="12-36 Ay" groupname="vade"
    runat="server"/><br>
24 <asp:button id="buton1" text="Faiz Oranı" onclick="oran"
    runat="server"/>
25 <asp:label id="label3" runat="server"/>
26 </form></body></html>
```



Örnekte kullanıcıdan kredi türünü ve vade süresini seçmesi istenmektedir. Seçimine göre faiz oranı görüntülenecektir.



Resim 2.5: faizOrani.aspx ekran çıktıları

2.1.4. Select Case Deyimi

Select Case deyimi, bir değişkenin aldığı değerlere göre farklı işlemler yapmayı sağlar.

Select Case deyiminin kullanım şekli aşağıdaki gibidir :

Select Case (değişken)

Case değer1

Değişkenin içeriği değer1 ise çalıştırılacak komutlar

```

Case değer2
    Değişkenin içeriği değer2 ise çalıştırılacak komutlar
Case değer3
    Değişkenin içeriği değer3 ise çalıştırılacak komutlar
case else
    Değişkenin içeriği belirtilen değerlerin dışında bir değerse
        çalıştırılacak komutlar
end select

```

Select Case deyimiyle ilgili gündelik hayattan örnekler Tablo 2.3'te verilmiştir.

Select Case Atlet Case 1.geldiyse Altın madalya ver Case 2.geldiyse Gümüş madalya ver. Case 3.geldiyse Bronz madalya ver. Case else Bir dahaki sefere inşallah ☺ end select	select Case sınav sonucu case <45 not = 1 case 45 to 54 not = 2 case 55 to 69 not = 3 case else not = 5 end select
--	--

Tablo 2.3: Select Case deyimiyle ilgili gündelik hayattan örnekler

Bir case satırına birden fazla değer konulabilir. Bu durumda uygulama, değişkenin konulan değerlerden birine sahip olması durumuna göre işlem yapar.

Case değer1, değer2, değer3 gibi.

Eğer bir değer aralığı belirtecek isek **Case değer1 to değer2** şeklindeki komut satırını oluştururuz.

Daha önce parola.aspx örneğinde if deyimi kullanarak parola kontrolü yapmıştık. Şimdi de Select Case deyimiyle parola kontrolünü yaptıralım.

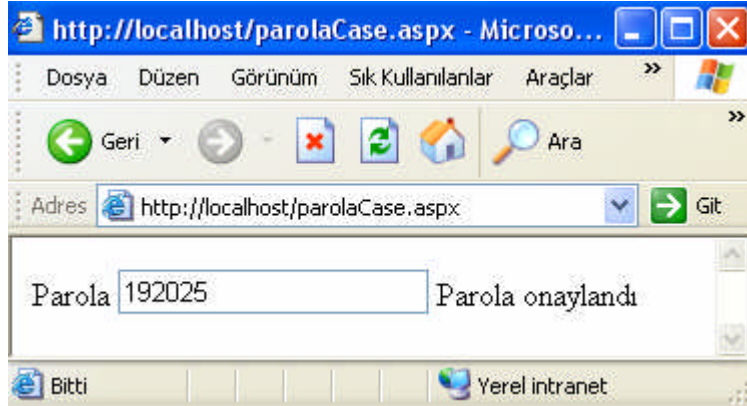
parolaCase.aspx

