

**T.C.  
MİLLÎ EĞİTİM BAKANLIĞI**

# **YENİLENEBİLİR ENERJİ TEKNOLOJİSİ**

## **GÖRSEL PROGRAMLAMA MANTIĞI**

**Ankara, 2014**

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

# İÇİNDEKİLER

AÇIKLAMALAR .....	iii
GİRİŞ .....	1
ÖĞRENME FAALİYETİ-1 .....	3
1. TEMEL KOD YAZIM KURALLARI .....	3
1.1. Temel Yazım Kuralları .....	3
1.2. Sabitler .....	4
1.3. Değişkenler .....	5
1.3.1. Değişken Tipleri .....	6
1.3.2. Değişkenlerin Tanımlanması .....	7
1.3.3. Tip bildirimi .....	8
UYGULAMA FAALİYETİ .....	9
ÖLÇME VE DEĞERLENDİRME .....	11
ÖĞRENME FAALİYETİ-2 .....	12
2. AKIŞ DENETİMİ .....	12
2.1. Şartlı Dallanma .....	12
2.1.1. If- Then .....	13
2.1.2. If-Then-Else .....	13
2.1.3. If – Then – Else If .....	14
2.1.4. Select Case / Case Else .....	17
UYGULAMA FAALİYETİ .....	19
ÖLÇME VE DEĞERLENDİRME .....	21
ÖĞRENME FAALİYETİ-3 .....	22
3. DÖNGÜLER .....	22
4.1. For - Next .....	22
4.2. Do - Loop .....	27
UYGULAMA FAALİYETİ .....	31
ÖLÇME VE DEĞERLENDİRME .....	33
ÖĞRENME FAALİYETİ-4 .....	34
4. DİZİLER .....	34
4.1. Dizi Nedir? .....	34
4.2. Tek Boyutlu Diziler .....	34
4.3. Tek Boyutlu Diziler .....	36
4.4. Statik Diziler .....	37
4.5. Dinamik Diziler .....	38
UYGULAMA FAALİYETİ .....	40
ÖLÇME VE DEĞERLENDİRME .....	42
ÖĞRENME FAALİYETİ-5 .....	44
5. PROSEDÜR (ALT YORDAM) OLUŞTURMA .....	44
5.1. Prosedürler (Alt Yordamlar) .....	44
5.2. Fonksiyonlar .....	46
5.3. Hazır Fonksiyon Örnekleri .....	47
5.3.1. Alfa Nümerik Fonksiyonlar (Karakter Dizisi (String)) .....	47
5.3.2. Nümerik Fonksiyonlar (.Sayısal Fonksiyonlar) .....	53
5.3.3. Kullanıcı Tanımlı Fonksiyonlar .....	55

---

UYGULAMA FAALİYETİ .....	56
ÖLÇME VE DEĞERLENDİRME .....	58
MODÜL DEĞERLENDİRME .....	60
CEVAP ANAHTARLARI.....	62
KAYNAKÇA .....	64

# AÇIKLAMALAR

<b>ALAN</b>	<b>Yenilenebilir Enerji Teknolojisi</b>
<b>DAL/MESLEK</b>	<b>Bilgisayarlı Kontrol</b>
<b>MODÜLÜN ADI</b>	<b>Görsel Programlama Mantığı</b>
<b>MODÜLÜN TANIMI</b>	“Visual Basic” programlama dilinde kullanılan değişken tanımlama, dizi, fonksiyon ve akış kontrol yapılarını öğrenme becerisinin ve tasarlanan algoritmaları program hâline getirebilme yeterliliğinin kazandırıldığı öğretim materyalidir.
<b>SÜRE</b>	40/32
<b>ÖN KOŞUL</b>	Görsel Programlamanın Temelleri modülünü almış olmak.
<b>YETERLİK</b>	Görsel programlamada kod yazmak
<b>MODÜLÜN AMACI</b>	<b>Genel amaç</b> Görsel programlamada kod yazımını program ifadelerine göre yapabileceksiniz ve temel programlama tekniklerini kullanarak programlar yazabileceksiniz. <b>Amaç:</b> <ol style="list-style-type: none"><li>1. Görsel programlama değişken atamasını hatasız olarak yapabileceksiniz.</li><li>2. Günlük hayattan basit örnekler araştırınız. Bu basit örneklere ait akış diyagramları oluşturunuz.</li><li>3. Bir işlemin tamamlanması veya sonucun bulunması için tekrarlanması gerekir. Bu öğrenme faaliyetinde Doo – Loop ve For – Next döngü işlemleri öğreneceksiniz.</li><li>4. Görsel programlamada dizi yapısını doğru olarak kullanabileceksiniz.</li><li>5. Görsel programlama alt yordam oluşturma işlemini doğru olarak yapabileceksiniz.</li></ol>
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	<b>Ortam:</b> Bilgisayar laboratuvarı <b>Donanım:</b> Bilgisayar
<b>ÖLÇME VE DEĞERLENDİRME</b>	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test,doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.



# GİRİŞ

## Sevgili Öğrenci,

Görsel programlama tekniđi, program geliştirme işine farklı bir anlayış kazandırırken klasik programlama dillerinin yöntemlerinden de faydalanır. Bu anlamda basic, pascal ve C gibi satır programlama dillerde kullanılan program kontrol tekniklerinin bilinmesi gereklidir.

Visual Basic, temel olarak basic programlama dilini esas alır. Basic dilinde kullanılan komutlar, yazım kuralları, deđişkenler gibi kavramlar aynen Visual Basic dilinde kullanılabilir.

Görsel Programlama Mantıđı modülü, tasarlanan form ve kullanılan nesnelere neyin nasıl yapılması gerektiđini gösteren kod satırlarını yazabilme temel becerilerini kazandırmak için hazırlanmış bir modüldür. Bu modülde sizler, deđişken ataması yapma, alt yordamlar ve fonksiyonlar oluşturma, program akışının kontrolünü gerçekleştirme gibi işlemleri yapabileceksiniz. Bu bilgi ve beceriler, bilgisayarlı otomasyon sistemlerinde Visual Basic programlama dili kullanarak kontrol yazılımları geliştirebilmek için gerekli temel noktalardır.

Bu modülde öğrencinin metin kutusu, etiket ve buton nesnelere kullanmayı bildiđi varsayılmıştır. İleriki modüllerde öğretilecek olan diđer nesnelere kullanılışından kaçınılmıştır. Bu modül ile hedeflenen, “Visual Basic”te kullanılan program yazma ve denetimini yapabileme yeterliđini kazanmanızdır.





# ÖĞRENME FAALİYETİ-1

## AMAÇ

Görsel programlama değişken atamasını hatasız olarak yapabileceksiniz.

## ARAŞTIRMA

- Günlük hayatta kullandığımız veri örnekleri üzerinde düşünerek bunların ne tür değerler alabilecekleri hakkında araştırma yapınız.

## 1. TEMEL KOD YAZIM KURALLARI

### 1.1. Temel Yazım Kuralları

Kod yazım kurallarına uyulmadan programların çalıştırılması mümkün değildir. Visual Basic, kod cümleleri yazılırken büyük küçük harf ayrımı yapmaz. Visual Basic editörü otomatik olarak kod cümlelerini formata uygun hâle getirir. Eğer yazım hatası yapılmış ise hatalı kod cümlesi kırmızı renk ile gösterilerek uyarı verir.

Kod cümlelerinin sonuna diğer dillerde olduğu gibi (örneğin; Pascal dilinde “ ; ” gibi) herhangi bir noktalama işareti konulmaz.

Kod cümlelerinin bulunduğu satırlara programın anlaşılmasını kolaylaştıran açıklayıcı bilgiler ilave edilecekse üst tırnak “ ’ ” işareti konulur. Editör açıklama cümlelerini aşağıda görüldüğü gibi otomatik olarak yeşil renk ile gösterir.

#### Örnek Kod:

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        'Bu buton sayfayı kapatacak - Editör açıklaması
        Me.Close()
    End Sub

End Class
```

Değişken isimler yazılırken semboller ve işaretler kullanılmaz. Sadece alt çizgi karakteri “ \_ ” kullanılabilir. Değişken isimler en fazla 255 karakter uzunluğunda olabilir. Bir alt programda birden fazla değişkene aynı isim verilemez. Değişken isminin ilk karakterinin her zaman harf ile başlaması zorunludur.

Visual Basic nesne tabanlı (object-oriented) ve olay güdümlü (event-driven) bir programlama dilidir. Örneğin; formun (kendisi de dâhil olmak üzere) üzerine yerleştirdiğimiz etiket, buton veya metin kutuları birer nesnedir. Bu nesnelerin altına gerekli kodlar yazılır. Her bir nesnenin altına yazılan kod parçacıklarına o nesneye ait alt yordam adı verilir. Alt yordamın tanımlama kısmında belirtilen olayın gerçekleşme durumuna göre alt yordam çalışır. Visual Basic dilinde çeşitli olaylar mevcuttur. Bunlardan bazılarını aşağıdaki tabloda görebiliriz.

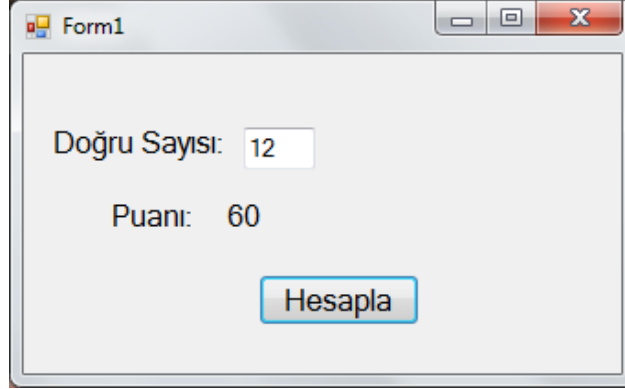
Olay	Açıklaması
Click	Form üzerinde fareyi bir defa tıklamak
DbClick	Form üzerinde fareyi çift tıklamak
DragDrop	Fareyle form üzerinde nesneyi tutup sürükleyerek hedef üzerine bırakmak
DragOver	Fareyle form üzerinde nesneyi tutup sürükleyerek hedef üzerine gelmek
KeyDown	Klavyedeki tuşlara basılma anı
KeyPress	Klavyedeki tuşlara basmak
KeyUp	Klavyedeki basılmış olan tuşu bırakma anı
GotFocus	Form üzerindeki bir nesneye odaklanma (seçme)
LostFocus	Form üzerindeki bir nesneden odağı (seçilmeyi) kaybetme
MouseOver	Fare işaretçisi (ok) ile nesne üzerine gelme
MouseUp	Form üzerinde kullanıcının bastığı fare tuşunu bırakması
Mousedown	Form üzerinde kullanıcının farenin herhangi bir tuşuna basması

## 1.2. Sabitler

Programlarımız içinde değerini bildiğimiz ve değişmeyen verileri bellekte bir isim altında saklamak ve yeri geldiğinde bu isimle kullanmak için tanımlanır. Değişkenler için geçerli isimlendirme kuralları sabit isimler için de geçerlidir. İngilizce **Constant** kelimesinin Türkçe karşılığı **sabit**'tir. Visual Basic'te (VB) sabit **Const**'tir.

**Kullanımı:** Const SabitAdı

**Örnek:** Her doğru sorunun değeri 5 puan olsun. Doğru yanıt sayısını 5 ile çarpıp puanı hesaplayan program aşağıdaki gibidir. Burada doğru sayı değeri (5 puan) bir sabittir.



