

Visual Basic'te birçok işlem görsel olarak yapılmaktadır. VB programcılığı menü hazırlamak ve programın kullanıcıya yansıyan arayüzünün hazırlanması için kod yazmak zorunda bırakmaz.

VB başlatıldığında "Project1" adında bir proje hazırlanmakta ve bu projeye "Form1" adında bir form otomatik olarak dahil edilmektedir. Visual basic'in temel bileşenidir. Windows pencerelerinin Visual Basic'te karşılığıdır. Herhangi bir uygulama çalıştırıldığında formlar pencereye dönüşür. Her Visual Basic uygulaması en az bir form içermek zorundadır. Bu formun içeriği başlangıçta boştur. Şimdi bu formda herhangi bir değişiklik yapmadan programı koşturalım.

Form üzerinde sağ tıkla.

Menu: View Code
Menu Editor
Lock Kontrols
Properties

Code penceresine geç. (F7)

Bu şekilde açılan kod penceresinin "Form1" adlı forma ait olduğunu kod penceresinin başlığında anlamak mümkündür. Kod penceresindeki bilgiler kendiliğinden geldi. İlk satırda **Private Sub Form_Load ()** bilgisi var. VB programları dahilinde hazırlanan alt program veya yordamlar **Sub** bildiri deyimi ile başlar ve **End Sub** bildiri deyimi ile biter.

Private bildiri deyimi ile alt programın geçerlilik alanı belirlenmektedir.

Üzerinde çalışılan projede şimdilik tek bileşen olan "Form1" un Load olduğunda işlenecek olan kod.

Proje çalıştırıldığında Form1 üzerine "İlk programımıza hoş geldiniz..." yazdıralım.

```
Private Sub Form_Load()  
    Show  
    Print "İlk programımıza hoşgeldiniz..."  
End Sub
```

"İlk programımıza hoşgeldiniz..." sabit bilgisini formun üzerindeki konumunu değiştirelim.

```
Private Sub Form_Load()  
    Show  
    CurrentX = 1440  
    CurrentY = 1440  
    Print "İlk programımıza hoşgeldiniz..."  
End Sub
```

1 Inch =1440 Twip
1 Inch = 2.54 cm

```

Private Sub Form_Load()
    Show
    CurrentX = 1440
    CurrentY = 1440
    FontName = "Verdana"
    FontSize = 15
    FontBold = True
    Print "İlk programımıza hoşgeldiniz..."
End Sub

```

Kaydedelim:
File – Save Project As

Dosya Uzantısı

İşlevi

Proje Dosyası (.VBP) : Bir projedeki tüm dosyaların yollarını, yerlerini ve dosya adlarını taşır. Genel olarak bir proje dosyası belirtilen proje dizininde saklanır.

Form Modülü Dosyası (.FRM) : Form'un, Form'un Kontrol nesnelerinin tanımını ve kontrol nesneleriyle bağlantılı kodları taşır.

Standart Modül Dosyası (.BAS): Bir projedeki tüm modüller tarafından paylaşılabilen program kodunu taşır. Standart modüller pek çok farklı proje arasında paylaşılabilir.

INPUTBOX() İLE BİLGİ GİRİŞİ (GİRİŞ KUTUSU)

VB programının icrası esnasında, klavyeden bilgi girişi yapılmasını sağlar.

InputBox (mesaj [,başlık] [varsayım] [,xpoz,ypoz] [,yardım dosyası,konu no]])

Mesaj 256 karakter uzunluğundadır.

Yardım Dosyası , Konu No

Bu iki özelliklerle bir yardım dosyası izni ve bir konu numarası belirlenebilir. Böylece kullanıcı mesaj kutusunda yardım istediğinde ilgili yardım dosyası açılır ve ilgili konu gösterilir.

```

Private Sub Form_Load()
    Dim ad
    Show
    CurrentX = 1440
    CurrentY = 1440
    FontName = "Verdana"
    FontSize = 15
    FontBold = True
    ad = InputBox("Adınızı Giriniz:", "InputBox'ın başlığı", "Serhat", 1440, 1440)
    Print ad
End Sub

```

MSGBOX() İLE MESAJ YAZMAK (MESAJ KUTUSU)

MsgBox deyimi özellikle kullanıcıya belli bir mesaj verme ve kullanıcıdan belli bir yanıt almak için kullanılır.

MsgBox deyiminin iki türlü kullanımı vardır: MsgBox Deyimi, MsgBox Fonksiyonu.

MsgBox Deyimi:

MsgBox deyimi belirtilen mesajı bir mesaj kutusu içinde ekranda görüntüler. OK tuşuna basılmanın ardından programın işleyişine devam edilir.

MsgBox mesaj [, simgeler, [başlık]]

```
Private Sub Form_Load()  
MsgBox "Merhaba Dünyalı", 16, "Kullanıcı Mesajı"  
End Sub
```

Simgeler

Anlamı	Değeri	Simgesi
vbCritical	16	X
vbQuestion	32	?
vbExclamation	48	!
vbInformation	64	i

MsgBox Fonksiyonu :

MsgBox Fonksiyonu mesajın gösteriminin yanı sıra mesaj kutusu üzerinde basılan bir düğmenin değerinin de geri dönmesini sağlar.