```

1 <script language="VB" runat="server">
2     dim parola as string = "192025"
3     sub onaylama(sender as object, e as eventargs)
4         select case textbox1.text
5             case parola
6                 label2.text="Parola onaylandı"

```

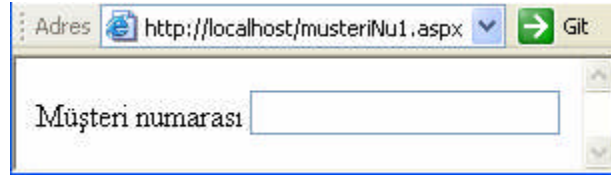
```
7         case else
8             label2.text="Parola hatası"
9         end select
10    end sub
11 </script>
12 <html><body>
13 <form runat="server">
14     <asp:label id="label1" text="Parola" runat="server"/>
15     <asp:textbox id="textbox1" onttextchanged="onaylama"
16         runat="server"/>
17     <asp:label id="label2" runat="server"/>
18 </form></body></html>
```



Resim 2.6: parolaCase.aspx ekran çıktısı

UYGULAMA FAALİYETİ

- Aşağıdaki ekran görüntüsüne benzer şekilde kullanıcıdan müşteri numarasını girmesini isteyen, eğer girilen müşteri numarası doğruysa “Siteye girebilirsiniz”, yanlışsa “Böyle bir müşteri numarası yoktur” mesajı veren ASP.NET sayfasını **If** deyimini kullanarak (Dosyayı musterinu1.aspx adıyla kaydedin) oluşturunuz. (Müşteri numarası **600546** olsun.)



İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Sayfada kullanılacak programlama dilini ifade eden bildirim yazınız.	<ul style="list-style-type: none">➤
<ul style="list-style-type: none">➤ Web formunu oluşturunuz.	<ul style="list-style-type: none">➤ Formun sunucu kontrolü olduğunu belirten runat="server" ifadesini eklemeyi unutmayınız. Etiket kapatmayı unutmayınız.
<ul style="list-style-type: none">➤ Metin kutusu ve etiket kontrollerini tanımlayınız.	<ul style="list-style-type: none">➤ Kontrollerin id ve text parametrelerine kısa, öz, içeriğe uygun ifadeler kullanmaya dikkat ediniz. Metin kutusundaki metnin değişmesi olayı TextChanged ile ifade edilir.
<ul style="list-style-type: none">➤ Alt programı oluşturunuz.	<ul style="list-style-type: none">➤ Program kodları <script> blokları arasında yazılmalıdır. Alt programı sonlandırmayı unutmayınız.
<ul style="list-style-type: none">➤ Müşteri numarasının doğru girilmesi durumunda yapılacak işlemi ifade eden komut satırlarını oluşturunuz.	<ul style="list-style-type: none">➤ Metin kutusundaki değeri isterseniz doğrudan geçerli müşteri numarasıyla karşılaştırabilir isterseniz müşteri numarasını tutan bir değişkenle karşılaştırabilirsiniz. If komutunu sonlandırmayı unutmayınız.
<ul style="list-style-type: none">➤ Müşteri numarasının yanlış girilmesi durumunda yapılacak işlemi ifade eden komut satırlarını oluşturunuz.	

- Yukarıdaki sayfayı **If then else** deyimini kullanarak (Dosyayı musterinu2.aspx adıyla kaydedin) oluşturunuz.
- Aynı sayfayı **Select Case** deyimini kullanarak (Dosyayı musterinu3.aspx adıyla kaydedin) oluşturunuz.

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TEST (ÖLÇME SORULARI)

Aşağıdaki soruları dikkatlice okuyarak uygun cevapları işaretleyiniz.

1. Aşağıdakilerden hangisi bir kontrol deyimi değildir?
A) If...then
B) Else if
C) Select case
D) Dim...as
2. Aşağıdakilerden hangisi doğrudur?
A) Bir uygulamada en fazla bir kontrol deyimi kullanılabilir.
B) ASP.NET uygulamalarında programlama dili kullanılamaz.
C) Şarta bağlı işlemler yapmak için kontrol deyimleri kullanılabilir.
D) Bir mesajı görüntülemek için Select komutu kullanılır.
3. Eğer string tipindeki “ad” değişkeninin içeriği “Ali” ise “Merhaba” mesajını görüntüleyen komut satırı aşağıdakilerden hangisidir?
A) If ad = “Merhaba” then Merhaba
B) If ad = Merhaba then response.write(“Merhaba”)
C) If ad = “Merhaba” then response(“Merhaba”)
D) If ad = “Merhaba” then response.write(“Merhaba”)
4. If komutunun kullanımında; belirtilen şartın gerçekleşmesi durumunda “then” kelimesi ile “end if” kelimeleri arasındaki komutlar çalıştırılır. (D / Y)
5. Else..if deyimi, bir değişkenin aldığı değerlere göre farklı işlemler yapmayı sağlar. (D / Y)

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konulara geri dönerek tekrar inceleyiniz. Tüm sorulara doğru cevap verdiyseniz diğer öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-3

AMAÇ

Döngü işlemlerinin gerçekleştirebileceksiniz.

ARAŞTIRMA

- Gündelik yaşamımızda bir şart gerçekleşene kadar tekrarlanan işlemler vardır. Örneğin, sabahleyin kapat düğmesine basılıncaya kadar telefonun alarmı çalar. Gece belirli saate kadar düzenli aralıklarla otobüs seferleri vardır. Cep telefonda “Cevap Yok” mesajı gelene kadar belirli sayıda telefon çalar. Siz de bir şart gerçekleşene kadar belirli sayıda tekrar eden durumlara örnekler bulunuz. Araştırma sonuçlarınızı öğretmene teslim edecek veya sınıfta sunacak şekilde hazırlayınız.

3. DÖNGÜ DEYİMLERİ

Gündelik yaşamımızda eylemlerimizi/davranışlarımızı belirli şartlar gerçekleşene kadar veya daha önceden belirlediğimiz sayı kadar tekrarladığımız olur. Örneğin, sağlıklı bir yaşam için sabahları egzersiz yapan kişi, her sabah bu egzersizi tekrarlar. Egzersizde belirlediği hareketleri belirli sürede/sayıda yineler. Mesela, her gün 10 dakikalık ısınma hareketlerinden sonra 20’şer defa diğer hareketleri yapar.

Halk oyunları kursuna giden bir kişi, bir hareketi istediği şekilde yapana kadar aralıklarla tekrarlar. Vücut geliştirme çalışması yapan bir kişi belirli kilo ve kas yoğunluğa gelene kadar düzenli egzersizlerine devam eder. Yaşamımızda bir şart gerçekleşene kadar bir işlemi tekrarlamamıza programcılıkta döngü adı verilir.

3.1. Döngü Deyimleri

Programlama dünyasında da bir işlemin belirli sayıda veya bir şart gerçekleşene kadar tekrarlanmasını gerektiren durumlar vardır. Böyle durumlarda döngü deyimlerini kullanırız.

VB.NET’te döngü komutları aşağıda gösterilmiştir.

- For..next
- For each..next
- While..end while

- Do while...loop
- Do until...loop

3.1.1. For..Next Deyimi

İşlemin kaç defa tekrarlanacağını bildiği durumlarda bu deyim kullanılır. For deyimin kullanım biçimi aşağıdaki gibidir:

```
For değişken = başlangıç değeri to bitiş değeri [step step değeri]
    Çalıştırılacak komutlar
Next
```

Bu komutun çalışma şeklini şu şekilde ifade edebiliriz. Değişkenin tuttuğu başlangıç değerinden bitiş değerine kadar belirtilen komutları tekrarlar.

For komutunun kullanımıyla ilgili gündelik hayattan örnekler verelim.

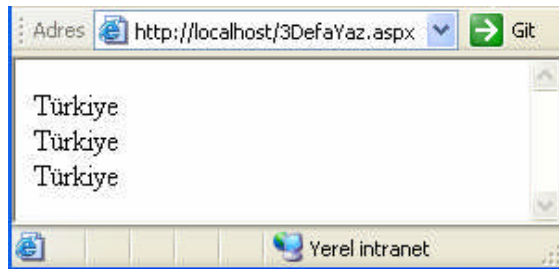
For sınav sayısı = 1 to 20 Sınav çek. Next	For süre = 1 dakika to 10 dakika Isınma hareketi yap. Next
--	--

Tablo 3.1: For komutunun kullanımıyla ilgili gündelik hayattan örnekler

For deyiminin kullanımıyla ilgili birkaç örnek yapalım.

3DefaYaz.aspx

```
1 <script language="VB" runat="server">
2     dim i as integer
3     sub Page_Load(sender as object, e as EventArgs)
4         for i=1 to 3
5             response.write("Türkiye" & "<br>")
6         next
7     end sub
8 </script>
```



Resim 3.1: 3DefaYaz.aspx ekran çıktısı

buyuyerek.aspx

```
1 <script language="VB" runat="server">
2 dim i as integer
3 sub Page_Load(sender as object, e as eventargs)
4     for i = 1 to 7
5         response.write("<font size=" & i & ">" & "Türkiye" &
6             "</font>" & "<br>")
7     next
8 end sub
9 </script>
```



Resim 3.2: buyuyerek.aspx ekran çıktısı



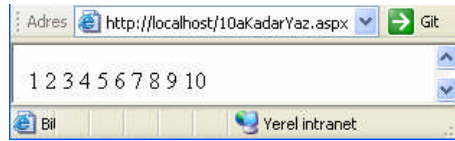
Resim 3.3: kuculerek.aspx ekran çıktısı

kuculerek.aspx