```
Public Class Form1
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click
```

```
        Const katsayi = 5  
        Label2.Text = katsayi * TextBox1.Text
```

```
    End Sub
```

```
End Class
```

### 1.3. Değişkenler

Değişken program kodları içinde kullanılacak ya da üretilen değerlerin aktarıldığı bilgisayar hafızası içinde ayrılmış sabit miktardaki alanların isimlerine denir. Bu alanların isimleri ve boyutları sabit olmakla beraber sakladıkları değerler program çalışması süresince değişebilir.

**Örneğin;** İki sayı toplanıp ismi SONUC olan bir değişkende saklanabilir. SONUC değişkenindeki değer gerektiğinde farklı amaçlar için kullanılabilir. Toplanacak sayılar değiştiğinde, toplam yine SONUC değişkeninde saklanır.

**Değişkenlerden önce şu bilgiyi hatırlamakta yarar vardır.**

Bayt (İngilizce: byte), elektronik ve bilgisayar bilimlerinde genellikle 8 bitlik dizilim boyunca 1 veya 0 değerlerini bünyesine alan ve kaydedilen bilgilerin türünden bağımsız bir bellek ölçüm birimidir. Bir bayt, Latin alfabesini baz alan 8 bitlik bir kodlamada herhangi bir harfi temsil eder.

8 bit = 1 bayt'tır.

Bitten sonraki ikinci en küçük sayısal bilgisayar birimidir. Bir Bayt, 0 ile 255 arasındaki değeri veya diğer anlamda 256 şalter durumunu temsil etmektedir. Yani  $2^8$ 'dir.

1 byte = 8 bit  $\rightarrow 2^8 = 256 \rightarrow 0$  ile 255 arası deęerler.  
2 byte = 16 bit  $\rightarrow 2^{16} = 65536 \rightarrow 0$  ile 65535 arası deęerler.  
4 byte = 32 bit  $\rightarrow 2^{32} = 4294967296 \rightarrow 0$  ile 4294967295 arası deęerler.

### 1.3.1. Deęişken Tipleri

Deęişken tiplerini sayısal bilgi, karakter bilgi, mantıksal bilgi ve tarih bilgi olmak üzere dört grupta incelenir. Bu dört grubu da kapsayan veri tipi olarak Variant kullanılır.

Tip	Açıklama	Uzunluk
Variant	Her türlü veri depolayabilen veri tipidir.	16 byte

#### 1.3.1.1. Sayısal Deęişkenler

Bilgisayarlarda işlenen verilerin çoęu sayısal bilgilerdir. Bellek kaynaklarını en verimli bir şekilde kullanmak için sayısal bilgilerin durumuna göre bellekte yer ayırmak gerekir. Bunun için de uygun veri tipinin seçilmesi gerekir.

Aşağıdaki tabloda Visual Basic dilinde kullanılan sayısal deęişken tipleri gösterilmiştir.

Tip	Açıklama	Uzunluk
Byte	0 ile 255 arası tam sayı	1 byte
Integer	Kısa tam sayı	2 byte
Long	Uzun tam sayı	4 byte
Single	Kısa rasyonel sayı	4 byte
Double	Uzun rasyonel sayı	8 byte

#### 1.3.1.2. String (Karakter Dizisi) Deęişkenler

Bu tip deęişkenler sadece metin türünde bilgiler saklar. Metin uzunluğu 0 ile 65535 karakter arasında olabilir. Bellekte kapladığı yer metin uzunluğu kadardır.

Tip	Açıklama	Uzunluk
String	Karakter dizisi	n byte

### 1.3.1.3. Mantıksal Değişkenler

Mantıksal değişken, 0 ve 1 olmak üzere yalnızca iki tane değer alır. Bellekte sadece 2 byte yer kaplar. Genellikle karar ve şart ifadelerinde kullanılır.

Tip	Açıklama	Uzunluk
Boolean	Doğru (true) ya da yanlış (false) şeklinde mantıksal tipdir.	2 byte

### 1.3.1.4. Tarih Değişkenleri

Kod içinde tarih ve zamana ilişkin verileri saklayan değişkenlerin türüdür. Bellekte 8 byte yer ayrılır.

Tip	Açıklama	Uzunluk
Date	Tarih ve saat türü bilgiler	8 byte

### 1.3.2. Değişkenlerin Tanımlanması

Program yazımında kullandığımız değişkenlerin tanımlanması değişkenin yerel (lokal) ve genel (global) olması durumuna göre yapılır. Yerel değişken yalnızca alt yordam (prosedür) ve fonksiyonlar içinde kullanılan değişkenlerdir. Bu değişkenler alt yordam ya da fonksiyon çalışmaya başladığında kullanılır. Genel değişken ise programın tamamı boyunca kullanılır ve değer alabilir.

Değişkenin tanımlanırken saklayabileceği veri türünün belirtilmesi hafızada ayrılacak miktarının belirli olmasını sağlar. Eğer değişkenlerin tipini belirtmeden bir kullanım yaparsak bu değişkenlerin “variant” tipinde olduğu kabul edilir. Bu da hafızada gereksiz yer kaybına sebep olur.

Değişken tanımlama formatı şu şekildedir:

**Tanımlama Deyimi** değişken adı **as** değişken tipi

Örnek; **Dim** sayı **As Integer**

### 1.3.3. Tip bildirimi

Değişkenlerde tanımlama deyimleri

<b>Dim</b>	<b>Bir form ya da alt yordam (prosedür) içinde kullanılacak değişkenlerin tanımlanmasında kullanılır.</b>	<b>Dim sayi As Integer</b>
<b>Private</b>	Değişkenin sadece tanımlandığı form ya da alt yordam içinde geçerli olmasını (lokal değişken) sağlar.	<b>Private isim As String</b>
<b>Public</b>	Değişkenlerin tüm form, modül ve alt yordamlarda geçerli olması isteniyorsa bu deyim kullanılır.	<b>Public isim As String</b>
<b>ReDim</b>	Dinamik dizilerde tekrar dizi boyu tanımlaması için kullanılır.	
<b>Static</b>	Form ya da alt yordamların tekrar çağrılmalarına kadar değişkenin önceki değerini saklaması istendiğinde kullanılır.	<b>Static yıl As Date</b>

Form ve alt yordamlarda tanımlama deyimleri

<b>Private</b>	<b>Değişkenin sadece tanımlandığı form ya da alt yordam içinde geçerli olmasını (lokal değişken) sağlar.</b>	<b>Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click</b>
<b>Public</b>	Değişkenlerin tüm form, modül ve alt yordamlarda geçerli olması isteniyorsa bu deyim kullanılır.	<b>Public Class Form1</b>

## UYGULAMA FAALİYETİ

Ekran görüntüsü verilen formu hazırlayınız. Metin kutularına yazılan sayılarla dört işlem yapan programı yazınız.

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two text input fields: "Birinci Sayı" with the value "34" and "İkinci Sayı" with the value "23". To the right of these fields is a group box labeled "İşlem Seçiniz" containing four radio button options: "Topla", "Çıkar", "Çarp" (which is selected), and "Böl". In the center of the form is a button labeled "HESAPLA ==>". To the right of this button is another text input field labeled "Sonuç" which contains the value "782".

İşlem Basamakları	Öneriler
➤ Microsoft Visual Basic programında bir proje oluşturunuz.	
➤ Formunuzu yukarıdaki ekran görüntüsündeki gibi tasarlayınız.	
➤ HESAPLA butonu Click Event'ini çift tıklayın ve gerekli kodu yazınız.	➤ Kod yazım kurallarına dikkat ediniz.
➤ Yazdığınız programı deneyiniz.	➤ Hata varsa düzeltiniz.
➤ Metin kutularına hiç sayı yazmadan HESAPLA butonuna tıklayınız. Formun çalışmadığını ve hata verdiğini göreceksiniz. Hata mesajı: <i>"Conversion from string "" to type 'Integer' is not valid."</i>	➤ Böyle durumlar için nasıl bir önlem almak gerektiğini araştırınız.

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Formu oluşturabildiniz mi?		
2. Buton Click Event'i için gerekli kodu yazabildiniz mi?		
3. Programınızı deneyebildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “Evet” ise “Ölçme ve Değerlendirme” ye geçiniz.



## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerde verilen bilgiler doğru ise (D), yanlış ise (Y) yazınız.

1. Aşağıdaki değişken isimlerinden hangisi yanlıştır ?  
A) Hasta\_kayıt B) xWq12 C) 12\_A\_sınıfı D) Per\_kimlik
2. "Single" tipi değişkenin uzunluğu kaçtır?  
A) 1 byte B) 2 byte C) 3 byte D) 4 byte
3. Değişken isimleri en fazla kaç karakter uzunluğunda olabilir?  
A) 64 B) 256 C) 255 D) 127
4. .... tipi her türlü veriyi depolayabilir.  
A) Double B) Variant C) Type D) Const
5. Değişkenlerin tüm form, modül ve alt yordamlarda geçerli olması isteniyorsa ..... deyimini kullanılır.  
A) Double B) Variant C) Local D) Public
6. Genellikle karar ve şart ifadelerinde değişken tipi olarak aşağıdakilerden hangisi kullanılır?  
A) Date B) String C) Boolean D) Currency
7. İçeriğini bildiğimiz ve değeri değişmeyen verileri program içinde nasıl tanımlayabiliriz?  
A) Olay B) Fonksiyon C) Değişken D) Sabit
8. Değişken isminin ilk karakteri her zaman ..... başlar.  
A) Harf ile B) Sayı ile C) # işareti ile D) & işareti ile
9. Programlarımızda açıklayıcı satırlar eklemek için satır başında aşağıdaki karakterlerden hangisi kullanılır?  
A) : B) \ C) ' D) \*

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-2

## AMAÇ

Görsel programlama program akış ifadelerini hatasız olarak kullanabileceksiniz.

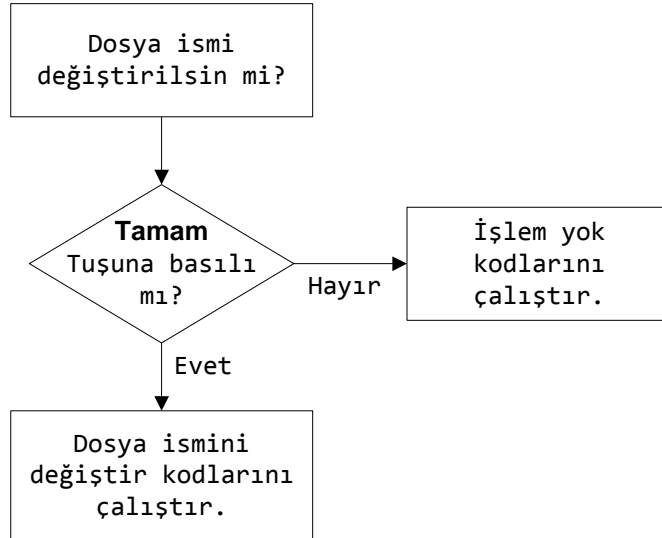
## ARAŞTIRMA

Günlük hayattan basit örnekler araştırınız. Bu basit örneklere ait akış diyagramları oluşturunuz.

## 2. AKIŞ DENETİMİ

### 2.1. Şartlı Dallanma

Şartlı dallanma kavramı program akışının bir ifadenin alacağı değere ya da duruma göre farklı program bloklarının çalışmasını ifade eder. Daha net bir ifadeyle karar verme yapısıdır. Şekil 2.1’de bir şartlı dallanma gösterimi verilmiştir.



Şekil 2.1: Şartlı dallanma akış diyagramı

Yukarıda dosya isiminin değiştirilip değiştirilemeyeceğini sorulmaktadır. Eger Tamam butonuna basılırsa, dosya ismi değiştirilecek, basılmazsa değiştirilmeyecektir.

## 2.1.1. If- Then

Türkçe karşılığı “EĞER..... İSE” anlamına gelen IF – THEN komutu bir şartın sonucunun doğru olması durumunda ne yapılması gerektiğini programa yazmamızı sağlar. IF ile başlayan şart END IF ile bitirilir.

IF –THEN yapısının kullanım formatı şu şekildedir;

```
if şart-cümlesi then  
    kod grubu  
end if
```

**Örnek:** 9 sayısı 5'ten büyükse, “9 sayısı 5'ten büyükmüş” mesajı görüntülensin.

```
If 9 > 5 Then  
    MsgBox("9 sayısı 5'ten büyükmüş")  
End If
```

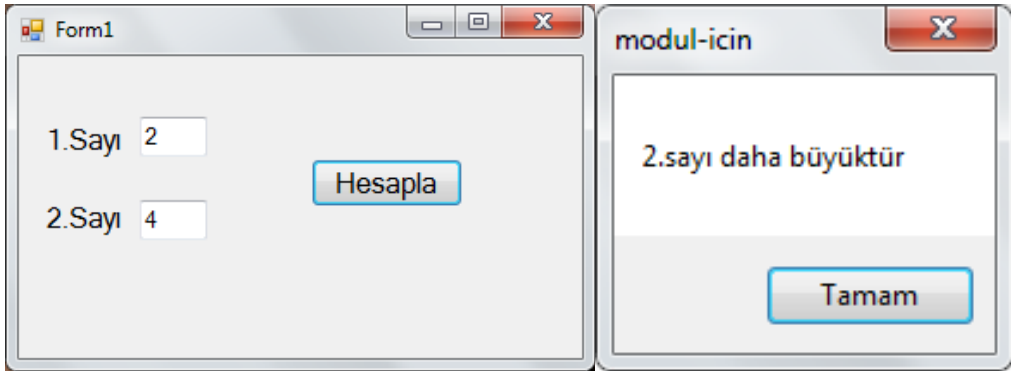
## 2.1.2. If-Then-Else

Şartın 2 adet olduğu durumlarda kullanılır. Örneğin; metin kutularına girilen 2 sayının büyüklüğü karşılaştırılsın. EĞER (IF) 1.sayı, 2.sayıdan büyükse “1.sayısı büyüktür” mesajı görüntülensin, DEĞİLSE (ELSE) “2.sayı büyüktür” mesajı görüntülensin. Örnekte de anlaşılacağı gibi zaten 2 durum mevcuttur.

**Örnek 1:**

Metin kutularına girilen sayıları karşılaştıran programın ekran görüntüsü ve kodları aşağıdadır.

### Ekran Görüntüsü



### Kodlar