Cevap=MsgBox (mesaj [,tip] , [başlık] , [yardım dosyası,konu no]))

Tip = düğme+ icon + varsayılan+ öncelik

Düğme

Sabit Bilgisi	Değeri
vbOKOnly	0
vbOKCancel	1
vbAbortRetryIgnore	2
vbYesNoCancel	3
vbYesNo	4
vbRetryCancel	5

İcon

Sabit Bilgisi	Değeri	
vbCritical	16	X
vbQuestion	32	?
vbExclamation	48	!
vbInformation	64	i

Varsayılan

Sabit Bilgisi	Değeri
vbDefaultButton1	0
vbDefaultButton2	256

vbDefaultButton3 512

Öncelik

Sabit Bilgisi	Değeri
vbApplicationModal	0
vbSystemModal	4096

3+16+256 = 275

```
Private Sub Form_Load()  
    Dim secim  
    secim = MsgBox("Değişiklikleri Kaydetmek İstiyor musunuz?", 275, "Uyarı")  
End Sub
```

Düğme Seçenekleri:

Sabit Bilgisi	Değer
vbOK	1
vbCancel	2
vbAbort	3
vbRetry	4
vbIgnore	5
vbYes	6
vbNo	7

```
Private Sub Form_Load()  
    Dim secim, mesaj, tip, baslik  
    mesaj = "Devam Etmek İstiyor musunuz?"  
    tip = vbYesNo + vbCritical + vbDefaultButton2  
    baslik = "Mesaj Kutusu Başlığı"  
    secim = MsgBox(mesaj, tip, baslik)  
    If secim = 6 Then  
        MsgBox "Evet Butonuna Tıklandı"  
    ElseIf secim = 7 Then  
        MsgBox "Hayır Butonuna Tıklandı"  
    End If  
End Sub
```

```
Private Sub Form_Load()  
    Dim secim, mesaj, tip, baslik  
    mesaj = "Devam Etmek İstiyor musunuz?"  
    tip = vbYesNo + vbCritical + vbDefaultButton2  
    baslik = "Mesaj Kutusu Başlığı"  
    secim = MsgBox(mesaj, tip, baslik)  
    If secim = vbYes Then  
        MsgBox "Evet Butonuna Tıklandı"  
    ElseIf secim = vbNo Then  
        MsgBox "Hayır Butonuna Tıklandı"  
    End If  
End Sub
```

KONTROL NESNELERİ

Kullanıcı ara yüzünü hazırlamak için formlar üzerine veri girişi için metin kutusu (text box) , bir işlemi başlatmak için komut düğmesi (command button) gibi birimler yerleştirilir. Bu veri-işlem birimlerine kontrol nesneleri denir. Bu kontrol nesneleri araç kutusundan seçilerek kullanılır.

Araç kutusuna ek kontroller eklemek: sağ tık- add tab – components – Microsoft ADO Controls 6.0 (OLEDB)

Her kontrolün kendisine ait özellikleri vardır. Bu özellik tasarım anında veya çalışma zamanında kod içinde ayarlanır.

Kontrol.Özellik = Değer

STANDARD KONTROL NESNELERİ VE İŞLEVLERİ

Pointer :

Picture Box (Resim Kutusu) : Resimlerin gösterilmesini sağlar.

```
Private Sub Form_Load()  
    Picture2.Picture = LoadPicture("c:\windows\Dalgalar.bmp")  
End Sub
```

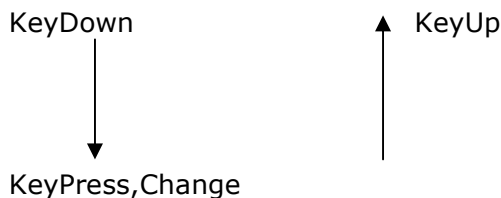
Label (Etiket): Form üzerinde bilginin gösterilmesini sağlar.

```
Private Sub Form_Load()  
    Label2.Caption = "Lütfen Adınızı ve Soyadınızı Giriniz:"  
End Sub
```

```
Private Sub Form_Load()  
    Label2.Caption = "Lütfen Adınızı ve Soyadınızı Giriniz:"  
    Label2.AutoSize = True  
End Sub
```

TextBox (Metin Kutusu): Bilgi girişi için kullanılır.

Kod penceresinde iki liste kutusu bulunmaktadır. İlk liste kutusunun adı Object, 2. liste kutusunun adı ise Procedure dır. Object de nesneler, Procedure de ise o nesnenin gerçekleştirebileceği olaylar bulunmaktadır.



```
Private Sub Text2_Change()  
    MsgBox "change olayı"  
End Sub
```

```
Private Sub Text2_KeyDown(KeyCode As Integer, Shift As Integer)  
    MsgBox "key down olayı"  
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)  
    MsgBox "key press olayı"  
End Sub
```

```
Private Sub Text2_KeyUp(KeyCode As Integer, Shift As Integer)  
    MsgBox "key up olayı"  
End Sub
```

```
Private Sub Text2_Change()  
    Text1.Text = Text2.Text  
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 13 Then MsgBox Text2.Text  
End Sub
```

Frame (Çerçeve)

Kontrolleri gruplamak için kullanılır. Çerçeve içinde bulunan kontrollerin toplu olarak özellikleri düzenlenebilir. Örneğin Enabled ve Visible özellikleri toplu olarak düzenlenebilir. Çerçevenin hareketiyle içindeki tüm kontrol de taşınır.