```
1 <script language="VB" runat="server">
2 dim i as integer
3 sub Page_Load(sender as object, e as eventargs)
4     for i = 7 to 1 step -1
5         response.write("<font size=" & i & ">" & "Türkiye" &
6             "</font>" & "<br>")
7     next
8 end sub
9 </script>
```

10aKadarYaz.aspx

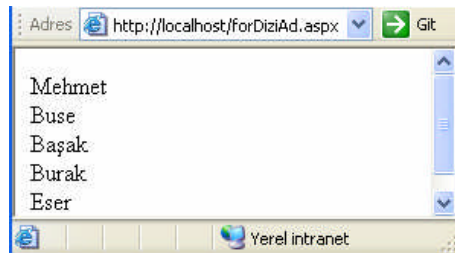
```
1 <script language="VB" runat="server">
2     dim i as integer
3     sub Page_Load(sender as object, e as eventargs)
4         for i=1 to 10
5             response.write(i & " ")
6         next
7     end sub
8 </script>
```



Resim 3.4: 10aKadarYaz.aspx ekran çıktısı

forDiziAd.aspx

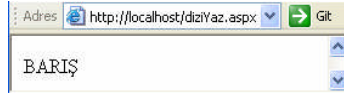
```
1 <script language="VB" runat="server">
2     dim ad(4) as string
3     dim i as integer
4     sub Page_Load(sender as object, e as eventargs)
5         ad(0) = "Mehmet"
6         ad(1) = "Buse"
7         ad(2) = "Başak"
8         ad(3) = "Burak"
9         ad(4) = "Eser"
10        for i=0 to 4
11            response.write(ad(i) & "<br>")
12        next
13    end sub
14 </script>
```



Resim 3.5: forDiziAd.aspx ekran çıktısı

diziYaz.aspx

```
1 <script language="VB" runat="server">
2 dim mesaj(4) as string
3 dim i as integer
4 sub Page_Load(sender as object, e as eventargs)
5     mesaj(0) = "B"
6     mesaj(1) = "A"
7     mesaj(2) = "R"
8     mesaj(3) = "I"
9     mesaj(4) = "Ş"
10    for i=0 to mesaj.Length - 1
11        response.write(mesaj(i))
12    next
13 end sub
14 </script>
```



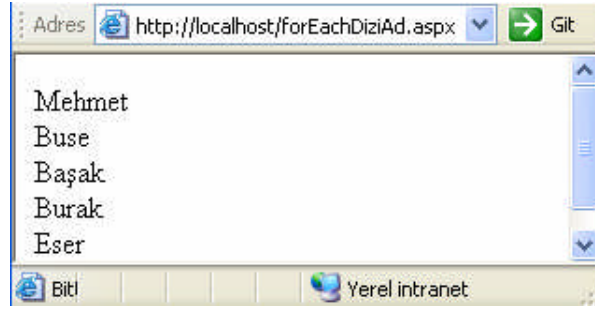
Resim 3.6: diziYaz.aspx ekran çıktısı

3.1.2. For Each Deyimi

Eğer değişen sayıda değer alan diziler veya içindeki kayıt sayısı sürekli değişen veritabanı nesnelerindeki değerlerle ilgili işlemler yapılacaksa For each deyimi kullanılabilir. Bu deyimin çalışma mantığını tam olarak anlayabilmek için aşağıdaki örneği inceleyelim.

forEachDiziAd.aspx

```
1 <script language="VB" runat="server">
2     dim ad(4) as string
3     dim yazdir as string
4     sub Page_Load(sender as object, e as eventargs)
5         ad(0) = "Mehmet"
6         ad(1) = "Buse"
7         ad(2) = "Başak"
8         ad(3) = "Burak"
9         ad(4) = "Eser"
10        for each yazdir in ad
11            response.write(yazdir & "<br>")
12        next
13    end sub
14 </script>
```



Resim 3.7: forEachDiziAd.aspx ekran çıktısı

3.1.3. While..End While Deyimi

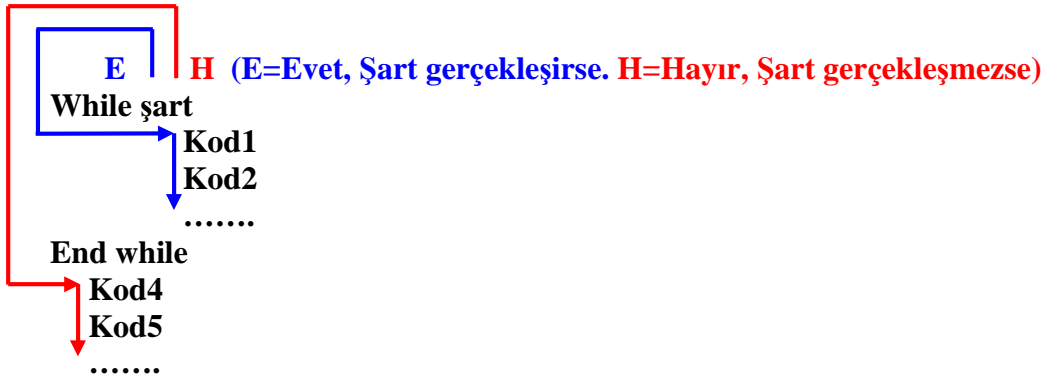
Döngünün kaç defa gerçekleşeceği bilinmediğinde for...next deyimi kullanılamaz. Böylesi durumlarda diğer döngü komutları tercih edilir. Bu komutlardan olan While...End While deyimi bir şart doğru olduğu sürece bir işlemi gerçekleştirmeyi sağlar.

Kullanım şekli:

While şart

Şart doğru olduğu sürece çalıştırılacak komutlar

End while



Şekil 3.1: While...End While deyiminin çalışma mantığı

Bu komutun çalışmasını şu şekilde ifade edebiliriz: Belirtilen şart gerçekleştiği sürece (doğru olduğu sürece) şart ifadesi ile end while arasındaki komutları çalıştırılır, şart gerçekleşmediği anda end while'dan sonraki komutlar çalıştırılır.

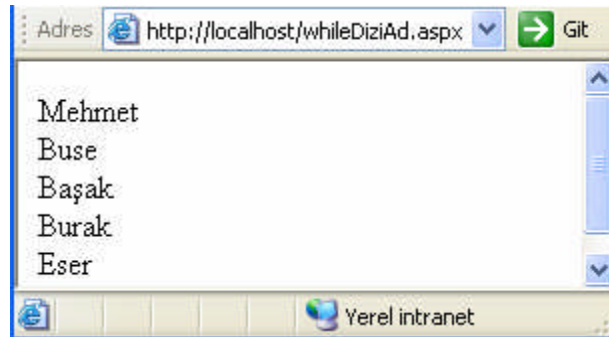
While komutunun kullanımıyla ilgili gündelik hayattan örnekler verelim.

Örneğin, saat 19:00'da tuttuğunuz takımın maçı var diyelim. While (saat daha 19:00 değil) Gündelik işlerinle uğraşmaya devam et End while Maç izlemeye başla.	Sabahleyin 07:30'da evden ayrılıyorsunuz diyelim. While (saat daha 07:30 değil) Yataktan kalk. Elini yüzünü yıka Üzerini giyin Kahvaltını yap End while Yola çık
---	---

While..End While deyiimiyle ilgili örnek yapalım.

whileDiziAd.aspx