```
If TextBox1.Text > TextBox2.Text Then  
    MsgBox("1.sayı daha büyüktür")  
Else  
    MsgBox("2.sayı daha büyüktür")  
End If
```

### 2.1.3. If – Then – Else If

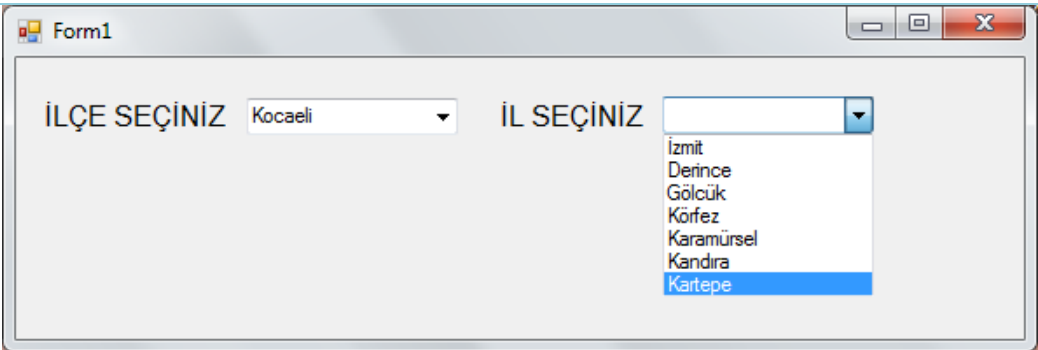
Üçüncü IF yapısı iç içe “eğer” cümlelerinin kurulmasına yardımcı olur. Arka arkaya birbirine bağlı birden fazla şartı sorgulayarak karar vermek için kullanılır.

IF –THEN – ELSE IF yapısının kullanım formatı şu şekildedir;

```
IF şart_cümlesi1 THEN
    Kod Grubu1
ELSE IF şart_cümlesi2 THEN
    Kod Grubu2
ELSE
    Kod Grubu3
END IF
```

**Örnek 2:** İl için tanımlanan liste kutusundan seçilen il adına göre ilçe liste kutusunun içeriği otomatik olarak dolduran program aşağıdadır.

#### Ekran Görüntüsü



The screenshot shows a Windows application window titled "Form1". Inside the window, there are two dropdown menus. The first dropdown is labeled "İLÇE SEÇİNİZ" and has "Kocaeli" selected. The second dropdown is labeled "İL SEÇİNİZ" and has a list of districts open, with "Kartepe" selected. The list of districts includes: İzmit, Derince, Gölcük, Körfez, Karamürsel, Kandıra, and Kartepe.

## Kodlar

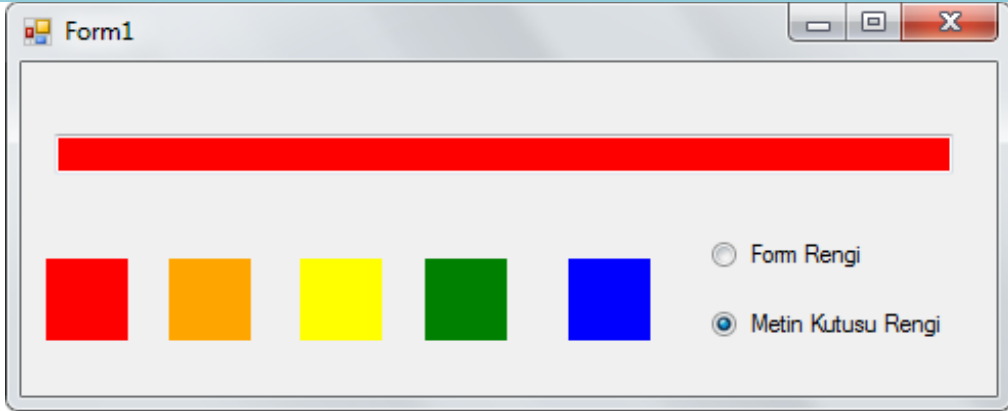
```
Public Class Form1

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
        ComboBox2.Items.Clear() 'İLÇE liste kutusunu temizleyelim

        If ComboBox1.Text = "Ankara" Then
            ComboBox2.Items.Add("Çankaya")
            ComboBox2.Items.Add("Etimesgut")
            ComboBox2.Items.Add("Keçiören")
            ComboBox2.Items.Add("Yenimahalle")
            ComboBox2.Items.Add("Mamak")
        ElseIf ComboBox1.Text = "İstanbul" Then
            ComboBox2.Items.Add("Beşiktaş")
            ComboBox2.Items.Add("Şişli")
            ComboBox2.Items.Add("Eminönü")
            ComboBox2.Items.Add("Beykoz")
            ComboBox2.Items.Add("Adalar")
        ElseIf ComboBox1.Text = "İzmir" Then
            ComboBox2.Items.Add("Karşıyaka")
            ComboBox2.Items.Add("Bornova")
            ComboBox2.Items.Add("Güzelyalı")
            ComboBox2.Items.Add("Buca")
            ComboBox2.Items.Add("Konak")
        ElseIf ComboBox1.Text = "Kocaeli" Then
            ComboBox2.Items.Add("İzmit")
            ComboBox2.Items.Add("Derince")
            ComboBox2.Items.Add("Gölcük")
            ComboBox2.Items.Add("Körfez")
            ComboBox2.Items.Add("Karamürsel")
            ComboBox2.Items.Add("Kandıra")
            ComboBox2.Items.Add("Kartepe")
        End If
    End Sub
End Class
```

**Örnek 3:**

Label (etiket) nesneleri kullanarak oluşturulmuş bir renk paletinin kullanımı. Renk paletleri tıklandığında ya form rengi veya metin kutusunun rengi değişecektir.

**Ekran Görüntüsü****Kodlar**

```
Public Class Form1

    Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label1.Click
        If RadioButton1.Checked Then 'Eğer formun rengi opsiyonu seçili ise
            Me.BackColor = Label1.BackColor
        Else
            TextBox1.BackColor = Label1.BackColor
        End If
    End Sub

    Private Sub Label2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label2.Click
        If RadioButton1.Checked Then
            Me.BackColor = Label2.BackColor
        Else
            TextBox1.BackColor = Label2.BackColor
        End If
    End Sub

    Private Sub Label3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label3.Click
        If RadioButton1.Checked Then
            Me.BackColor = Label3.BackColor
        Else
            TextBox1.BackColor = Label3.BackColor
        End If
    End Sub

    Private Sub Label4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label4.Click
```

```

    If RadioButton1.Checked Then
        Me.BackColor = Label4.BackColor
    Else
        TextBox1.BackColor = Label4.BackColor
    End If
End Sub

Private Sub Label5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label5.Click
    If RadioButton1.Checked Then
        Me.BackColor = Label5.BackColor
    Else
        TextBox1.BackColor = Label5.BackColor
    End If
End Sub
End Class

```

#### 2.1.4. Select Case / Case Else

Birden fazla seçenek içinden seçim yapabilme anlamında oldukça yaygın bir karar verme yapısıdır. İç içe bir sürü IF kodu kullanmak programın hızını yavaşlatır ve hantal bir program haline getirir. Hatta içinden çıkılmaz bir sonuca götürebilir. Bu anlamda “Select Case” yapısı program anlaşılabilirlik ve hız kazandırır.

“Select Case” yapısında seçici adı verilen bir değişken kullanılır. Seçici değişkenin alabileceği değerler ve Case (durum) blokları ile yapılacak olan işlemler sıralanır.

Aşağıda Select Case yapısının formatı verilmiştir.

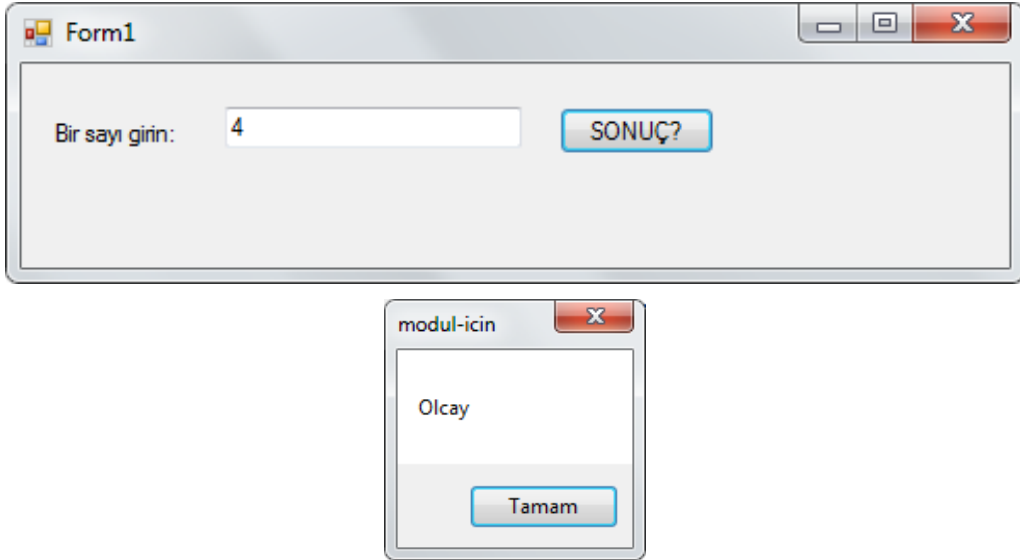
```

Select Case seçici
    Case değer1
        Kod grubu1
    Case değer2
        Kod grubu2
    Case değer3
        Kod grubu3
    Case Else
        Kod grubu X
End Select

```

**Örnek 4:**

Metin kutusuna girilen sayı 1 ise, Ergün  
2 ise, Erhan  
3 ise, Nuran  
4 ise, Olcay  
5 ise, Gönül  
hiçbiriye, “yanlış değer” mesajı görüntülensin.

**Ekran Görüntüsü****Kodlar**

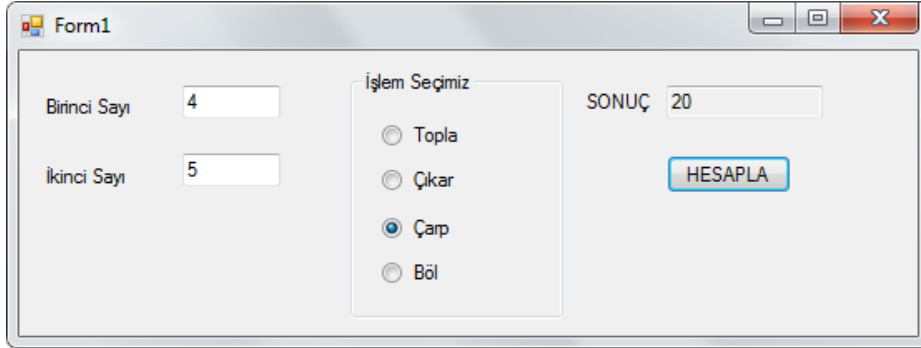
```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Select Case TextBox1.Text
            Case 1
                MsgBox("Ergün")
            Case 2
                MsgBox("Erhan")
            Case 3
                MsgBox("Nuran")
            Case 4
                MsgBox("Olcay")
            Case 5
                MsgBox("Gönül")
            Case Else
                MsgBox("Yanlış değer")
        End Select
    End Sub
End Class
```



## UYGULAMA FAALİYETİ

İki sayı için seçilen işleme (4 işlem) göre hesap yapan programın yazılması.



İşlem Basamakları	Öneriler
<ul style="list-style-type: none"><li>➤ Program için yukarıda ekran görüntüsü verilmiş form tasarımı yapınız.</li></ul>	<ul style="list-style-type: none"><li>➤ Form boyutlarınızı program çıktısını öngörerek ayarlayınız. Tasarladığınız formunuzu öğretmeninize göstererek onun fikrini alınız.</li></ul>
<ul style="list-style-type: none"><li>➤ Form üzerindeki nesne özellikleri ayarlayınız.</li></ul>	
<ul style="list-style-type: none"><li>➤ Aşağıda verilen kodları formunuza uyarlayınız.</li></ul>	<ul style="list-style-type: none"><li>➤ Kod yazım kurallarını hatırlayınız.</li></ul>
<ul style="list-style-type: none"><li>➤ Programınızı çalıştırarak deneyiniz.</li></ul>	
Kodlar	
<pre>Public Class Form1     Dim s1, s2 As Integer     Dim s As Double      Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click         s1 = Sayi1.Text         s2 = Sayi2.Text         If Topla.Checked = True Then             s = s1 + s2         ElseIf cikar.Checked = True Then             s = s1 - s2         ElseIf carp.Checked = True Then             s = s1 * s2         ElseIf bol.Checked = True And s2 &lt;&gt; 0 Then 'Sıfıra bölme yapmasın.             s = s1 / s2         End If         sonuc.Text = s     End Sub End Class</pre>	

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Formu tasarlayabildiniz mi?		
2. Formdaki nesne ayarlarını yapabildiniz mi?		
3. Kodları yazabildiniz mi?		
4. Programı denemek amacıyla çalıştırabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınızı “Evet” ise “Ölçme ve Değerlendirme” ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

- Aşağıda kodu çalıştırıldığında hangi sonuç görüntülenir?  