Command Button (Komut Düğmesi)

Bir işlemin başlatılması için kullanılır.
Caption="&Çıkış" ----ALT Ç

```
Private Sub Command1_Click()  
    End  
End Sub
```

Butona resim eklemek için style=1 olmalı.

CheckBox (İşaret Kutusu)

Belli seçeneklerin seçili olup olmadıklarını belirtmeye yarar.

0-işaretsiz

1-işaretili

```
Private Sub Command1_Click()
```

```
    Dim deger
```

```
    If Check1.Value = 1 Then deger = deger + vbCrLf + Check1.Caption
```

```
    If Check2.Value = 1 Then deger = deger + vbCrLf + Check2.Caption
```

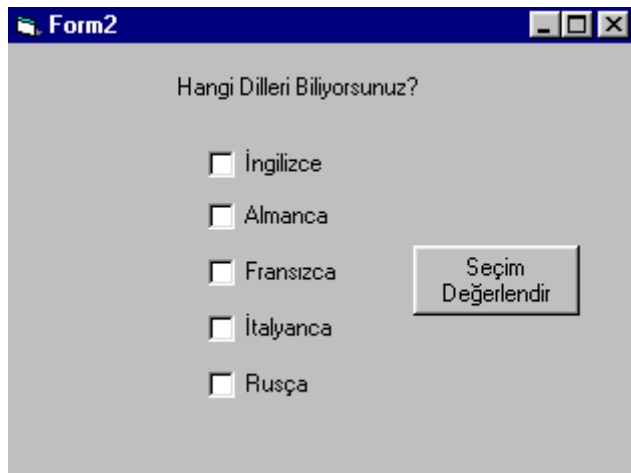
```
    If Check3.Value = 1 Then deger = deger + vbCrLf + Check3.Caption
```

```
    If Check4.Value = 1 Then deger = deger + vbCrLf + Check4.Caption
```

```
    If Check5.Value = 1 Then deger = deger + vbCrLf + Check5.Caption
```

```
    MsgBox deger, , "Bildiğim Diller"
```

```
End Sub
```



Option Button (Seçenek Düğmesi)

Birçok seçenekten sadece bir tanesinin seçildiği bir seçenek kontrolüdür.

```
Private Sub Command1_Click()
```

```
    If Option1.Value = True Then MsgBox "İlkokul Mezunuyum"
```

```
    If Option2.Value = True Then MsgBox "Ortaokul Mezunuyum"
```

```
    If Option3.Value = True Then MsgBox "Lise Mezunuyum"
```

```
    If Option4.Value = True Then MsgBox "Üniversite Mezunuyum"
```

```
End Sub
```

```
Private Sub Form_Load() 'Varsayılan Değeri belirleyelim
```

```
    Option1.Value = True
```

```
End Sub
```

ComboBox (Bileşik Kutu)

Birçok seçenekten birinin seçilmesini sağlayan bir kutudur. Özellikle belli seçeneklerden birisinin açılan bir listeden seçilmesini sağlar.

```
Private Sub Combo1_Click()  
    If Combo1.Text = "İstanbul" Then MsgBox "İstanbul"  
    If Combo1.ListIndex = 1 Then MsgBox Combo1.Text  
    If Combo1.Text = "İzmir" Then MsgBox "İzmir"  
    If Combo1.ListIndex = 3 Then MsgBox Combo1.Text  
End Sub
```

```
Private Sub Form_Load()  
    Combo1.Text = ""  
    Combo1.AddItem "İstanbul" 'ListIndex'i 0  
    Combo1.AddItem "Ankara" 'ListIndex'i 1  
    Combo1.AddItem "İzmir" 'ListIndex'i 2  
    Combo1.AddItem "Muğla" 'ListIndex'i 3  
End Sub
```

RemoveItem Özelliği : Listedeki belirtilen elemanı çıkartır.

Örn: Combo1.RemoveItem 0

Style Özelliği : 0,1,2 değerleriyle ComboBox'ın tipini ve çalışma biçimini düzenler. Varsayılan değeri 0 dır.

Style Özelliği	İşlevi
0	DropDownCombo
1	SimpleList
2	DropDownList

0 değeri aşağı doğru açılan ve kullanıcı tarafından bilgi girilebilen bir ComboBox yaratır.

1 değeri aşağı doğru açılmayan ancak içeriği değiştirilebilen bir ComboBox yaratır.

2 değeri aşağı doğru açılan ve kullanıcı tarafından bilgi girilemeyen bir ComboBox yaratır.

Text Özelliği: TextBox ile kullanıma benzer şekilde ComboBox'un o anda içerdiği metni döndürür.