```
1 <script language="VB" runat="server">
2     dim ad(4) as string
3     dim i as integer = 0
4     sub Page_Load(sender as object, e as eventargs)
5         ad(0) = "Mehmet"
6         ad(1) = "Buse"
7         ad(2) = "Başak"
8         ad(3) = "Burak"
9         ad(4) = "Eser"
10        while i<=4
11            response.write(ad(i) & "<br>")
12            i = i + 1
13        end while
14    end sub
15 </script>
```



Resim 3.8: whileDiziAd.aspx ekran çıktısı

3.1.4. Do While..Loop Deyimi

Do While..Loop deyiminin 2 tür kullanımı vardır.

1. tür kullanım

```
Do while şart
    Kodlar
Loop
```

2. tür kullanım

```
Do
    Kodlar
Loop while şart
```

Birinci tür kullanımda şart kontrol edilir, şart doğruysa kodlar çalıştırılır, şart yanlışsa kodlar çalıştırılmadan program devam eder. İkinci tür kullanımda kodlar bir defa çalıştırdıktan sonra şart kontrol edilir, şart gerçekleştiği sürece kodların çalıştırılmasına devam edilir. İkinci tür kullanımda şart yanlışsa bile kodlar bir defa çalıştırılır.

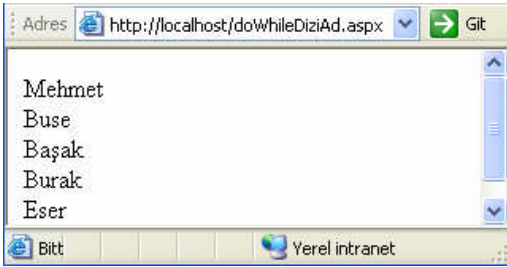
Bu deyimin iki türlü kullanımını içeren aşağıdakileri örnekleri uygulayınız, işleyiş mantığını inceleyiniz.

doWhileDiziAd.aspx

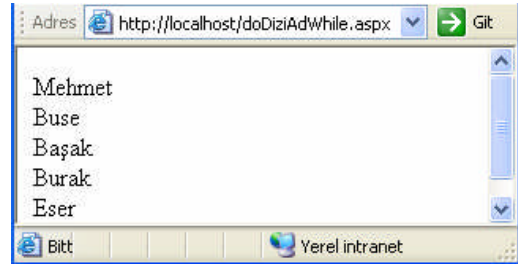
```
1 <script language="VB" runat="server">
2     dim ad(4) as string
3     dim i as integer = 0
4     sub Page_Load(sender as object, e as eventargs)
5         ad(0) = "Mehmet"
6         ad(1) = "Buse"
7         ad(2) = "Başak"
8         ad(3) = "Burak"
9         ad(4) = "Eser"
10        do while i<=4
11            response.write(ad(i) & "<br>")
12            i = i + 1
13        loop
14    end sub
15 </script>
```


doDiziAdWhile.aspx

```
1 <script language="VB" runat="server">
2     dim ad(4) as string
3     dim i as integer = 0
4     sub Page_Load(sender as object, e as eventargs)
5         ad(0) = "Mehmet"
6         ad(1) = "Buse"
7         ad(2) = "Başak"
8         ad(3) = "Burak"
9         ad(4) = "Eser"
10        do
11            response.write(ad(i) & "<br>")
12            i = i + 1
13        loop while i<=4
14    end sub
15 </script>
```



Resim 3.9: doWhileDiziAd.aspx ekran çıktısı



Resim 3.10: doDiziAdWhile.aspx ekran çıktısı

3.1.5. Do Until..Loop Deyimi

Do Until..Loop deyiminin 2 tür kullanımı vardır.

1. tür kullanım

Do until şart

Kodlar

Loop

2. tür kullanım

Do

Kodlar

Loop until şart

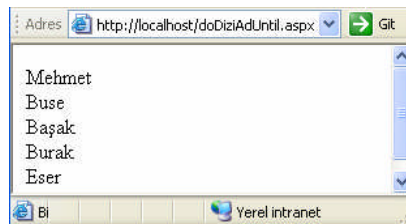
1.kullanımda şart kontrol edilir, şart doğru olana kadar kodlar çalıştırılır, şart doğruysa kodlar çalıştırılmadan program devam eder. 2.kullanımda kodlar bir defa çalıştırdıktan sonra şart kontrol edilir, şart doğru olana kadar kodların çalıştırılması tekrarlanır. Yani, 2.kullanımda şart doğruysa bile kodlar bir defa çalıştırılır.

doUntilDiziAd.aspx