```
Const a = 23  
Const b = 18  
If a + b > 41 Then  
    MsgBox("merhaba")  
Else  
    MsgBox("selam")  
End If
```

A) Selam      B) Merhaba      C) Const      D) Print
- Else If bloku aşağıdakilerden hangisi ile sonlandırılır?  
A) End      B) End Else      C) Next      D) End If
- Select Case bloku hangi komutla sonlandırılır?  
A) End Select      B) Select End      C) Next Select      D) Exit Select
- Birden fazla seçeneğin değerlendirilmesine ..... komutu kullanılır?  
A) End      B) Else      C) If Else      D) Select Case
- Bir programda IF kodu çoksa, programın hızı o kadar .....?  
A) Azalır      B) Artar      C) Değişmez      D) Hiçbiri
- Select Case bloku hangi komutla sonlandırılır?  
A) End Select      B) Select End      C) Next Select      D) Exit Select

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-3

## AMAÇ

Bir işlemin tamamlanması veya sonucun bulunması için tekrarlanması gerekir. Bu öğrenme faaliyetinde Doo – Loop ve For – Next döngü işlemleri öğreneceksiniz.

## ARAŞTIRMA

- İnternet aracılığı ile Doo – Loop ve For – Next döngü işlemleri hakkında bilgi ediniz.

## 3. DÖNGÜLER

Programda benzer bir işlemi çok defalar tekrar etmek zorunda kalabilirsiniz. Bu bazen belirli bir sayı kadar tekrarlama olur bazen de bir durumun ortaya çıkması ya da şartların oluşmasına kadar geçen süre içinde tekrarlamalar olur. İşin ya da işlerin tekrarlanarak yapılması hâline döngü adı veririz. Döngü bloku içine tekrarlanacak işlem ya da işlemler bir defa yazılır. Program satırlarımız azalır ve daha etkin bir hâl alır.

### 4.1. For - Next

Eğer tekrar bloğu içinde kaç defa döngü yapılacağı belli ise For Next kullanmak çok faydalıdır. For Next döngüsünde sayıcı bir değişken belirlenerek başlangıç ve bitiş değerleri atanarak döngünün kaç kez yapılacağı belirlenir. For Next bloğu arasında kalan kod grubu, belirlenen tekrar sayısı kadar işletilir. Kod grubunun her işletilmesinde sayıcının değeri verilen başlangıç değerinden itibaren artar ya da azalır. Böylelikle sayıcı bitiş değerine ulaştığında döngüden çıkılır.

For – Next döngü yapısı;

**For** sayıcı=başlangıç değeri **to** bitiş değeri **step** adım sayısı

Kod bloğu

**Next** sayıcı

**Örnek 5:**

Aşağıda program girilen sayının istenilen kuvveti ve faktöriyeli hesaplamaktadır. “For-Next” döngü yapısının kullanım örneğidir.

**Ekran Görüntüsü****Kodlar**

```
Public Class Form1

    Dim sayi, kuvvet As Integer
    Dim sonuc As Long
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

        sayi = txtsayi.Text
        kuvvet = txtkuvvet.Text
        sonuc = 1

        For sayac = 1 To kuvvet 'FOR döngüsü 1 den girilen kuvvet kadar
            tekrarlanacak
            sonuc = sayi * sonuc 'Sayı, kuvveti kadar kendisi ile
            çarpılıyor.
        Next (sayac) 'sayac değeri 1 artırılabacak ve FOR döngüsü
        tekrarlanacak
        kuvvet_sonucu.Text = sonuc 'FOR döngüsü bittiğinde işlem sonuc
        değişkenindeki değer kuvvet_sonucu'na atanıyor

    End Sub

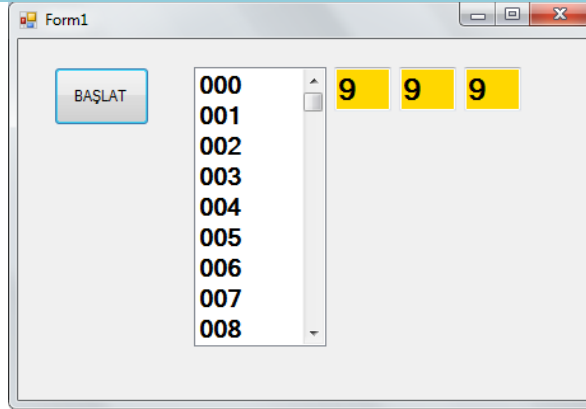
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

        sayi = txtsayi.Text
        sonuc = 1
        For sayac = 1 To sayi
            sonuc = sayac * sonuc 'Sayıya kadar olan bütün sayılar
            çarpılıyor.
        Next (sayac)
        faktoriyel_sonucu.Text = sonuc

    End Sub
End Class
End Class
```

**Örnek 6:**

000 ile 999 arasındaki sayılar iç içe düzenlenmiş üç for-next döngü yapısı kullanılarak oluşturuluyor. Kod içerisinde yer alan “*System.Threading.Thread.Sleep(10)*” ifadesi, her adımda bekleme işlevini gerçekleştiriyor. Dolayısıyla sayaçların değişimi gözlenebiliyor.

**Ekran Görüntüsü****Kodlar**

```
Public Class Form1

    Dim sayi As Integer
    Dim k As Long

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

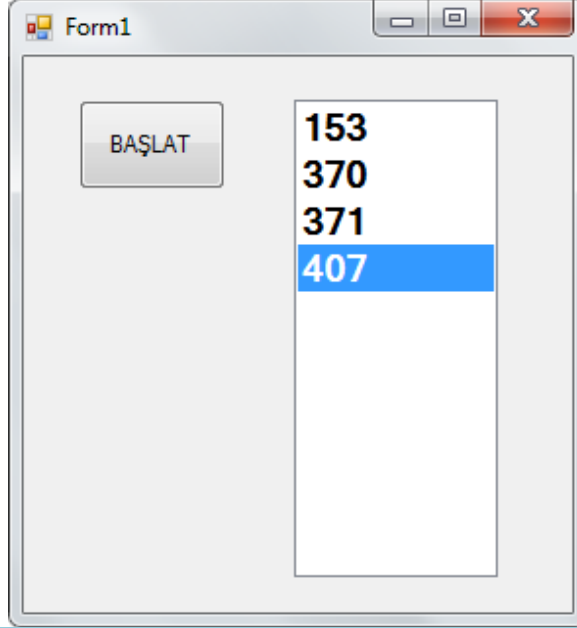
        For sayac1 = 0 To 9
            For sayac2 = 0 To 9
                For sayac3 = 0 To 9
                    ListBox1.Items.Add(sayac1.ToString + sayac2.ToString + sayac3.ToString)
                    TextBox3.Text = sayac3
                    TextBox2.Text = sayac2
                    TextBox1.Text = sayac1
                    TextBox3.Refresh()
                    TextBox2.Refresh()
                    TextBox1.Refresh()
                    System.Threading.Thread.Sleep(10) 'Her adımda 10 ms bekleyecek.
                Next (sayac3)
            Next sayac2
        Next sayac1

    End Sub

End Class
```

**Örnek 7:**

153 Sayısı şöyle bir özelliğe sahiptir:  $153 = 1^3 + 5^3 + 3^3$   
Acaba 100 ile 999 arasında bu özelliği sağlayan, yani basamaklarının küplerinin toplamına eşit olan başka sayı ya da sayılar var mıdır?

**Ekran Görüntüsü****Kodlar**

```
Public Class Form1

    Dim sayi As Integer

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

        For yuzler = 1 To 9
            For onlar = 0 To 9
                For birler = 0 To 9
                    sayi = 100 * yuzler + 10 * onlar + birler
                    If sayi = yuzler ^ 3 + onlar ^ 3 + birler ^ 3 Then
                        ListBox1.Items.Add(sayi)
                    End If
                Next birler
            Next onlar
        Next yuzler

    End Sub

End Class
```

Kaprekar sayıları, 1949 yılında Hintli matematikçi Kaprekar tarafından tariflenen sayılardır.

n basamaklı bir t Kaprekar sayısının karesi alınıp sağdaki n basamağı solda kalan n-1 basamağa eklendiğinde sonuç yine t sayısını verir.

Örnek:

**Örnek 8:**

55, iki basamaklı bir sayıdır.

$55^2 = 3025$ , sağdan iki basamak 25, soldan iki basamak 30.

Bu iki sayının toplamı  $30+25=55$  yani sayının kendisidir.

1, 9, 45, 55, 99, 297, 703, 999, 2223, 2728, 4879 sayıları da diğer bazı Kaprekar sayılarıdır.

#### Ekran Görüntüsü

Form1

Aralık Başlangıcı: 9

Aralık Bitişi: 1000

BAŞLAT

9  
45  
55  
297  
703



## Kodlar

```
Public Class Form1

    Dim sayi, uz, sayi_kare, sagtaraf, soltaraf, basla, bitir,
    sagtaraf_uzunluk, sayi_kare_uzunluk As Long

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button1.Click

        ListBox1.Items.Clear() 'Liste kutusu temizleniyor
        basla = TextBox1.Text ' Bu sayı aralığın başlangıç sayısı ve en az 9
        girilmeli. Sayının karesi 2 basamaklı olmalıdır.
        bitir = TextBox2.Text
        For sayi = basla To bitir
            uz = Strings.Len(sayi.ToString)
            sayi_kare = sayi * sayi
            sagtaraf = Strings.Right(sayi_kare.ToString, uz)
            sayi_kare_uzunluk = Strings.Len(sayi_kare.ToString)
            sagtaraf_uzunluk = Strings.Len(sagtaraf.ToString)
            soltaraf = Strings.Left(sayi_kare.ToString, sayi_kare_uzunluk -
            sagtaraf_uzunluk)
            If sagtaraf + soltaraf = sayi And sayi <> 10 Then
                ListBox1.Items.Add(sayi)
            End If
        Next

    End Sub

End Class
```

## 4.2. Do - Loop

Kuracağımız döngünün tekrarlama sayısı bilinmiyorsa ve döngü işlemi bir şartın ya da durumun gerçekleşmesi hâlinde sonlanacak ise Do-Loop yapısı idealdir. Bu döngü yapısının iki çeşidi vardır. Birincisi, önce blok içinde kalan kod grubunun çalıştırılması ve en sonda şart ya da durumun sorgulanmasıdır. Şart ya da durum gerçekleşene kadar tekrarlama yapılacaksa Until (kadar) takısı, şart ya da durumun sonucu bozulana kadar tekrarlama yapılacak ise While (iken) takısı kullanılır.

Aşağıda her iki takı ile Do - Loop formatı gösterilmektedir.

<b>Do</b> Kod grubu <b>Loop Until</b> şart cümlesi	<b>Do</b> Kod grubu <b>Loop While</b> şart cümlesi
<b>Do Until</b> şart cümlesi Kod grubu <b>Loop</b>	<b>Do While</b> şart cümlesi Kod grubu <b>Loop</b>

Bazı durumlarda program algoritmamız gereği herhangi bir durumun gerçekleşmesine bağlı olarak Do-loop döngüsünden çıkılması gerektiğinde “Exit Do” ifadesi kullanılır. Bu ifade döngünün herhangi bir yerinde kullanılabilir. Genellikle “IF” yapısı içine yerleştirilerek şart cümlesinin gerçekleşme durumuna göre program akışını döngüden çıkarır.

<b>Örnek 9:</b>	<p>1 dâhil, kendisi hariç olmak üzere bölenlerinin toplamına eşit olan sayılara mükemmel sayı adı verilir.</p> <p>Örneğin <math>6=1+2+3</math> olduğundan bir mükemmel sayıdır.</p> <p>Bir sonraki mükemmel sayı 28’dir. Çünkü <math>28=1+2+4+7+14</math> olur.</p> <p>Geliştirdiğimiz uygulama, kullanıcı tarafından girilecek bir sayının mükemmel olup olmadığını test etmekte, değil ise bir sonraki mükemmel sayıyı hesaplamaktadır.</p>
-----------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### Ekran Görüntüsü

The screenshot shows a Windows application window titled 'Form1'. Inside the window, there is a text box labeled 'Sayı Giriniz' containing the number '28'. To the right of the text box is a button labeled 'Sayı Mükemmel mi?'. Below the main window, a smaller dialog box titled 'modul-icin' is open, displaying the message 'Sayı MÜKEMMEL bir sayıdır.' and a 'Tamam' button.

## Kodlar

```
Public Class Form1

    Dim sayi, sayi1, toplam, k As Long

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

        sayi = TextBox1.Text
        toplam = 0
        For k = 2 To sayi / 2
            If sayi Mod k = 0 Then toplam = toplam + k
        Next
        If toplam + 1 = sayi Then
            MsgBox("Sayı MÜKEMMEL bir sayıdır.")
        Else 'Sayı mükemmel değil ise bir sonrakini ara
            sayi1 = sayi
            Do
                toplam = 0
                sayi1 = sayi1 + 1
                For k = 2 To sayi1 / 2
                    If sayi1 Mod k = 0 Then toplam = toplam + k
                Next
            Loop Until toplam + 1 = sayi1
            MsgBox("MÜKEMMEL sayı değil.")
        End If