Örn: Combo1.Text

ListIndex Özelliği: Listedeki aktif elemanı belirtir. ListIndex=0 değeri listenin 1. elemanının seçilmesini sağlar.

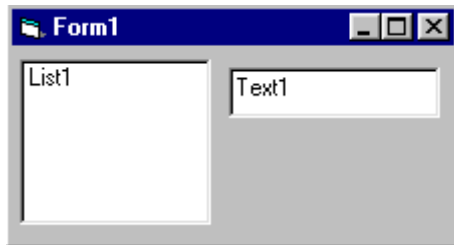
ListCount : Listedeki eleman sayısını verir.

ListBox (Liste Kutusu)

Birçok elemanı bir liste olarak göstermeye yarar.

```
Private Sub Form_Load()  
    Text1.Text = ""  
    List1.AddItem "İstanbul" 'ListIndex'i 0  
    List1.AddItem "Ankara" 'ListIndex'i 1  
    List1.AddItem "İzmir" 'ListIndex'i 2  
    List1.AddItem "Muğla" 'ListIndex'i 3  
End Sub
```

```
Private Sub List1_Click()  
    Text1.Text = List1.Text  
End Sub
```



AddItem:Listeye bir eleman eklemek için kullanılır.

RemoveItem: Listedeki belirtilen elemanı çıkartır.

Örn : List1.RemoveItem 0 ` Listenin 1. elemanını listeden çıkartır.

Text Özelliği:TextBox ile kullanıma benzer şekilde ListBox'un o anda içerdiği metni döndürür.

ListIndex Özelliği: Listedeki aktif elemanı belirtir. ListIndex=0 değeri listenin 1. elemanının seçilmesini sağlar.

Örn: List1.RemoveItem List1.ListIndex ` seçilen elemanı listeden siler.

ListCount Özelliği: Listedeki eleman sayısını verir.

Örn: MsgBox List1.ListCount

Horizontal / Vertical Scroll Bar

Nesnelerin ya da uygulama ilgili verilerin form üzerinde ileri/geri ya da sağa/sola kaydırılmasını sağlayan kontroldür.

```
Private Sub Form_Load()  
    VScroll1.Min = 0  
    VScroll1.Max = 256  
    VScroll2.Min = 0  
    VScroll2.Max = 256  
    VScroll3.Min = 0  
    VScroll3.Max = 256
```

```

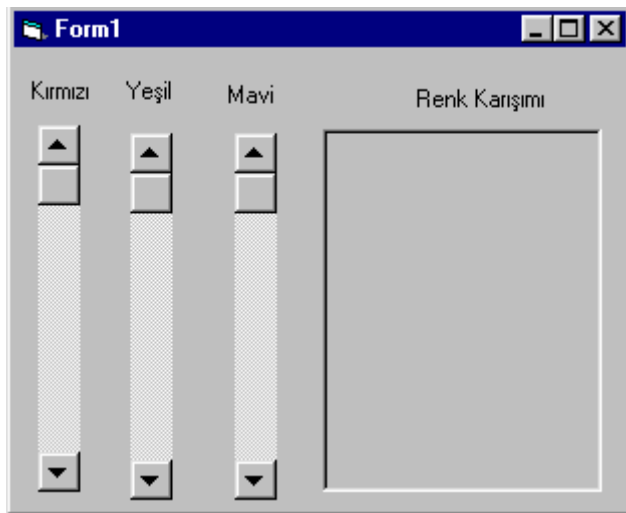
End Sub
Private Sub VScroll1_Change()
    Picture1.BackColor = RGB(VScroll1.Value, VScroll2.Value, VScroll3.Value)
End Sub

Private Sub VScroll2_Change()
    Picture1.BackColor = RGB(VScroll1.Value, VScroll2.Value, VScroll3.Value)
End Sub

Private Sub VScroll3_Change()
    Picture1.BackColor = RGB(VScroll1.Value, VScroll2.Value, VScroll3.Value)
End Sub

```

Min ve Max değerleri ile kaydırma çubuğunun hareketlerinin başını ve sonunu temsil eder. Kaydırma çubuğunun oklarına tıklamak smallchange olayı, kaydırma çubuğu üzerindeki alanlara tıklamak largechange olayıdır. Properties kutusundan ayarlanır.



Timer (Zaman Kontrolü)

Program içinde zamanı kullanmanızı sağlar. Tasarım zamanında forma eklenen Timer kontrolü çalışma zamanında görülmez.

Enabled Özelliği: False ve True değerleri ile zamanın çalışmasını ve durmasını sağlar.

Interval Özelliği: Saatin çalışma periyodunu belirtir. 1-65535 mili saniye arasında değişen değer. Eğer zamanın saniye olarak görülmesi istenirse Interval değeri 1000 yapılır.

```

Private Sub Command1_Click()
    Label1.Caption = "0"
    Timer1.Enabled = True
    Command1.Enabled = False
    Command2.Enabled = True
End Sub

```

```

Private Sub Command2_Click()

```

```

    Timer1.Enabled = False
    Command1.Enabled = True
    Command2.Enabled = False
End Sub

Private Sub Timer1_Timer()
    Label1.Caption = Str(Val(Label1.Caption) + 0.1)
End Sub

```



DriveListBox, DirectoryListBox, FileListBox

DriveList kontrolü sistemde bulunan sürücüler, DirectoryList kontrolü sürücülerdeki dizinleri, FileList kontrolü de dizinlerdeki dosyaları listeler. Bu kontroller genellikle birlikte kullanılırlar. Özellikle sabit disk üzerindeki dosyaları görmek, aramak, açmak vb. işlemleri yapmak için kullanılır.