```
1 <script language="VB" runat="server">
2     dim ad(4) as string
3     dim i as integer = 0
4     sub Page_Load(sender as object, e as eventargs)
5         ad(0) = "Mehmet"
6         ad(1) = "Buse"
7         ad(2) = "Başak"
8         ad(3) = "Burak"
9         ad(4) = "Eser"
10        do until i>4
11            response.write(ad(i) & "<br>")
12            i=i+1
13        loop
14    end sub
15 </script>
```

doDiziAdUntil.aspx

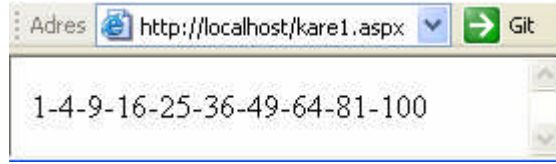
```
1 <script language="VB" runat="server">
2     dim ad(4) as string
3     dim i as integer = 0
4     sub Page_Load(sender as object, e as eventargs)
5         ad(0) = "Mehmet"
6         ad(1) = "Buse"
7         ad(2) = "Başak"
8         ad(3) = "Burak"
9         ad(4) = "Eser"
10        do
11            response.write(ad(i) & "<br>")
12            i=i+1
13        loop until i>4
14    end sub
15 </script>
```



Resim 3.11: doDiziAdUntil.aspx ekran çıktısı

UYGULAMA FAALİYETİ

- Sayfa yüklendiğinde 1’den 10’a kadar sayıların karelerini görüntüleyen ASP.NET sayfasını **for** deyimini kullanarak **kare1.aspx** adıyla oluşturunuz. Sayfanızın tarayıcıdaki görüntüsü aşağıdaki resme benzer olacaktır.



İşlem Basamakları	Öneriler
➤ Sayfada kullanılacak programlama dilini ifade eden bildirimini yazınız.	
➤ Alt programı oluşturunuz.	➤ Program kodları <script> blokları arasına yazılmalıdır. Sayfa yüklendiğinde yapılacak işlemlerin yazıldığı alt programın isminin Page_Load olarak verilmesi gerektiğine dikkat ediniz. Alt programı sonlandırmayı unutmayınız.
➤ Değişkeni tanımlayınız.	➤ Değişkeni isterseniz alt programın dışında da tanımlayabilirsiniz. Tüm alt programların dışında tanımlanan değişkenler bütün alt programlar tarafından kullanılabilir. Belirli bir alt program içinde tanımlanan değişkenler sadece o alt program tarafından kullanılabilir ve diğer alt programlar tarafından tanınmazlar.
➤ For deyimini kullanarak uygun komut satırlarını yazınız.	➤ Bu komutlar sayının karesini alan, yazdıran ve sayılar arasına tire işareti koyan komutlar olacaktır.

- Yukarıdaki sayfayı
 - **While..End While** deyimini kullanarak **kare2.aspx** adıyla oluşturunuz.
 - **Do While..Loop** deyimini kullanarak **kare3.aspx** adıyla oluşturunuz.
 - **Do Until..Loop** deyimini kullanarak **kare4.aspx** adıyla oluşturunuz.

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TEST (ÖLÇME SORULARI)

Aşağıdaki soruları dikkatlice okuyarak uygun cevapları işaretleyiniz.

1. Aşağıdakilerden hangisi bir döngü deyimi değildir?
A) Do while..loop
B) For if
C) Do until..loop
D) While..end while
2. Aşağıdakilerden hangisi doğrudur?
A) Döngü ve kontrol deyimleri aynı amaç için kullanılırlar.
B) Döngü deyimleri kullanılan uygulamada kontrol deyimleri kullanılamaz.
C) Bir işlemin kaç defa tekrarlanacağı belli ise while..end while deyimi kullanılır.
D) Döngünün kaç defa gerçekleşeceği bilinmediğinde for..next deyimi kullanılamaz.
3. Aşağıdaki komutlar çalışınca "Ankara" kelimesi kaç defa ekranda görüntülenir?
sub Page_Load(sender as object, e as EventArgs)
dim i as byte
dim j as byte = 5
for i=1 to j
response.write("Ankara" & "
")
next
end sub
A) 3 B) 4 C) 5 D) 6
4. Eğer değişen sayıda değer alan diziler veya içindeki kayıt sayısı sürekli değişen veritabanı nesnelindeki değerlerle ilgili işlemler yapılacaksa DoWhile..Loop deyimi kullanılabilir. (D / Y)
5. For..next deyimi; döngünün kaç defa gerçekleşeceğinin bilinmediğinde durumlarda kullanılır. (D / Y)

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konulara geri dönerek tekrar inceleyiniz. Tüm sorulara doğru cevap verdiyseniz diğer öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-4

AMAÇ

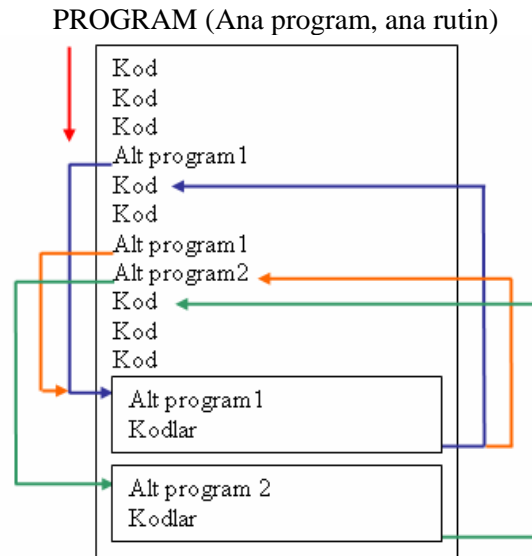
Alt program ve fonksiyonları kullanabileceksiniz.

ARAŞTIRMA

- Programcılıkta hazırlanan programlar alt programlardan oluşur. Her biri bir işlemi yerine getiren alt programlar, programın bütünü oluşturur. Programları alt programlardan oluşan bir bütün hâlinde tasarlamamızın amacı, faydaları ne olabilir? Araştırınız.

4. ALT PROGRAMLAR

Programlar farklı işlemleri yerine getirmekten sorumlu alt programlardan oluşur. Ana programla alt programlar arasındaki çalışma ilişkisini basitçe şu şekilde özetleyebiliriz: Programın kodlarının çalışması esnasında hangi işlem gerçekleştirilecekse o işlemde sorumlu alt program çalıştırılır. Alt programın çalışması bittikten sonra ana program kodları çalıştırılmaya devam edilir. Ayrıca bir alt programdan başka bir alt program da çağrılabilir.



Şekil 4.1: Ana program ve alt programlar arasındaki çalışma ilişkisi

Programların alt programlardan oluşmasının çeşitli yararları vardır. Programlarda bazı işlemler birden fazla gerçekleştirilir. İşlemin gerçekleştirileceği her yerde kodları tekrar tekrar yazmaktansa kodlar alt program içine yazılır ve ne zaman bu kodlara ihtiyaç duyulursa alt program çağrılır ve kodlar çalıştırılır. Böylelikle alt programlar fazladan ve gereksiz kod yazımını azaltarak programların daha küçük boyutlu olmalarını sağlar.

Alt programların kullanımını modüllerimizdeki sözlük bölümünün kullanımına benzetebiliriz. Modülü okumaya başlarız, karşımıza bilmediğimiz İngilizce bir kelime çıkınca sözlük bölümüne geçer, kelimenin Türkçe'sine bakar, tekrar kaldığımız yerden okumaya devam ederiz. Okumaya devam ettikçe bilmediğimiz İngilizce bir kelimeye her rastladığımızda sözlükten kelimenin anlamını öğrenir, ardından modüle kaldığımız yerden devam ederiz. İngilizce kelimenin her geçtiği yerde Türkçe açıklamasını yazmakla uğraşmayız.

Alt programların diğer bir yararı karmaşık, zor bir problemin çözümü için hazırlanan programın daha küçük parçalara ayrılmasını sağlamasıdır. Ayrıca programda bir hata olduğunda programın hepsiyle uğraşmaktansa sadece hatalı olan alt programla uğraşarak zaman ve emekten tasarruf sağlanmış olunur.

Programcılıkta alt program kavramını ifade etmek için alt rutin, fonksiyon, prosedür, yordam kelimeleri de kullanılmaktadır. Alt programlar farklı programlama dillerinde farklı isimlerle kullanılırlar. VB.NET, belirli bir görevi, işlemi gerçekleştirmek için kullanılan ve ana programa bir değer döndürmeyen alt programlar için Subroutine kelimesini (kısaca Sub), ana programa bir değer geri döndüren alt programlar için ise Function kelimesini kullanır. Modüller boyunca Subroutine kavramını ifade etmek için **alt program**, function kavramını ifade etmek için **fonksiyon** kavramı kullanılacaktır.

4.1. Alt Programlar (Subroutines)

Bir eylemi yerine getirmek için kullanılan, ana programa değer göndermeyen alt programlara VB.NET'te subroutine adı verilir. Yazım biçimi aşağıdaki şekildedir :