    End Sub

End Class
```

### Örnek 10:

**Collatz Problemi:** Dolu tanesi sayıları olarak da bilinen problem, girilecek bir pozitif tamsayı ile başlayan bir sayı dizisi oluşturmaya dayanır. Dizinin her bir elemanı, bir önceki sayı kullanılarak şu kural dâhilinde hesaplanır: Eğer sayı çift ise bir sonraki sayı bu sayının yarısı, tek ise üç katının bir fazlası olacaktır. Böylece ilk sayısı verilen bir sayı dizisi oluşturulur. Örneğin, ilk sayı 7 ise dizi şu şekilde oluşacaktır: 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1,...

Ya da ilk sayı 21 ise,  
21, 64, 32, 16, 8, 4, 2, 1, ...

Görüldüğü gibi dizi 1 sayısına ulaştığı anda sonsuz döngüye girmektedir. Aslında başlangıç sayısı olarak ne girersek girelim dizi her zaman 1 sayısına ulaşmaktadır, ama bu durum henüz matematiksel olarak kanıtlanamamıştır.

## Ekran Görüntüsü

Form1

İlk Sayı Giriniz 9

BAŞLAT

Adım Sayısı 20

En Büyük Değer 52

9  
28  
14  
7  
22  
11  
34  
17  
52  
26  
13  
40  
20  
10  
5  
16  
8  
4  
2  
1

## Kodlar

```
Public Class Form1

    Dim sayi, sayac, enbuyuk As Long

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

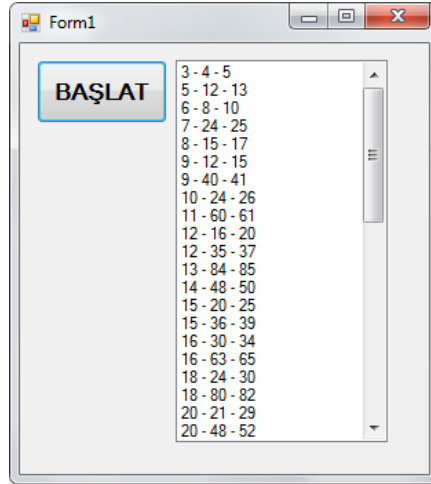
        sayi = TextBox1.Text
        ListBox1.Items.Clear()
        sayac = 0
        enbuyuk = 0
        Do
            ListBox1.Items.Add(sayi)
            If sayi Mod 2 = 0 Then 'Sayı çift ise.
                sayi = sayi / 2
            Else 'Sayı tek ise.
                sayi = sayi * 3 + 1
            End If
            sayac = sayac + 1
            If sayi > enbuyuk Then enbuyuk = sayi 'sayi enbuyuk'ten daha büyük
            ise yeni enbuyuk değeri sayi olsun.
        Loop Until sayi = 1
        ListBox1.Items.Add(1)
        TextBox2.Text = sayac + 1
        TextBox3.Text = enbuyuk

    End Sub

End Class
```

## UYGULAMA FAALİYETİ

**Pisagor ölçüleri:**  $a^2 + b^2 = c^2$  özelliğini sağlayan sayılar, dik üçgenin kenar uzunluklarını gösterirler ve Pisagor üçlüleri olarak anılırlar. 1 ile 99 arasındaki doğal sayılardan hangilerinin bu eşitliği sağladığını bulan programı yazınız.



İşlem Basamakları	Öneriler
➤ Program için yukarıda ekran görüntüsü verilmiş form tasarımı yapınız.	➤ Form boyutlarınızı program çıktısını öngörerek ayarlayınız. Tasarladığınız formunuzu öğretmeninize göstererek onun fikrini alınız.
➤ Form üzerindeki nesne özellikleriyarlayınız.	
➤ Aşağıda verilen kodları formunuza uyarlayınız.	➤ Kod yazım kurallarını hatırlayınız.
➤ Programınızı çalıştırarak deneyiniz.	
Kodlar	
<pre>Public Class Form1      Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click          For a = 1 To 99             For b = 1 To 99                 For c = 1 To 99                     If (c ^ 2 = a ^ 2 + b ^ 2) And (a &lt; b) Then                         ListBox1.Items.Add(a.ToString + " - " + b.ToString + " - " + c.ToString)                     Next c                 Next b             Next a         End Sub     End Class</pre>	

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Formu tasarlayabildiniz mi?		
2. Formdaki nesne ayarlarını yapabildiniz mi?		
3. Kodları yazabildiniz mi?		
4. Programı denemek amacıyla çalıştırabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınızı “Evet” ise “Ölçme ve Değerlendirme” ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

1. Program parçasındaki noktalı kısma aşağıdakilerden hangisi gelmelidir?

```
For i = 1 To 10  
    toplam = toplam + 1  
    .....
```

- A) End                      B) Next i                      C) Next toplam                      D) End For
2. Aşağıdaki döngü sonucunda topla değişkeninin içeriği kaç olur?

```
topla=0  
i=0  
Do While i < 5  
    topla = topla + 2  
    i = i + 1  
Loop
```

- A) 6                      B) 8                      C) 10                      D) 20
3. Aşağıdakilerden hangisi “Visual Basic”te kullanılan bir döngü tipi değildir?
- A) Do-Loop While                      B) Do Until –Loop                      C) While-For                      D) For-Next
4. Döngünün tekrarlama sayısı bilinmiyorsa ve döngü işlemi bir şartın ya da durumun gerçekleşmesi hâlinde sonlanacak ise ..... yapısı idealdir.
- A) Do-Loop                      B) Until –Loop                      C) While-For                      D) For-Next
5. Şart ya da durum gerçekleşene kadar tekrarlama yapılacaksa ..... takısı kullanılır.
- A) Do                      B) Until                      C) While                      D) Next
6. Şart ya da durumun sonucu bozulana kadar tekrarlama yapılacak ise ..... takısı kullanılır.
- A) Do                      B) Until                      C) While                      D) Next
7. Aşağıdaki kod bloğundaki SONUC değişkeninin değeri kaçtır?

```
For a = 1 To 3  
    For b = 1 To 2  
        For c = 1 To 3  
            Sonuc = sonuc + 1  
        Next c  
    Next b  
Next a
```

- A) 5                      B) 8                      C) 15                      D) 18

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-4

## AMAÇ

Görsel programlamada dizi yapısını doğru olarak kullanabileceksiniz.

## ARAŞTIRMA

- Matematiksel anlamda dizi kavramını araştırınız.

## 4. DİZİLER

### 4.1. Dizi Nedir?

Diziler, aynı tipteki birden çok veriyi bir arada tutmaya yarar. Dizideki tüm elemanların veri tipi aynıdır. Yani 10 elemanlı bir dizinin tüm elemanları ya string ya integer ya da diğer veri tiplerindedir.

Aşağıdaki takım dizilişinde Umut isimli oyuncunun dizi içindeki yeri 1, emre isimli oyuncunun ise 5'tir. Bu diziliş sırasına programcılıkta "indis" adı verilir.

0	Ahmet
1	Umut
2	Tolga
3	Serkan
4	Fatih
5	Emre

Dizi indisinin başlangıç numarası aksi belirtilmediği sürece 0'dır.

### 4.2. Tek Boyutlu Diziler

Tek boyutlu (indisli) Kullanım şekli şu şekildedir:

**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. elemanı ve ad(4) ifadesi dizinin 5. elemanıdır.



Dizi elemanlarına deęer aktarmak için ařađıdaki gibi bir kod yazılır;

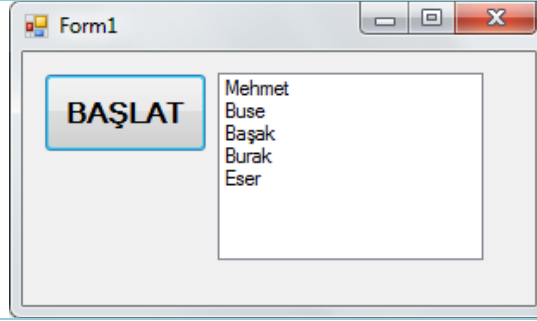
```
ad(0) = "Mehmet"  
ad(1) = "Buse"  
ad(2) = "Bařak"  
ad(3) = "Burak"  
ad(4) = "Eser"
```

İstenirse dizi tanımlanırken dizi elemanlarına deęerleri aktarılabilir. Bu tanımlamada dizi deęiřkenine isim verirken parantez içinde dizinin eleman sayısı belirtilmez.

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

**Örnek 11:** 5 elemanlı **ad** dizi deęiřkeni tanımlanmış, dizi deęiřkeninin her elemanına bir deęer verilmiş ve son olarak dizi deęiřkeninin her elemanı ekrana yazdırılmıştır.

#### Ekran Görüntüsü



#### Kodlar

```
Public Class Form1  
  
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
        Dim ad(4) As String  
        ad(0) = "Mehmet"  
        ad(1) = "Buse"  
        ad(2) = "Bařak"  
        ad(3) = "Burak"  
        ad(4) = "Eser"  
  
        Dim i = 0  
        Do Until i = 5  
            ListBox1.Items.Add(ad(i))  
            i = i + 1  
        Loop  
  
    End Sub  
  
End Class
```

### 4.3. Tek Boyutlu Diziler

Yukarıdaki örnekte sadece isimler bir dizi şeklinde sıralanmıştı. İsimlerin yanı sıra doğum yerlerini de diziye dâhil edersek aşağıdaki gibi bir tablo elde ederiz. Tabloyu incelersek 2 sütun 5 satırdan oluştuğunu görebiliriz.

	0	1
0	Mehmet	1996
1	Buse	1995
2	Başak	1990
3	Burak	1982
4	Eser	1973

Çift indisli bir dizi tanımlaması şu şekilde yapılır;

**Dim** dizinin\_adı (sıra sayısı, sütun sayısı) as tip

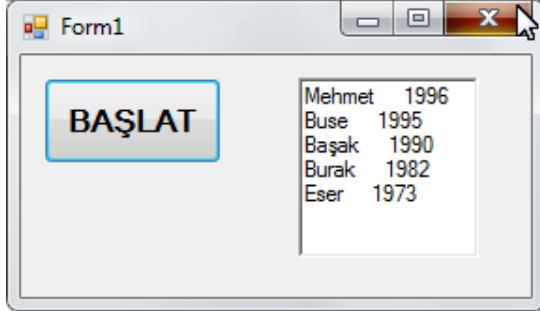
Yukarıdaki takım tabloyu çift indisli dizi olarak tanımlarsak;

```
Dim isanlar = {"Mehmet", 1996}, {"Buse", 1995}, {"Başak", 1990}, {"Burak", 1982}, {"Eser", 1973}
```

Örnek 12: Aşağıdaki tabloyu ekranda görüntüleyen programın yazılması.

	0	1
0	Mehmet	1996
1	Buse	1995
2	Başak	1990
3	Burak	1982
4	Eser	1973

**Ekran Görüntüsü**



## Kodlar

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim isanlar = {"Mehmet", 1996}, {"Buse", 1995}, {"Başak", 1990},
{"Burak", 1982}, {"Eser", 1973}}

        For index0 = 0 To isanlar.GetUpperBound(0)
'isanlar.GetUpperBound(0): Dizinin 1.satırını okunuyor (İsimler)
            For index1 = 0 To isanlar.GetUpperBound(1)
'isanlar.GetUpperBound(1): Dizinin 2.satırını okunuyor (Doğum tarihleri
                RichTextBox1.Text = RichTextBox1.Text + isanlar(index0,
index1).ToString & "      "
            Next
        Next
    End Sub

End Class
```

## 4.4. Statik Diziler

Statik kelimesi durağan, sabit anlamlarına gelir. Eğer dizinin kaç elemandan oluşacağı biliniyorsa bu tür dizilere statik dizi adı verilir. Bellekte eleman sayısı kadar yer ayrılır ve program sonlanıncaya kadar değişmez.

Aşağıdaki örnekte 6 elemanlı “akşam menü” isimli bir dizi vardır.

	AKŞAM MENÜ		AKŞAM MENÜ
0	Çorba	0	Çorba
1	Izgara tavuk	1	Et sote
2	Pilav	2	Makarna
3	Salata	3	[BOŞ]
4	Kazandibi	4	Keşkül
5	Türk Kahvesi	5	Meyve suyu

Bu dizinin tanımlama satırı aşağıdaki gibi yazılır.

```
Dim akşam_menü(6) As Object
```

## 4.5. Dinamik Diziler

Her zaman dizi elemanlarının sayısı önceden bilinmeyebilir. Programın çalışması esnasında algoritma gereği değişkenlik gösterebilir. Visual Basic dilinde eleman sayısı değişkenlik gösteren diziler için dinamik dizi tanımlaması yapılır. Statik dizilerde bellekte ayrılan yer sabit olduğu için kullanılsa da kullanılmasa da yer kaplar. Dinamik dizilerde ise bellekten gerektiği kadar yer kullanıldığı için verimli kullanılmış olur.

	Şehir		Şehir		Şehir
1	İ	1	U	1	T
2	Z	2	Ş	2	R
3	M	3	A	3	A
4	İ	4	K	4	B
5	R			5	Z
				6	O
				7	N

Yukarıdaki tabloda “şehir” isimli dizinin üç farklı içerik durumu gösterilmiştir. “İZMİR” kelimesi 5 harften oluşmaktadır ve diziyeye aktarıldığında dizi boyutu 5 olur. “UŞAK” ise; 4 harflidir ve program içinde diziyeye bu bilgi aktarıldığında dizinin boyutu 4 olur ve bellekte bu kadar bir yer kaplar.

Dinamik dizilerin tanımlaması yapılırken dizi boyu verilmez. Programın çalışması esnasında alınan boyut bilgisi “ReDim” ifadesiyle dizi boyu olarak atanır.

ukarıdaki örnek için tanımlama ve boyut atama şu şekilde yapılır.