DriveListBox Kontrolü

Sistemdeki sürücüler gösterir. Drive özelliği ile aktif sürücüye ulaşılır. Sürücünün seçilmesi Change olayını oluşturur.

```

Sub Drive1_Change( )
    Dir1.Path = Drive1.Drive
End Sub

```

DirectoryListBox Kontrolü

Sistemde bulunan sürücülerdeki dizinleri gösterir. Sürücüdeki dizinleri göstermek için Path özelliği kullanılır.

```

Sub Dir1_Change( )
    File1.Path = Dir1.Path
End Sub

```

FileListBox Kontrolü

Dizindeki dosyaları gösterir. FileName özelliği ile dosyaların adlarına ulaşılır. Pattern özelliği ile de listelenecek dosya uzantıları belirtilir. Örneğin *.EXE gibi.

```

Private Sub Combo1_Click()

```

```

File1.Pattern = Combo1.Text
End Sub

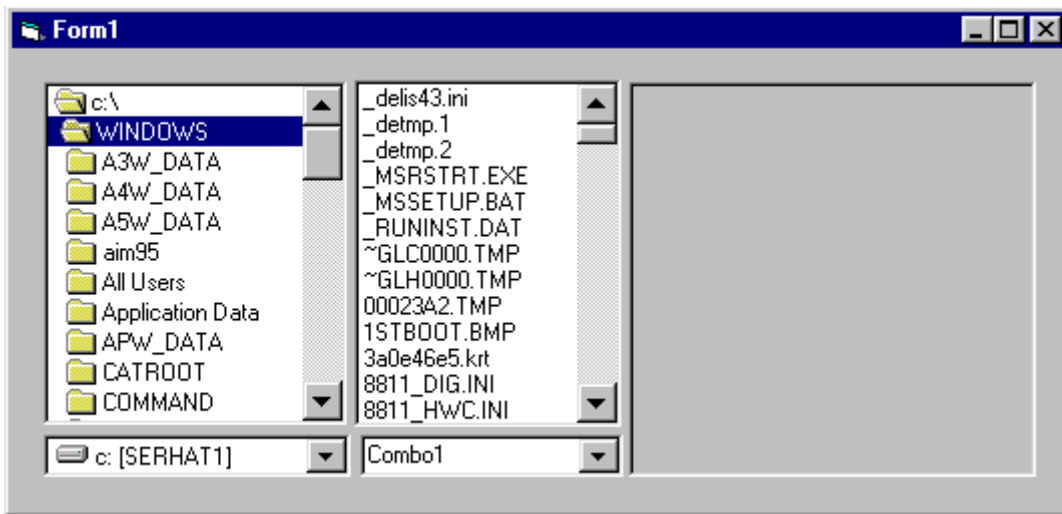
Private Sub Dir1_Change()
File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change()
Dir1.Path = Drive1.Drive
End Sub

Private Sub File1_Click()
Dim uzanti
uzanti = Right(File1.FileName, 3)
If uzanti = "BMP" Or uzanti = "bmp" Then
Picture1.Picture = LoadPicture(File1.FileName)
End If
End Sub

Private Sub Form_Load()
Combo1.AddItem "*. *"
Combo1.AddItem "*.exe;*.com;*.bat"
Combo1.AddItem "*.txt"
Combo1.AddItem "*.bmp"
Combo1.ListIndex = 0
End Sub

```



Shape (Şekil)

Grafik şekillerin oluşturulmasını sağlar.

Shape Özelliği: 0 ve 5 arasında değerler alarak değişik şekillerin çizilmesini sağlar.

- | | |
|---|------------|
| 0 | Dikdörtgen |
| 1 | Kare |
| 2 | Elips |
| 3 | Çember |

4	Oval Dörtgen
5	Oval Kare

FillStyle: Şeklin içini boyamak için kullanılacak deseni belirtir.

0	Tam Dolu
1	Nesnenin zemin rengi
2	Yatay çizgi
3	Dikey çizgi
4	Sola eğik
5	Sağa eğik
6	Kareli
7	Çapraz

Line (Çizgi)

Form üzerine çizgi çizmek için kullanılır.

X1,X2,Y1,Y2 Özellikleri

X1 ve Y1 çizginin başlangıç koordinatlarını, X2 ve Y2 çizginin bitiş koordinatlarını belirtir.

Line Metodu : Form üzerine çizgi çizer.

nesne.Line (x1,y1)-(x2,y2),renk

Örn:

```
Private Sub Form_Load()
    Show
    Line (20, 20)-(1000, 1000), vbRed
End Sub
```

Image (Resim)

Resim göstermeyi sağlar. PictureBox'a göre daha az yetenekli olan Image kontrolü özellikle daha az sistem kaynağına gereksinim duyduğu için kullanılabilir.