```
Sub isim [parametreler]
    Kodlar
End Sub
```

Alt program oluşturma işlemi Subroutine kelimesinin kısaltılmış hâli olan **Sub** kelimesiyle başlar. Ardından alt programın ismi yazılır ve istenirse parantez içerisinde parametreler belirtilir. Parametreler sayesinde ana programdan veriler alt programa aktarılır. Parametre kullanımı zorunlu değildir. Son olarak kodlar yazılır ve **end sub** ifadesiyle alt program sonlandırılır.

Dikkat ederseniz şimdiye kadar tüm örneklerimizde alt program kullandık.

```
sub page_load(obj as object, e as eventargs)
sub tiklandi(sender as object, e as eventargs) satırlarını hatırlayın.
```

Alt programların kullanımıyla ilgili olarak toplama ve çıkarma işlemi yapmayı sağlayan basit bir örnek yapalım.

toplaCikar.aspx

```
1 <script language="VB" runat="server">
2     dim sonuc as integer
3     sub sonuclar(sender As Object, e As EventArgs)
4         if islem1.checked then toplama(sayil.text,sayi2.text)
5         if islem2.checked then cikarma(sayil.text,sayi2.text)
6     end sub
7     sub toplama(deger1 as integer, deger2 as integer)
8         sonuc = deger1 + deger2
9         sonucKutusu.text = sonuc
10    end sub
11    sub cikarma(deger1 as integer, deger2 as integer)
12        sonuc = deger1 - deger2
13        sonucKutusu.text = sonuc
14    end sub
15 </script>
16 <html><body>
17 <form runat="server">
18     <asp:label id="baslik" text="<h1>DÖRT İŞLEM</h1>"
19     runat="server" /><p>
20     <asp:label id="label1" text="1.Sayı : " runat="server" />
21     <asp:textbox id="sayil" runat="server" /><br>
22     <asp:label id="label2" text="2.Sayı : " runat="server" />
23     <asp:textbox id="sayi2" runat="server" /><br>
24     <asp:label id="label3" text="Sonuç : " runat="server" />
25     <asp:textbox id="sonucKutusu" runat="server" /><br>
26     <asp:label id="label4" text="İşlemi seçiniz"
27     runat="server"/><br>
28     <asp:radiobutton id="islem1" text="Toplama"
29     groupname="islemler" runat="server"/>
30     <asp:radiobutton id="islem2" text="Çıkarma"
31     groupname="islemler" runat="server"/>
```

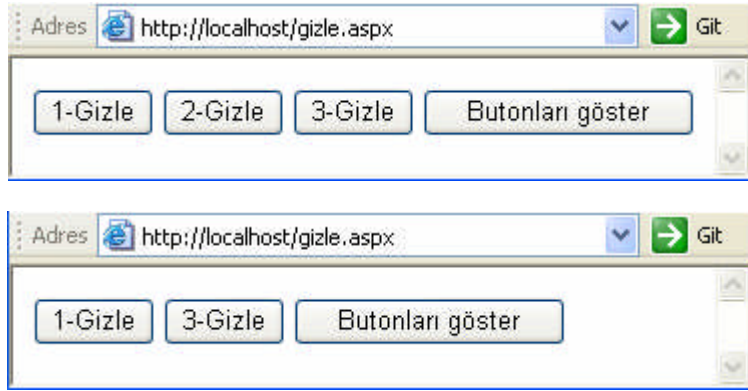
```
28 <asp:button id="buton1" text="Hesapla" onclick="sonuclar"
    runat="server" />
29 </form></body></html>
```

Uygulama: toplaCikar.aspx sayfasını çarpma ve bölme işlemini de yapacak şekilde geliştiriniz. İpucu : Bölme işlemi sonucunda ortaya çıkan bölüm ondalık bir sayı olabilir.

Alt programlar, parametreler vasıtasıyla alt program dışından değerler alır. Aşağıda bu durumla ilgili bir örnek verilmiştir.

gizle.aspx

```
1 <script language="vb" runat="server">
2 sub gizle(sender as object, e as eventargs)
3     sender.visible = false
4 end sub
5 sub goster(sender as object, e as eventargs)
6     buton1.visible = true
7     buton2.visible = true
8     buton3.visible = true
9 end sub
10 </script>
11 <html><body>
12 <form runat="server">
13     <asp:button id="buton1" text="1-Gizle" onclick="gizle"
14         runat="server" />
15     <asp:button id="buton2" text="2-Gizle" onclick="gizle"
16         runat="server" />
17     <asp:button id="buton3" text="3-Gizle" onclick="gizle"
18         runat="server" />
19     <asp:button id="buton4" text="Butonları göster"
20         onclick="goster" runat="server" />
21 </form></body></html>
```

Resim 4.1: gizle.aspx ekran çıktıları

“Gizle” alt programında “sender as object” parametresiyle “sender” ismiyle “object” veri tipinde bir değişken tanımlanmıştır. Bu değişken alt programın çalışmasına neden olan nesneyi tutar. Eğer buton1 nesnesi tıklanarak alt program çalıştıysa sender değişkeninde buton1 nesnesi, buton2 nesnesi tıklanarak alt program çalıştıysa sender değişkeninde buton2 nesnesi tutulmaktadır. Alt program, tıklanılan butonun tarayıcıda görüntülenip görüntülenmemesini sağlayan “visible” (görünür) özelliğini “false” yaparak butonun gizlenmesini sağlamaktadır.

4.2. Fonksiyonlar (Functions)

Alt programlarla benzer işleyişine sahip olan fonksiyonların alt programlardan farkı, ana programa değer döndürmesidir. Alt programlar, tek başlarına bir eylemi yerine getirir, fonksiyonlar ise sadece çağrıldıkları yere değer gönderirler. Fonksiyonların yazım biçimi aşağıdaki gibidir:

```
Function isim (parametreler) as veri tipi
    Kodlar
End Function
```

Fonksiyon tanımlamasında parametrelerden sonraki veri türü, döndürülecek verinin türünü tanımlamaktadır. Alt program kullanarak yaptığımız toplaCikar.aspx uygulamasını şimdi de fonksiyonlarla yapalım.

toplaCikarFunction.aspx

```
1 <script language="VB" runat="server">
2     sub sonuclar(sender As Object, e As EventArgs)
3         if islem1.checked then sonucKutusu.text=
4             toplama(sayil.text,sayi2.text)
5         if islem2.checked then sonucKutusu.text=
6             cikarma(sayil.text,sayi2.text)
```

```

5     end sub
6     function toplama(deger1 as integer, deger2 as integer) as
       integer
7     return deger1 + deger2
8     end function
9     function cikarma(deger1 as integer, deger2 as integer) as
       integer
10    return deger1 - deger2
11    end function
12 </script>
13 <html><body>
14 <form runat="server">
15     <asp:label id="baslik" text="<h1>DÖRT İŞLEM</h1>"
       runat="server" /><p>
16     <asp:label id="label1" text="1.Sayı : " runat="server" />
17     <asp:textbox id="sayi1" runat="server" /><br>
18     <asp:label id="label2" text="2.Sayı : " runat="server" />
19     <asp:textbox id="sayi2" runat="server" /><br>
20     <asp:label id="label3" text="Sonuç : " runat="server" />
21     <asp:textbox id="sonucKutusu" runat="server" /><br>
22     <asp:label id="label4" text="İşlemi seçiniz"
       runat="server"/><br>
23     <asp:radiobutton id="islem1" text="Toplama"
       groupname="islemler" runat="server"/>
24     <asp:radiobutton id="islem2" text="Çıkarma"
       groupname="islemler" runat="server"/>
25     <asp:button id="buton1" text="Hesapla" onclick="sonuclar"
       runat="server"/>
26 </form></body></html>

```

Return komutu fonksiyonun sonuç değerini ana programa göndermek için kullanılır. Fonksiyon içinde kullandığımız return kelimesi yerine aşağıda görüldüğü gibi fonksiyonun ismini de kullanabiliriz.

```

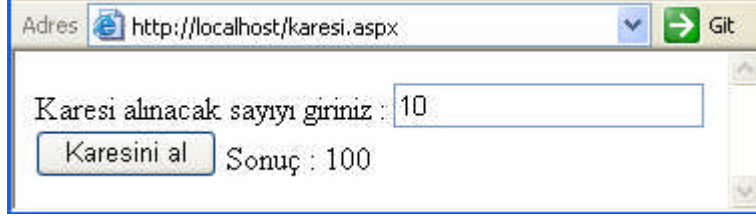
function toplama(deger1 as integer, deger2 as integer) as integer
    toplama = deger1 + deger2
end function

```

Uygulama: toplaCikarFunction.aspx sayfasını çarpma ve bölme işlemini de yapacak şekilde geliştiriniz.

UYGULAMA FAALİYETİ

- Klavyeden girilen sayının karesini alıp tarayıcıda görüntüleyen ASP.NET sayfasını **karesi.aspx** adıyla geliştiriniz. Sayfanızın tarayıcıdaki görüntüsü aşağıdaki resme benzer olacaktır.



İşlem Basamakları	Öneriler
➤ Sayfada kullanılacak programlama dilini ifade eden bildirim yazınız.	➤
➤ Web formunu oluşturunuz.	➤ Formun sunucu kontrolü olduğunu belirten runat="server" ifadesini eklemeyi unutmayınız. Etiket kapatmayı unutmayınız.
➤ Etiket, metin kutusu ve buton kontrollerini tanımlayınız.	➤ Kontrollerin id ve text parametrelerine kısa, öz, içeriğe uygun ifadeler kullanmaya dikkat ediniz. Butona tıkladığında (click) hangi alt programın çalışacağı belirtiniz.
➤ Alt programı oluşturunuz.	➤ Program kodları <script> blokları arasında yazılmalıdır. Alt programı sonlandırmayı unutmayınız.
➤ Metin kutusundaki sayının karesini alarak etiket kontrolüne aktaran komutları yazınız.	➤ Bu işlem tek satırlık bir komutla da gerçekleştirilebilir.

- Aynı sayfayı fonksiyon kullanarak geliştiriniz.

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TEST (ÖLÇME SORULARI)

Aşağıdaki ifadeleri okuyunuz ve doğru olan ifadelerin önündeki boşluğa “D”, yanlış olan ifadelerin önündeki boşluğa “Y” harfi koyunuz.

1. () VB.NET’te ana programa bir değer döndürmeyen alt programlar için sub ifadesi kullanılır.
2. () Function ifadesi VB.NET’te fonksiyon kullanmak için kullanılır.
3. () ASP.NET uygulamalarında fonksiyon kullanılamaz.
4. () Alt program kullanılan uygulamada fonksiyon kullanılamaz.
5. () Farklı programlama dillerinde alt program kavramı için alt rutin, fonksiyon, prosedür, yordam kelimeleri de kullanılır.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konulara geri dönerek tekrar inceleyiniz. Tüm sorulara doğru cevap verdiyseniz diğer öğrenme faaliyetine geçiniz.

MODÜL DEĞERLENDİRME

OBJEKTİF TEST (ÖLÇME SORULARI)

Aşağıdaki soruları dikkatlice okuyarak uygun cevapları işaretleyiniz.

1. Dim as gezegen(9) string şeklinde tanımlanan diziyle ilgili olarak aşağıdakilerden hangisi yanlıştır?
A) Dizi 9 elemandan oluşmaktadır.
B) Dizinin ilk elemanı gezegen(0)'dır.
C) Dizinin son elemanı gezegen(9)'dur.
D) Dizinin tüm elemanları "string" tipindedir.
2. Aşağıdakilerden hangisi, 5 elemanlı, elemanlarının tipi string olan ogrenim_durumu adlı dizinin tanımlanmasıdır?
A) dim ogrenim_durumu(5) as integer
B) dim ogrenim_durumu as string
C) dim ogrenim_durumu(5) as string
D) dim ogrenim_durumu(4) as string
3. İşlem önceliği kavramına göre aşağıdaki işlemlerden hangisi doğrudur?
A) $7 + 14 / 7 = 3$
B) $2 + 7 * 3 = 27$
C) $12 + 3 / 3 = 13$
D) $15 - 5 / 1 = 11$
4. Aşağıdaki komut satırlarının çalışması sonucunda ekran çıktısı ne olur? (Integer olarak tanımlanan x değişkeninin değeri 5'dir.)