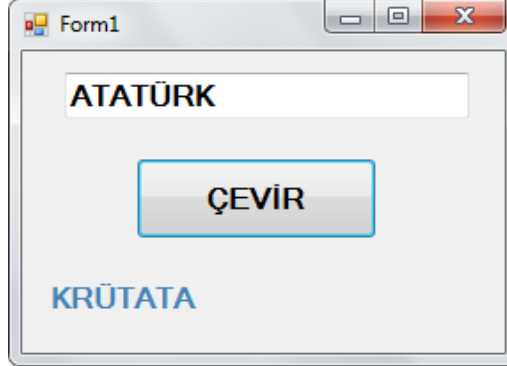
```
Public şehir() as string
```

Kod içinde girilen “a” değerinin dizi boyu olarak atanması için;

```
ReDim şehir(a)
```

**Örnek 13:** Girilen metni tersine çeviren program.

### Ekran Görüntüsü



### Kodlar

```
Public Class Form1

    Public metin() As Object

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

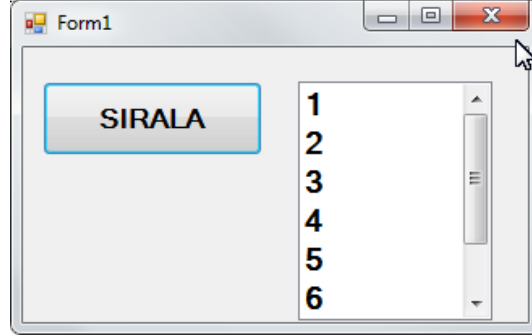
        Dim düz, ters, x As String
        Dim i As Integer
        düz = TextBox1.Text
        x = Len(düz) 'Len komutu kelime veya cümlenin karakter sayısını hesaplar. a=Len("merhaba") ise, a değeri 7 dir.
        ReDim metin(x) '
        For i = x To 1 Step -1
            metin(i) = Mid(düz, i, 1) ' Mid(): Bir karakter dizisinin içinde belli sayıda bir eleman grubunu kesip(alabilir). Mid("kelime", başlangıç, adet) başlangıç : soldan itibaren ilk harfin sıra numarası adet : alınacak harf sayısı
            ters = ters + metin(i)
        Next i
        Label1.Text = ters

    End Sub

End Class
```

## UYGULAMA FAALİYETİ

Karışık sayılardan oluşan bir dizinin elemanlarını küçükten büyüğe sıralayan programı yazınız.



İşlem Basamakları	Öneriler
<ul style="list-style-type: none"><li>➤ Program için yukarıda ekran görüntüsü verilmiş form tasarımı yapınız.</li></ul>	<ul style="list-style-type: none"><li>➤ Form boyutlarınızı program çıktısını öngörerek ayarlayınız.</li><li>➤ Tasarladığınız formunuzu öğretmeninize göstererek onun fikrini alınız.</li></ul>
<ul style="list-style-type: none"><li>➤ Form üzerindeki nesne özellikleri ayarlayınız.</li></ul>	
<ul style="list-style-type: none"><li>➤ Aşağıda verilen kodları formunuza uyarlayınız.</li></ul>	<ul style="list-style-type: none"><li>➤ Kod yazım kurallarını hatırlayınız.</li></ul>
<ul style="list-style-type: none"><li>➤ Programınızı çalıştırarak deneyiniz.</li></ul>	
Kodlar	
<pre>Public Class Form1      Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click          Dim sayilar = {8, 5, 4, 3, 6, 2, 7, 1}         Array.Sort(sayilar) 'Sayılar sıraya alınıyor         For index0 = 0 To sayilar.GetUpperBound(0)             ListBox1.Items.Add(sayilar(index0))         Next      End Sub  End Class</pre>	

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

<b>Değerlendirme Ölçütleri</b>	<b>Evet</b>	<b>Hayır</b>
1. Formu tasarlayabildiniz mi?		
2. Formdaki nesne ayarlarını yapabildiniz mi?		
3. Kodları yazabildiniz mi?		
4. Programı denemek amacıyla çalıştırabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “Evet” ise “Ölçme ve Değerlendirme” ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. “ReDim tanımlama deyimi ..... için kullanılır.”  
A) Tekrar bir dizi tanımlamak B) Dizinin bellekte kapladığı yeri uzatmak  
C) İki boyutlu bir dizi tanımlamak D) Dinamik dizinin boyunu belirlemek

2. Aşağıdaki ifadelerden hangisi yanlıştır?  
A) Dinamik dizilerin uzunluğu değişkendir.  
B) Diziler bellekte yer kaplar.  
C) Dizilerin tipi her zaman string’dir.  
D) Haftanın günleri tek boyutlu bir dizidir

3. Aşağıdaki matris şeklinde gösterilen A dizisinde A(3,2) indisli eleman hangisidir?

45	0	20	11	30
82	99	32	35	40
75	77	54	90	50
42	67	3	78	60

- A) 77 B) 54 C) 32 D) 99
4. Soru 3’te verilen A dizisinin tanımlama satırı aşağıdakilerden hangisinde doğru verilmiştir?  
A) Public A(4x5) as string B) Public A(5x4) as integer  
C) Public A(5,4) as integer D) Public A(4,5) as integer
5. “Bir dizide kaç eleman olduğunu ..... fonksiyonu ile buluruz?”  
A) Number() B) Ubound() C) Mid() D) Len()
6. Aşağıdakilerden hangisi haftanın günlerini bir dizi olarak doğru bir şekilde tanımlanmıştır?  
A) Dim günler(7) as string B) günler(x) as dim string  
C) As dim string günler(7) D) Dim(7) string as günler



7. Aşağıdaki programın çalışması sonucu Form1 üzerindeki TextBox1 isimli metin kutusunda seçeneklerde verilenlerden hangisi görülür?

```
Dim z As Variant
Dim uz As Byte
z="AKINCILAR"
s= Mid(z,4,3)
TextBox1.Text=s
```

- A) AKI                      B) AKIN                      C) NCI                      D) CILAR
8. Aşağıdakilerden hangisi çift boyutlu bir dizi olabilir?
- A) Bir istasyona uğrayan trenlerin isimleri ve kalkış saatleri  
B) Bir basketbol takımındaki oyuncular  
C) 10/A sınıfının öğrenci listesi  
D) Bilgisayarı oluşturan donanım elemanlarının listesi

# ÖĞRENME FAALİYETİ-5

## AMAÇ

Görsel programlama alt yordam oluşturma işlemini doğru olarak yapabileceksiniz.

## ARAŞTIRMA

- Prosedür (alt yordam) kavramını interneti kullanarak araştırıp yorumlayınız.

## 5. PROSEDÜR (ALT YORDAM) OLUŞTURMA

### 5.1. Prosedürler (Alt Yordamlar)

Hangi dille olursa olsun program yazarken algoritma gereği benzer bir işlemi programın farklı yerlerinde tekrar tekrar yazmak zorunda kalabilirsiniz. Örneğin, bir hastane programı yazdığınızı ve programınızın farklı yerlerinde hasta kayıt kabul işlemi yaptırdığınızı düşününüz. Kayıt edilme işleminde yapılması gerekenler hep aynı ise ve her hasta için bu işlemin yapılması gerekiyorsa, her hasta kabulü için kod yazmamız imkânsızdır. Çünkü gelen hasta sayısı belirsizdir. Yapmanız gereken, ortak bir kayıt kabul kodları yazmak ve her hasta kaydı için bu kodların çalışmasını sağlamaktır.

Tek bir defa yazarak programımızın gerektirdiği yerlerinde çağrılan kod bloklarına alt yordam ya da prosedür adı verilir. Alt yordamları kendi başına çalışabilen küçük programlar olarak denilebilir. Bu küçük programcılara birer isim verilir ve ana programda ihtiyaç olduğunda ismiyle çağrılır. Böylelikle yazılan programlar daha kısa, anlaşılır ve hızlı olacaktır, çünkü sadece gerekli olduğunda bu alt programcılar çalışacaktır.

Alt yordam yazma kuralı;

```
Sub altyordam_ismi (değişken1 as tip , değişken2 as tip,  
.....)  
    kodlar  
End Sub
```

Alt yordam ifadesi **Sub** ile başlar **End Sub** ile biter.

Pisagor teoremi:  $a = \sqrt{b^2 + c^2} = (b^2 + c^2)^{0.5}$

#### Aritmetik operatörler:

Toplama, çıkarma, çarpma gibi aritmetik işlemlerde kullanılan işaretlerdir.

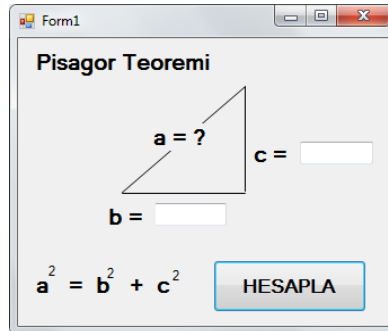
Örnek 14:

İşlem	VB karşılığı
Toplama	+
Çıkarma	-
Çarpma	*
Bölme	/
Üs alma	^

Matematik	VB karşılığı
$-2r+3,26$	$-2*r+3.26$
$ax^{2-b} - 267,009$	$a*x^(2-b)-267.009$

#### Ekran Görüntüsü



#### Kodlar

```
Public Class Form1
    Dim a, b, c As Integer

    Sub hesapla()'alt yordam oluşturuluyor
        a = (b ^ 2 + c ^ 2) ^ 0.5 'pisagor teoremi teoremi
        Label17.Text = a
    End Sub 'alt yordam bitti

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

        b = TextBox1.Text
        c = TextBox2.Text
        hesapla()'alt yordam çağrıldı

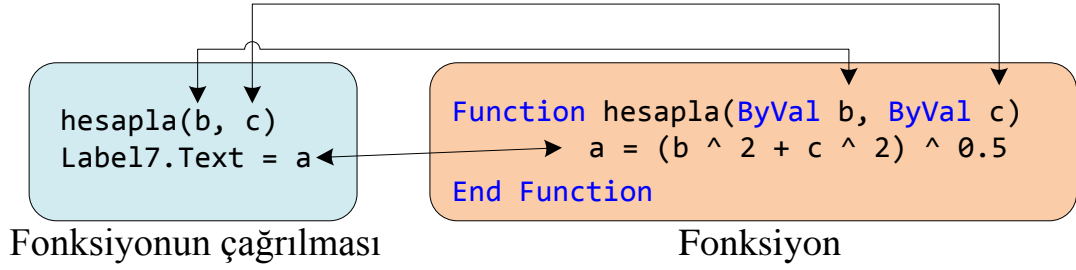
    End Sub

End Class
```

## 5.2. Fonksiyonlar

Alt yordamlar ana programa deęer göndermez. Bu, “Ana program alt yordamı çağırıldığında alt yordam varsa deęerleri alır, işini yapar ve bitirir.” demektir. Ana program tarafından kullanılmak üzere herhangi bir deęer üretmez.

Fonksiyonlar ise adı üzerinde işlevseldir. Geliştirilen algoritma doğrultusunda gerekiyorsa, ana programdan deęerler alır, bir sonuç üretir ve bunu kullanması için tekrar ana programa geri döndürür.



Resim 5.1: Fonksiyonun çağırılması

<b>Örnek 15:</b>	Bir ürünün satış fiyatı şu şekilde hesaplanacaktır. $\text{Satış fiyatı} = \text{Alış fiyatı} + (\text{Alış fiyatı} * \% \text{kar oranı}) + (\text{Alış fiyatı} + (\text{Alış fiyatı} * \% \text{kar oranı})) * \% \text{KDV}$
<b>Ekran Görüntüsü</b>	
<p>Satış Fiyatı Hesaplama</p> <p>Alış Fiyatı 100</p> <p>Kar Oranı 20</p> <p>KDV % 18</p> <p>Satış Fiyatı 141,6</p> <p>HESAPLA</p>	

## Kodlar

```
Public Class Form1
    Dim a, b, c, satis_fiyati As Single 'küsürlü sayıları göstermesi için
    Single olarak tanımlandı

    Function hesapla(ByVal a, ByVal b, ByVal c)
        satis_fiyati = (a + (a * b / 100)) + (a + (a * b / 100)) * (c / 100)
    'Satış fiyatı

    End Function

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button1.Click

        a = TextBox1.Text 'alış fiyatı
        b = TextBox2.Text '% kar oranı
        c = TextBox3.Text '% KDV

        hesapla(a, b, c) 'Fonksiyon çağrılıyor ve işletiliyor
        TextBox4.Text = satis_fiyati
    End Sub

End Class
```

## 5.3. Hazır Fonksiyon Örnekleri

Visual Basic'de çok kullanılan standart fonksiyonlar vardır. Bu fonksiyonların bir kısmı alfa-nümerik fonksiyon bir kısmı da nümerik fonksiyonlardır.

### 5.3.1. Alfa Nümerik Fonksiyonlar (Karakter Dizisi (String))

#### 5.3.1.1. Len() Fonksiyonu

Bu fonksiyon yardımı ile alfa-nümerik değişkenlerin veya ifadelerin karakter sayısını, nümerik ifadelerin ise bellekte kapladıkları alan miktarını (byte) öğrenebiliriz.