Picture özelliği Image içindeki resmi belirtir. LoadPicture fonksiyonundan yararlanılır.

```
Private Sub Form_Load()
    Image1.Picture = LoadPicture("C:\WINDOWS\Bulutlar.bmp")
End Sub
```

Data (Veri)

Bir veritabanında bulunan verilere erişmeyi sağlar.

VB 6.0 ile birlikte ADO veri erişim yöntemi kullanılmaktadır. ADO ile birlikte ADO Data Control geliştirilmiştir ve Data Controlün yerini almıştır.

OLE (Nesne Birleştirme) (Object Linking and Embedding)

OLE teknolojisini destekleyen iki program arasında bağlantı kurmayı sağlar. Bağlantının sonucunda veri transferi yapmak ya da verileri kullanmak mümkün olmaktadır.

Data – OLE Örneği

Nesne	Özellik	Ayar
Data1	Connect DatabaseName RecordSource	Access C:\DERS_2\Yetenek.mdb Artists
Text1	DataSource DataField	Data1 Name
Text2	DataSource DataField	Data1 Address
Text3	DataSource DataField	Data1 City
Text4	DataSource DataField	Data1 State
Text5	DataSource DataField	Data1 WorkPhone
Command1	Caption	Çık

```
Private Sub Command1_Click()  
    End  
End Sub
```

OLE örneği için bir OLE nesnesi ekleyelim. Nesne Ekle diyalog kutusunda uygulamamıza bağlayabileceğimiz nesnelerin listesi görünmektedir.

Dosyadan Oluştur` u seçelim. *Bağla`* yı seçelim. *Gözet kutusundan* C:\DERS_2\Satışlar_01.xls dosyasını bulalım. Tamam a tıklayalım.

Şimdide sağ alt köşeye yeni bir OLE nesnesi çizelim. Nesne Ekle kutusunu İptal ile geçelim. C:\DERS_2 dizinindeki excel dosyasını açıp, grafiği kopyalayalım. VB de forma yeni eklediğimiz OLE nesnesine sağ tıklayıp *Paste Special* diyelim. *Özel Yapıştır* kutusunda *Bağlantı Yapıştır`ı* seçelim.

```
Private Sub OLE1_Updated(Code As Integer)  
    OLE2.Update  
End Sub
```

Form1

Name:

Address:

City:

State:

Work Phone:

CD Satış Kayıtları		
Altı Aylık CD Satışı (2007)		
	İstanbul	Ankara
Ocak	1250	900
Şubat	1800	1400
Mart	2000	1600
Nisan	1650	1200
Mayıs	1500	1250
Haziran	1450	1150

DEĞİŞKEN TANIMLAMAK VE KULLANMAK

Bir programda geçici olarak verilerin temsili ve değerlerin saklanması için değişkenler kullanılır. Değişkenlerin değerleri, program akışı içerisinde sürekli olarak değişebilir. Örneğin:

```
x=36
y=24
x=x+3
y=y+x
```

şeklindeki atama ifadeleri ile, x değişkeninin değeri önce 36 ve y değişkeninin değeri 24 olarak atanmıştır. Sonra x'in değeri 39 ve y'nin değeri 63 olmuştur. Atama ifadesinde, = sembolünün sağ tarafındaki ifade hesaplanır ve bulunan sonuç sol taraftaki değişkenin bellekteki yerine atanır. Bu anlamda, bir değişkene yeni bir değer atanırsa, bu değişkenin önceki değeri silinecektir.

Değişkenlerin Tanımlanması:

Bir değişken kullanılmadan önce tanımlanır. Bu tanımlama genellikle Dim deyimi ile yordamın başında yapılır:

Yapısı : Dim Değişken [As tip]

Bir değişkenin özellikleri:

- Bir harf ile başlar.
- En çok 255 karakterden oluşur.
- (Nokta, +, -, / , boşluk vb) semboller içermez.
- Kapsama alanı içinde tek olmalıdır.
- Bir yordam adı ile aynı olmamalıdır.
- Örn: Geçerli değişken isimleri : MAAS

Yas1
NetMaas

Genel_Toplam
Geçersiz Değişken İsimleri: 1X
A-1
Brut Maas

Değişken tanımlanmaz veya tanımlanırken tip belirtilmezse, VB o değişkeni Variant veri tipinde algılar. Variant bir değişken herhangi bir tip veriyi içerebilir. Variant tipli değişken kullanmanın iki kötü yanı olabilir:

- Daha fazla bellek kullanımına neden olur.
- Variant veri tipi bazı veri işlem fonksiyonları için geçersiz olabilir.

Değişkenler Nerede Tanımlanır?

Bilindiği gibi VB projelerine .frm uzantılı form dosyaları ve .bas uzantılı Module dosyaları eklenebiliyor. Buna göre değişkenler ya Formlarda ya da BAS uzantılı modüllerde tanımlanabilir.