```
If x = 0 then  
    Response.write("x sifıra eşittir.")  
Else if x > 0 then  
    Response.write("x pozitif bir sayıdır. ")  
Else  
    Response.write("x negatif bir sayıdır.")  
End if
```


A) x sifıra eşittir.
B) x pozitif bir sayıdır.
C) x negatif bir sayıdır.
D) Ekranda bir mesaj gözükmez.
5. Bir ASP.NET uygulamada bir (1) sayısını hafızada tutmak için aşağıdaki veri tiplerinden hangisi kullanılamaz?
A) String
B) Char
C) Integer
D) Date

PERFORMANS TESTİ (YETERLİK ÖLÇME)

Modül ile kazandığınız yeterlik, aşağıdaki ölçütlere göre değerlendirilecektir.

Değerlendirme Ölçütleri	Evet	Hayır
1. Değişkenin tanımını yapabilir misiniz?		
2. Değişkenin ne olduğunu açıklayabilir misiniz?		
3. Elindeki veri için uygun veri tipini seçebilir misiniz?		
4. Hatasız değişken isimlendirmesi yapabilir misiniz?		
5. Hatasız bir biçimde değişken tanımlayabilir misiniz?		
6. Değişkene değer atayabilir misiniz?		
7. Değişkenler arasında tip dönüşümünü gerçekleştirebilir misiniz?		
8. Ne tür veriler için dizileri kullanmak uygundur? Açıklayabilir misiniz?		
9. İşlemlerde işlem önceliğini uygulayabilir misiniz?		
10. Bir problemin çözümünde hangi kontrol deyiminin kullanılması gerektiğine karar verip uygulayabilir misiniz?		
11. Bir problemin çözümünde hangi döngü deyiminin kullanılması gerektiğine karar verip kullanabilir misiniz?		
12. Bir problemin çözümünde hangi kontrol deyiminin kullanılması gerektiğine karar verip kullanabilir misiniz?		
13. Gerektiğinde alt programları kullanabilir misiniz?		
14. Gerektiğinde fonksiyonları kullanabilir misiniz?		

DEĞERLENDİRME

Yaptığınız değerlendirme sonucunda eksikleriniz varsa öğrenme faaliyetlerini tekrarlayınız.

Modülü tamamladınız, tebrik ederiz. Öğretmeniniz size çeşitli ölçme araçları uygulayacaktır, öğretmeninizle iletişime geçiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	C
2	C
3	B
4	D
5	A
6	A

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	D
2	C
3	D
4	D
5	Y

ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

1	B
2	D
3	C
4	Y
5	Y

ÖĞRENME FAALİYETİ-4'ÜN CEVAP ANAHTARI

1	D
2	D
3	Y
4	Y
5	D

MODÜL DEĞERLENDİRME CEVAP ANAHTARI

1	A
2	D
3	C
4	B
5	D

KAYNAKÇA

- DEMİRKOL, Zafer, **ASP.NET**, Pusula Yayıncılık, İstanbul, 2005.
- DUTHIE, G. Andrew, **Adım Adım Microsoft ASP.NET**, Arkadaş Yayınevi, Ankara, 2005.
- SANKUR, Bülent, **Bilişim Sözlüğü 2005 Programı**, Yazılım: Hakan GÜLERYÜZ, Pusula Yayıncılık.
- PALA, Zeydin, **ASP.NET ile Adım Adım Web Uygulamaları**, Türkmen Kitabevi, İstanbul, 2006.
- Data Type Summary
(VB).msdn2.microsoft.com/en-us/library/47zceaw7(VS.80).aspx
- www.15seconds.com/howto/pg000056.htm
- www.asp101.com/articles/sample_chapters/sitepoint_byoaspnet/default.asp
- www.easerve.com/developer/tutorials/asp-net-tutorials-dates.aspx
- www.evcil.net/Makale/58.aspx
- www.harding.edu/fmccown/vbnet_csharp_comparison.html
- www.informit.com/articles/article.asp?p=30110&r1=1
- msdn.microsoft.com
- www.startvbdotnet.com/language/conversion.aspx
- visualbasic.about.com/od/usingvbnet/a/datatypes.htm

SÖZLÜK

Terim	Okunuşu	Anlamı, Açıklaması
Array	Erey	Dizi
Boolean	Bulyın	Bir veri türü
Button	Batın	Düğme, buton
Byte	Bayt	Bayt
Case	Keys	1.Durum 2.Büyük küçük şıkkı. Metin işlemede bir harfin büyük mü küçük mü yazılacağıının belirtimi
case-sensitive	keys sensitiv	Büyük/Küçük Harfe Duyarlı
Char	Çar	Bir veri tipi
Character	Keriktır	Karakter
Date	Deyt	Tarih
Do Until...Loop	Du Antıl...Lup	Döngü deyimi
Do While...Loop	Du Vayl lup	Döngü deyimi
Double	Dabil	Bir veri türü
Else	Els	Değilse
End	End	Son, bitiş.
False	Fols	Yanlış
For Each...Next	For içç...Nekst	Döngü deyimi
Function	Fankşın	Fonksiyon
Group	Grup	Grup
If	İf	Eğer
Integer	İnticır	Tam sayı
Long	Long	Bir veri türü
Loop	Lup	Döngü
Select	Silekt	Seçmek
Short	Şort	Bir veri türü
Single	Singıl	Bir veri türü
String	sitrin	Metin, Dizgi, Dizge
Subroutine	Sabrutin	Alt rutin, alt program, yordam
Text	Tekst	Metin
Then	Den	İse
True	Turu	Doğru
Variable	Veriyıbl	Değişken