#### Örnek:

```
ders = "Visual Basic"
Uzunluk = Len(ders) 'uzunluk 12 olur.
Uzunluk = Len("Visual Basic") 'uzunluk 12 olur.
```

### 5.3.1.2. Left() Fonksiyonu

Bu fonksiyon verilen değişkenin içinde soldan başlayarak istenilen karakterdeki kısmı ayırmak için kullanılır.

**Microsoft.VisualBasic.Left** (değişken, sol baştan itibaren alınacak karakter sayısı)

**Örnek:**

```
Dim sehir, ilkharf, ilkikiharf As String
sehir = "Eskişehir"
ilkharf = Microsoft.VisualBasic.Left(sehir, 1)      'ilkharf E olur.
ilkikiharf = Microsoft.VisualBasic.Left(sehir, 2)   'İlkikiharf Es olur
```

### 5.3.1.3. Right() Fonksiyonu

Bu fonksiyon, verilen değişkenin içinde sağdan başlayarak istenilen karakterde kısmı ayırmak için kullanılır. Değişken çeşitli tiplerde olur geriye dönen değer isteğe göre belirlenebilir.

**Microsoft.VisualBasic.Right** (değişken, sağ baştan itibaren alınacak karakter sayısı)

**Örnek:**

```
Dim sehir, ilkharf, ilkikiharf As String
sehir = "Eskişehir"
ilkharf = Microsoft.VisualBasic.Right(sehir, 1)     'ilkharf r olur.
ilkikiharf = Microsoft.VisualBasic.Right(sehir, 2)  'İlkikiharf ir
olur
```

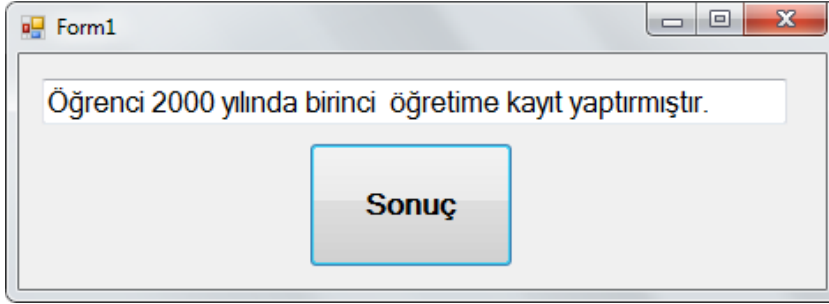
### 5.3.1.4. Mid() Fonksiyonu

Bu fonksiyon kendisine verilen string veya variant tipindeki değişken içerisinden bir kısmı çıkarıp ayırmak için kullanılır.

**Mid**(değişken, başlangıç karakteri, alınacak karakter uzunluğu)

**Örnek:**

```
Dim numara As Object, kayityili As String
Dim ogretim As String
numara = 151420001045.0#
kayityili = Mid(numara, 5, 4)
If Mid(numara, 9, 1) = "1" Then
    ogretim = "birinci"
Else
    ogretim = "ikinci"
End If
TextBox1.Text = "Öğrenci " & kayityili & " yılında " & ogretim & "
öğretime kayıt yaptırmıştır."
```



Resim 5.1: Örneğin ekran görüntüsü

### 5.3.1.5. LTrim() Fonksiyonu

Bu fonksiyon, verilen değişke içinde sol tarafında bulunan boşluk karakterle-rini siler.

**LTrim**(değişken)

**Örnek:**

```
Dim sehir As String
sehir = "    Eskişehir" '5 karakter boşluktan sonra Eskişehir
yazılmıştır.
sehir = LTrim(sehir)
TextBox1.Text = sehir 'Sonuç: 5 boşluk karakter silindikten sonra
sadece Eskişehir sonucu kalır.
```

### 5.3.1.6. RTrim() Fonksiyonu

Bu fonksiyon, verilen değişke içinde sağ tarafında bulunan boşluk karakterlerini siler.

**RTrim**(değişken)

**Örnek:**

```
Dim sehir As String
sehir = "Eskişehir    " 'Eskişehir den sonra '5 karakter boşluk
yazılmıştır.
sehir = RTrim(sehir)
TextBox1.Text = sehir 'Sonuç: 5 boşluk karakter silindikten sonra
sadece Eskişehir sonucu kalır.
```

### 5.3.1.7. Trim() Fonksiyonu

Bu fonksiyon, yukarıda bahsedilen her iki fonksiyonun birleşimi gibidir. Değişken içinde bulunan boşluk karakterlerini her iki taraftan siler.

**Trim**(değişken)

### Örnek:

```
Dim mesaj As String
mesaj = "    Bu fonksiyonlar kelime aralarındaki boşlukları silmez.  "
TextBox1.Text = Trim(mesaj) 'Baştaki ve sondaki boşluk karakterleri
silindi.
```

Bu fonksiyon genellikle dosyadan bilgi okunduğunda veya kullanıcının girmiş olduğu değerler işlenirken kullanılır. Yanlış girilmiş değerler dosyaya ya da veritabanına kaydedildiğinde sorgu yaparken sorunlar çıkabilir (Sıralama işlemleri veya herhangi bir harf ile başlayan kayıtların bulunması gibi).

#### 5.3.1.8. LCase( ) Fonksiyonu

Bu fonksiyon kendisine verilen string tipindeki değişken içerisinde bulunan ifadede BÜYÜK yazılmış harfleri, küçük harfe çevirir.

LCase(değişken)

### Örnek:

```
Dim mesaj As String
mesaj = "TÜM harFLer KÜçük olacAK"
TextBox1.Text = LCase(mesaj) 'tüm harfler küçük olacak.
```

#### 5.3.1.9. UCase( ) Fonksiyonu

Bu fonksiyon kendisine verilen string tipindeki değişken içerisinde bulunan ifadede küçük yazılmış harfleri, BÜYÜK harfe çevirir.

UCase(değişken)

### Örnek:

```
Dim mesaj As String
mesaj = "tüm harfler büyük olacak"
TextBox1.Text = UCase(mesaj) 'TÜM HARFLER BÜYÜK OLACAK.
```

#### 5.3.1.10. Str() Fonksiyonu

Bu fonksiyon verilen herhangi bir tipteki sayısal değeri string tipindeki değişkene çevirir.

Str(Sayısal değer)



### Örnek:

```
TextBox1.Text = Str(123.456) '123.456 sayısı artık bir metin biçimindedir
```

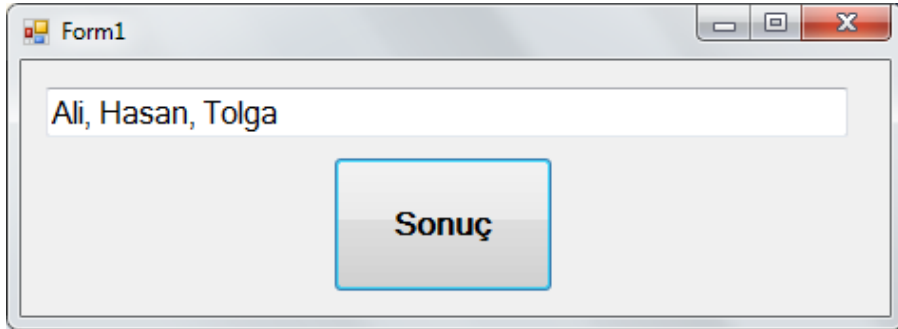
### 5.3.1.11. Joing Fonksiyonu

Bu fonksiyon tek boyutlu bir dizide depolanmış değerleri tek bir değişkene ayrıç olarak belirtilen karakterle birbirinden ayrılarak aktarılmasını sağlar.

**Join**(Birleştirilecek dizi, ayrıç karakteri)

### Örnek:

```
Dim Deneme() As String = {"Ali", "Hasan", "Tolga"}  
Dim DenemeSonucu As String = Join(Deneme, ", ") ' 3 dizibirleştirildi  
ve sonuç: "Ali, Hasan, Tolga"  
TextBox1.Text = DenemeSonucu
```



Resim 5.2: Örneğin ekran görüntüsü

### 5.3.1.12. Split() Fonksiyonu

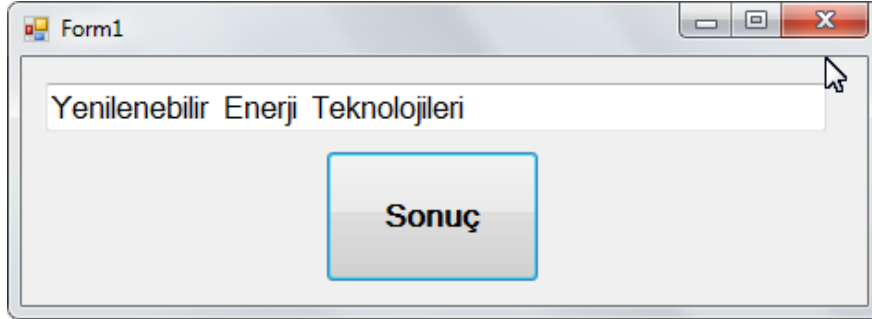
Bu fonksiyon Join fonksiyonunun tam tersi işi yapar. Belirli bir karakter ile birbirinden ayrılmış tek bir değişkenin içinde depolanmış verileri tek boyutlu bir dizi değişkene aktarır.

**Split**(değişken, ayraç karakteri, eleman sayısı, karşılaştırma şekli)

Split Fonksiyonu	Sonuç Değeri
<code>Split("42, 12, 19")</code>	<code>{"42", "12", "19"}</code>
<code>Split("42, 12, 19", ",")</code>	<code>{"42", "12", "19"}</code>
<code>Split("42, 12, 19", ", ", 2)</code>	<code>{"42", "12, 19"}</code>
<code>Split("192.168.0.1", ".")</code>	<code>{"192", "168", "0", "1"}</code>
<code>Split("Alice and Bob", " AND ")</code>	<code>{"Alice and Bob"}</code>
<code>Split("Alice and Bob", " AND ", ,CompareMethod.Text)</code>	<code>{"Alice", "Bob"}</code>
<code>Split("someone@example.com", "@",1)</code>	<code>{"someone@example.com"}</code>
<code>Split("someone@example.com", "@",2)</code>	<code>{"someone", "example.com"}</code>

**Örnek:**

```
Dim DenemeSonucu As String = "Yenilenebilir - Enerji - Teknolojileri"  
Dim TestArray() As String = Split(DenemeSonucu, "-")  
Dim sayac As Integer = -1  
For i As Integer = 0 To TestArray.Length - 1  
    If TestArray(i) <> "" Then  
        sayac += 1 'sayac 1 arttırılıyor  
        TestArray(sayac) = TestArray(i) 'Kelimeler belirleniyor  
    End If  
    TextBox1.Text = TextBox1.Text + TestArray(sayac) 'Sonuç:  
Yenilenebilir Enerji Teknolojileri  
Next  
ReDim Preserve TestArray(sayac)
```



Resim 5.3: Örneğin ekran görüntüsü

### 5.3.1.13. Date, Time, Now Zaman Fonksiyonları

Bu fonksiyonlar sistemin fonksiyonun kullanıldığı andaki tarih, saat bilgilerini öğrenmemizi sağlar.

`Date.Today`: Günün tarihini gösterir.  
`TimeOfDay` : Saati gösterir.  
`Now`: 0 anki tarih ve saati gösterir.

### Örnek:

```
TextBox1.Text = "Tarih: " + Date.Today + " Saat: " + TimeOfDay + " Şu  
an: " + Now
```

## 5.3.2. Nümerik Fonksiyonlar (.Sayısal Fonksiyonlar)

### 5.3.2.1. Aritmetik ifadeler

Sabit ve değişkenlerin aritmetik operatörler (toplama, çarpma, üs, ...) ve parantez kullanılarak bağlanmaları ile bir aritmetik ifade oluşur.

Matematik	VB karşılığı	Açıklama
$-2r+3,26$	$-2*r+3.26$	Çarpma işareti daima yazılır.
$ax^{2-b} - 267,009$	$a*x^(2-b)-267.009$	İlk önce parantez içi hesaplanır.
$\left[ \frac{b(a+b)^2}{3,8} - c \right]^5$	$(b*(a+b)^2/3.8-c)^5$	Sadece yuvarlak parantez kullanılır.
$\sqrt[c]{a^b}$		
$-a^{\frac{1}{3}}$		
$(-a)^{\frac{1}{3}}$	YOK!	Üs ondalık bir sayı ise taban pozitif olmalı! (kompleks sayı da yok!)
$(-a)^3$	$(-a)^3$	Üs tam sayı ise taban negatif olabilir.
$\sqrt{a_{ij}}$	$a_{ij}^{0.5}$	İndisli değişken yoktur!

### İŞLEM SIRASI KURALI

Önce üs alma, sonra çarpma ve bölmeler ve daha sonra da toplama çıkarmalar yapılır. Çarpma ve bölme (veya toplama ve çıkarma) gibi eşdeğer işlemlerde hesap soldan sağa **doğru sıra** ile yapılır. Parantezli ifadeler (varsa), yukarıdaki kurala uy-gun olarak ve en içteki parantezli ifadeden başlanarak, **ilk önce** hesaplanır.