Vb deki hazır nesnelerin içerdiği kod penceresindeki her yordamda değişken tanımlı yapılabilir. Ancak bu değişkenler sadece o yordam dahilinde kullanılabilir.

Object liste kutusunda bulunan (General) seçeneği seçildiğinde , Procedure liste kutusunda da (Declarations) aktif olur. Forma ait bütün yordamlarda kullanılmak istenen değişkenler General – Declarations konumunda tanımlanır. Ve bu değişkenler söz konusu form aktif olduğu süre içerisinde yaşarlar. Başka bir forma geçildiğinde bellekten silinir ve içeriği boşalır.

Özetlemek gerekirse Formlar söz konusu olduğunda değişkenler yordamların içinde ya da Forma ait (declaration) penceresinde tanımlanır. Yordam dahilinde tanımlanan değişkenler ancak tanımlandığı yordam içinde kullanılabilir. Söz konusu formun (declaration) penceresinde tanımlanan değişkenler Forma ait bütün yordamlarda kullanılabilir.

BAS uzantılı modüllerde de değişkenler tanımlanıp kullanılabilir. Projeye modül eklemek için Project – Add Module menüsünden faydalanılır.

Kod girişi yapılan bu pencere, formlara ait olan kod penceresinden farklı değildir. Fakat bu pencerede kod yoktur. Ama yazılabilir. Aynı formlarda olduğu gibi Sub ve End Sub arasında yordam olarak yazılmalıdır.

Formlarda olduğu gibi modüle de yordam eklemek için Tools menüsünden Add Procedure komutu kullanılır. Modül içinde tüm yordamlarda kullanılmak istenen değişken modülün (General) (Declarations) kısmında tanımlanır.

Global Değişkenler:

Projenin her yerinde yani Form ve modülerde kullanılmak istenen değişkenler BAS uzantılı bir modülün declaration penceresinde Global veya Public bildiri deyimi ile tanımlanır.

Private Sözcüğü:

Private sözcüğü ile, form düzeyinde değişkenler belirleyebilirsiniz. Formun General Declarations kısmında Dim yerine Private sözcüğünü kullanarak da değişkenleri form düzeyinde geçerli kılabilirsiniz. Gerçekte, Dim deyimi ile Private deyimi birbiri ile değiştirilebilir şekilde kullanılabilmektedir.

Statik ve Dinamik Değişkenler:

Default olarak tüm lokal değişkenler dinamiktir, yani olay prosedürünün her çağırılışında, bir önceki çağırılış sonunda hesaplanmış değerlerini saklamazlar. Bunun yerine prosedür içinde ilk atanan değerleri ile yeniden işleme başlarlar.


```
Private Sub Command1_Click()
    Dim x As Double
    x = x + 10
    MsgBox x
End Sub
```

Lokal değişkenleri, prosedürün her çağırılışında, bir önceki çağırılış sonunda hesaplanmış olan değerlerini saklar hale getirmek için, Static sözcüğü ile tanımlanmaları gereklidir.

Static x As Integer

```
Private Sub Command1_Click()
    Static x As Double
    x = x + 10
    MsgBox x
End Sub
```

Açık Tanımlamaya Zorlama:

Değişkenlerin tanımlanmasını mecburi tutmak için Declarations kısmında *Option Explicit* deyimi yazılmalıdır.

Tools menüsünden Options komutu seçilip, Editör sekmesinde Require Variable

Declaration seçeneği işaretlenirse *Option Explicit* deyimi otomatik olarak gelir.

Artık değişken tanımlanmadan kullanılmak istenirde Variable Not Defined hatası oluşur.

Veri Tipleri :

Bildiri Deyimi	Veri Tipi	Bellek Alanı	Değer Aralığı
String	Karaktersel	1 karakter 1 bayt	65.535
Integer	Tamsayı	2 bayt	-32.768 to 32.767
Long	uzun tamsayı	4 bayt	2.147.483.647
Single	tek duyarlı sayısal	4 bayt	3.4×10^{38}
Double	çift duyarlı sayısal	8 bayt	$1,7 \times 10^{308}$
Currency	para-çok büyük		
	Çok küçük Sayı	8 bayt	922.337.203.685.447
Byte	tamsayı	1 bayt	0 to 255
Boolean		2 bayt	True / False
Date	tarih	8 bayt	1/1/100 to 12/31/9999
Variant	16 bayt + 1 bayt her karakter için		

String Türü Değişkenler: Arasına + veya & sembolü konarak birbirlerine eklenebilirler.

```
Private Sub Command1_Click()
    Dim x As String, y As String, z As String, c As String
    x = "Marmara"
    y = "Çağdaş"
    z = "Bilimler Vakfı"
    c = x + y & z: MsgBox c
End Sub
```

Sabit Uzunluklu Stringler:

```
Dim ad As String * 8
```

Şeklinde ad değişkeni String türünde ve 8 karakter uzunluğunda tanımlanmıştır. Bu durumda ad="Abdülkerim" ataması yapılırsa, ad değişkenine "Abdülker" parçası atanacaktır. Ad="Ali" ataması yapılırsa, ad değişkeninde "Ali " şeklinde 5 boşluklu bir string bulunacaktır.