### 5.3.2.2. Standart Fonksiyonlar

VB Fonksiyonu	Matematik Anlamı	Açıklama
y= <code>Math.Sin(x)</code>	y= Sin x	x radyan
y= <code>Math.Cos(x)</code>	y= Cos x	x radyan
y= <code>Math.Tan(x)</code>	y= Tan x	x radyan $x \neq \pm \pi/2$
y= <code>Math.Atn(x)</code>	y= Arctan x	$-\pi/2 < y < \pi/2$
y= <code>Math.Log(x)</code>	y= Ln x	$\text{Log}_e x = \text{Ln } x$ $x > 0$
y= <code>Math.Sqrt(x)</code>	$\sqrt{x}$	$x \geq 0$
y= <code>Math.Exp(x)</code>	y= $e^x$	$x < 709.782712893$
y= <code>Math.Abs(x)</code>	y= $ x $	Mutlak değer
y= <code>Math.Sign(x)</code>	y= Sign x	x in işareti
y= <code>Math.Round(x)</code>		x in (yuvarlanmış) tam sayı kısmı
y= <code>rnd()</code>		$0 \leq y < 1$ arasında rasgele bir sayı

#### Örnekler:

<p>y= <code>Math.Sign(2000)</code> den y=1  y= <code>Math.Sign(-2000)</code> den y= -1  y= <code>Math.Sign(0)</code> den y=0</p>	<p>y= <code>rnd(1)</code> den y=0.7055475  y= <code>rnd(0)</code> den y= 0.0195312  y= <code>rnd(-1)</code> den y=0.224007  y= <code>rnd</code> den y=0.7055475</p>
<p><math>y = \text{Ln}  x^3 - x  + e^{\sqrt{5x}}</math>  İfadesinin VB karşılığı:</p> <p>y= <code>Math.Log(Abs(x^3-x)) + Math.Exp(Math.Sqrt(5*x))</code></p>	<p><math>y = x \sin \alpha^3 + \sqrt{x+3} \cos^2 \beta</math>  ifadesinin VB karşılığı, A= <math>\alpha</math> ve B= <math>\beta</math> alınarak</p> <p><b><math>\alpha, \beta</math> radyan biriminde ise:</b>  y=x* <code>Math.Sin(A^3)</code>+ <code>Math.Sqrt(x+3)</code>*  <code>Math.Cos(B)^2</code></p> <p><b><math>\alpha, \beta</math> derece biriminde ise:</b>  <math>\pi</math> sayısının karşılığı P=4* <code>Math.Atan(1)</code> olmak üzere</p> <p>y=x* <code>Math.Sin((A*P/180)^3)</code>+  <code>Math.Sqrt(x+3)</code>* <code>Math.Cos(B*P/180)^2</code></p>
<p>y=a+ <code>Math.Round</code> *b</p> <p>İfadesi a sayısı ile b sayısı ara-sında rasgele bir sayı üretir:  <math>a \leq y \leq b</math></p>	<p>y= -500+ <code>Math.Round</code> *1500 fadesi  <math>-500 \leq y \leq 1500</math> olur. Çok fazla veri gerektiren programların test aşamasında bu imkandan yararlanır.</p>

### 5.3.3. Kullanıcı Tanımlı Fonksiyonlar

Visual Basic dilinde istersek algoritmanın gerektirdiği düzene göre kendimiz fonksiyonlar yazabiliriz. Bu yazdığımız fonksiyonları program içinde çağırarak etkin bir programlama tekniği gösterebiliriz.

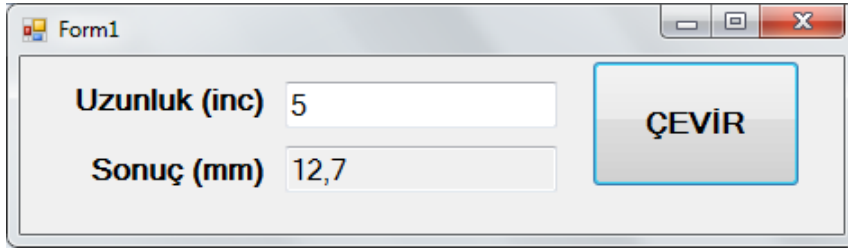
Fonksiyon oluşturmak tıpkı alt yordam oluşturmaya benzer. Fonksiyonların başlıksatırında ana programla değer alış verişini sağlayacak parametreleri ve değişken tanımlamaları yaparız. Aşağıda gösterilen fonksiyon formatını inceleyelim.

**Public Function** *fonksiyon\_ismi (değişken1 as tip , değişken2 as tip , .....)* *as tip*  
*kodlar*  
**End Function**

**Örnek 16:**

İnc uzunluk biriminin mm'ye dönüştürülmesi.  
1 inç=25,4 milimetre

#### Ekran Görüntüsü



#### Kodlar

```
Public Class Form1
    Dim uzunluk, a As Single

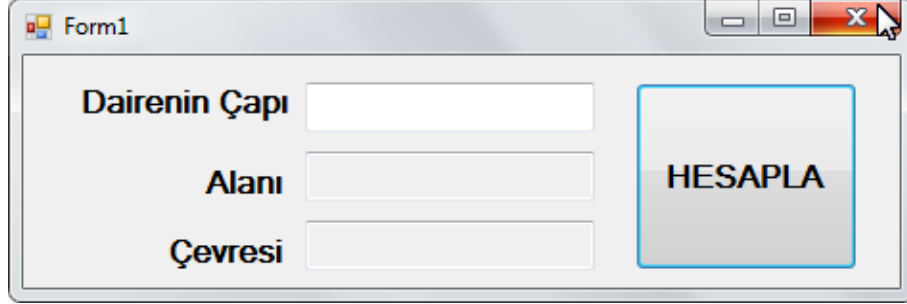
    Function cevir(ByVal uzunluk As Single) As Single
        cevir = uzunluk * 2.54
    End Function

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        a = TextBox1.Text
        TextBox2.Text = cevir(a) 'Fonkiyon çağrılıyor
    End Sub
End Class
```

## UYGULAMA FAALİYETİ

R çapı verilen dairenin alanını ve çevresini hesaplayan fonksiyon kullanarak yazınız.

**Alan:**  $A=\pi.R^2/4$       **Çevre:**  $S=\pi.R$



İşlem Basamakları	Öneriler
➤ Program için yukarıda ekran görüntüsü verilmiş form tasarımı yapınız.	➤ Form boyutlarınızı program çıktısını öngörerek ayarlayınız. Tasarladığınız formunuzu öğretmeninize göstererek onun fikrini alınız.
➤ Form üzerindeki nesne özellikleri ayarlayınız.	
➤ Aşağıda verilen kodları formunuza uyarlayınız.	➤ Kod yazım kurallarını hatırlayınız.
➤ Programınızı çalıştırarak deneyiniz.	

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

<b>Değerlendirme Ölçütleri</b>	<b>Evet</b>	<b>Hayır</b>
1. Formu tasarlayabildiniz mi?		
2. Formdaki nesne ayarlarını yapabildiniz mi?		
3. Kodları yazabildiniz mi?		
4. Programı denemek amacıyla çalıştırabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “Evet” ise “Ölçme ve Değerlendirme” ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Alt yordam yazmak programlarımıza aşağıdaki faydalardan hangisini sağlamaz?  
A) Hataları otomatik düzeltir.                      B) Anlaşılır yapar.  
C) Kısaltır.                                                      D) Hızlandırır.
2. Aşağıdaki program sonunda "a" değişkeninin içeriği seçeneklerin hangisinde verilmiştir?

```
Public Function hesap(ByVal x As Integer, ByVal y As Integer) As Integer
    Hesap = 2 * (x + y) - (x - y)
End Function
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    a = hesap
End Sub
```

- A) 20                      B) 18                      C) 16                      D) 10
3. Aşağıdakilerden hangisinde "Sayıcı" isimli fonksiyonun tanımlama satırı doğru verilmiştir?  
A) Public Function Sayıcı[x as integer] as integer  
B) Public sayıcı Function(x as integer) as integer  
C) Public Function sayıcı (x as integer) as integer  
D) Public Function (x as integer) as sayıcı
  4. Aşağıdakilerden hangisi standart Microsoft Visual Basic fonksiyonu değildir?  
A) Math.Now                      B) Rnd                      C) Form                      D) Math.Abs
  5. "Bir dizide kaç eleman olduğunu ..... fonksiyonu ile buluruz?"  
A) Number()                      B) Ubound()                      C) Mid()                      D) Len()
  6. Aşağıdaki fonksiyonlardan hangisi x sayısının mutlak değerini verir?  
A) Math.Atn(x)                      B) Math.Sqrt(x)                      C) Val(x)                      D) Math.Abs(x)



7. A  $x=3$   
 $n=10*(2*x-8)$   
 $a= \text{Math.sign}(n)$

Yukarıdaki ifade sonunda a değişkeninin içeriği aşağıdakilerden hangisi olur?  
A) a= -1      B) a=0      C) a=-20      D) a=1

8. `Date.Today` hangi fonksiyonun görevi nedir?  
A) Tarihi gösterir      B) Saati gösterir  
C) O anki tarih ve saati gösterir      D) Hiçbiri
9. “Belirli bir karakter ile birbirinden ayrılmış tek bir değişkenin içinde depolanmış verileri tek boyutlu bir dizi değişkene aktarır” tanımı hangi fonksiyonun tanımıdır?  
A) Str      B) Join      C) Lcase      D) Split
10. Aşağıda komut satırındaki `TextBox1` metin kutusunun değeri nedir?

```
Dim mesaj As String  
mesaj = "Milli Eğitim Bakanlığı"  
TextBox1.Text = LCase(mesaj)
```

- A) Mili Eğitim Bakanlığı      B) MİLLİ EĞİTİM BAKALIĞI  
C) milli eğitim bakanlığı      D) MilliEğitimBakanlığı

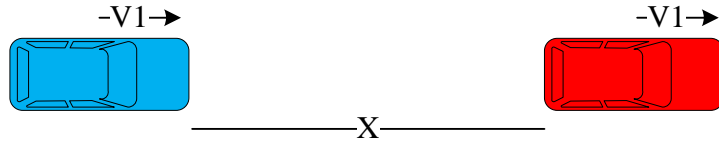
## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme”ye geçiniz.

# MODÜL DEĞERLENDİRME

Soru

Mavi araç kırmızı araçtan  $X = 15$  km geridedir. Mavi araç  $V_1 = 120$  km/h, kırmızı araç  $80$  km/h hızla aynı anda aynı yönde harekete başlamaktadır. Mavi araç kırmızı aracı **kaç dakika** sonra ( $t$ ) yakalayacağına ait programı yazınız.



$$t = \frac{X}{V_1 - V_2} \quad 1 \text{ saat} = 60 \text{ dakika}$$

## Ekran Görüntüsü

## Kodlar

```
Public Class Form1
    Dim v1, v2, x, yakalama_suresi, yakalama_suresi_dakika As Single
    Public Function hesapla(ByVal v1 As Single, ByVal v2 As Single, ByVal x
As Single) As Integer
        yakalama_suresi = x / (v1 - v2)
        yakalama_suresi_dakika = yakalama_suresi * 60
    End Function

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        v1 = TextBox1.Text
        v2 = TextBox2.Text
        x = TextBox3.Text
        hesapla(v1, v2, x)
        Label4.Text = "Yakalama Süresi: " + yakalama_suresi_dakika.ToString
+ " dakikadır."
    End Sub
End Class
```

## KONTROL LİSTESİ

Aşağıda verilen değerlendirme ölçütleriyle kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Programlarınızı yazmadan önce algoritma çıkardınız mı?		
2. Aynı uygulamayı farklı algoritmalar geliştirerek yapmayı denediniz mi?		
3. Tasarladığınız form programınızın kullanacak kişi açısından kullanışlı olduğunu düşünüyor musunuz?		
4. Motor sargı kontrolünü yapabiliyormusunuz?		
5. Yazdığınız kodlarda gereksiz satırlardan kaçındınız mı?		
6. Programlarınız doğru ve hatasız çalıştı mı?		

## DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “Evet” ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	C
2	D
3	C
4	A
5	D
6	C
7	D
8	A
9	C

## ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

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

## ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

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

### ÖĞRENME FAALİYETİ-4'ÜN CEVAP ANAHTARI

1	D
2	C
3	A
4	D
5	D
6	A
7	C
8	A

### ÖĞRENME FAALİYETİ-5'İN CEVAP ANAHTARI

1	A
2	B
3	C
4	C
5	D
6	D
7	A
8	A
9	D
10	C

### MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

#### Kodlar

```
Public Class Form1
    Dim v1, v2, x, yakalama_suresi, yakalama_suresi_dakika As Single
    Public Function hesapla(ByVal v1 As Single, ByVal v2 As Single, ByVal x
As Single) As Integer
        yakalama_suresi = x / (v1 - v2)
        yakalama_suresi_dakika = yakalama_suresi * 60
    End Function

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        v1 = TextBox1.Text
        v2 = TextBox2.Text
        x = TextBox3.Text
        hesapla(v1, v2, x)
        Label14.Text = "Yakalama Süresi: " + yakalama_suresi_dakika.ToString
+ " dakikadır."
    End Sub
End Class
```

# KAYNAKÇA