Veri Tiplerinin Dönüştürülmesi

VB çok sayıda dönüştürme fonksiyonuna sahiptir. Bu fonksiyonlar değerlerin tiplerinin birbirlerine çevrilmesini sağlar.

Function	Return Type
Cbool	Boolean
Cbyte	Byte
Ccur	Currency
Cdate	Date
CDbl	Double
Cdec	Decimal
Cint	Integer
CLng	Long
CSng	Single
CStr	String
Cvar	Variant

Not: Bir veri tipindeki değer bir başka tipe çevrileceği zaman karşı tarafın tipine uygun olmalıdır. Örneğin Long bir değer Integer bir değişkene atanacaksa Long değer Integer sınırlarında olmalıdır.

VarType Fonksiyonu:

Variant değişken içindeki verinin tipini öğrenmek için VarType fonksiyonu kullanılır. Bir değişkenin tipini belirten bir tamsayı döndürür.

VarType (Değişken)

Return Values

Constant	Value	Description
vbEmpty	0	Empty (boş, değer verilmemiş)
vbNull	1	Null (no valid data) Geçerli veri yok
vbInteger	2	Integer
vbLong	3	Long integer

vbSingle	4	Single-precision floating-point number
vbDouble	5	Double-precision floating-point number
vbCurrency	6	Currency value
vbDate	7	Date value
vbString	8	String
vbObject	9	Object
vbError	10	Error value
vbBoolean	11	Boolean value
vbVariant	12	Variant (used only with arrays of variants)
vbDataObject	13	A data access object
vbDecimal	14	Decimal value
vbByte	17	Byte value
vbUserDefinedType	36	User-defined tip içeren veri tipi
vbArray	8192	Array

Örn:

```
Private Sub Command1_Click()
    Dim IntVar, StrVar, DateVar, kontrol1, kontrol2, kontrol3
    ' Initialize variables.
    IntVar = 459: StrVar = "Hello World": DateVar = #2/12/1969#
    kontrol1 = VarType(IntVar) ' Returns 2.
    kontrol2 = VarType(DateVar) ' Returns 7.
    kontrol3 = VarType(StrVar) ' Returns 8.
    MsgBox kontrol1
    MsgBox kontrol2
    MsgBox kontrol3
End Sub
```

ISEMPTY() Fonksiyonu: Variant tipli bir değişkenin bilgi içerip içermediğini öğrenmek için bu fonksiyondan yararlanılır. Bu fonksiyona parametre olarak verilen değişkene daha önceden bilgi aktarılmış yani belleğin değişkene ayrılan yerinde bilgi varsa, geriye mantıksal yanlış False , değişkene daha önce bilgi aktarılmışsa geriye mantıksal doğru yani True değeri döndürülür.

IsEmpty(Variant Değişken Adı)

```
Private Sub Form_Load()
    Dim ad, soyad
    ad = "Serhat"
    Show
    FontSize = 16
    CurrentX = 400
    CurrentY = 400
    Print IsEmpty(ad), IsEmpty(soyad)
```

End Sub

ISNUMERIC() Fonksiyonu: Variant tipli bir değişken sayısal bilgi içeriyorsa geriye True, içermiyorsa False değeri döndürür.

IsNumeric(Variant Değişken Adı)

Private Sub Form_Load()

Dim x

x = InputBox("Sayı giriniz")

If IsNumeric(x) Then

MsgBox ("Değişkene Sayısal Bilgi Aktardınız")

Else

MsgBox ("Değişkene Aktardığınız Bilgi Sayısal Değil")

End If

End Sub

Variant Tipinin Dezavantajı:

Private Sub Command1_Click()

Dim x, y

x = InputBox("1. sayı") '5 girilirse

y = InputBox("2. sayı") '4 girilirse

Print x + y

End Sub

Çıktı: 54 olur.

Private Sub Command1_Click()

Dim x As Integer, y As Integer

x = InputBox("1. sayı")

y = InputBox("2. sayı")

Print x + y

End Sub

Çıktı: 9 olur

Veya

Private Sub Command1_Click()

Dim x, y

x = InputBox("1. sayı")

y = InputBox("2. sayı")

Print Val(x) + Val(y)

End Sub

Çıktı: 9 olur, string veri tipli bilgi nümerik olarak kullanılacaksa VAL() fonksiyonu kullanılır.

KULLANICI TARAFINDAN BELİRLENEN BERİ TİPLERİ (USER DEFINED TYPES)

Çeşitli tiplerdeki değişkenleri biraraya getirerek tek bir isim altında toplanarak oluşturulan veri tipine Kayıt (Record) denir. Bir kayıt tanımlamak için bir Modülün Declarations kısmına

```
Type personelkayit
    sicil As Integer
    ad As String * 20
    soyad As String * 30
    adres As String
End Type
```

```
Public personel As personelkayit
```

Veya formun Declarations kısmına

```
Private personel As personelkayit
```

yazılır.

Tanımlanan kayıt tipi veri yapısını oluşturan bileşenlere erişmek içinse :

```
Private Sub Command1_Click()
    Print personel.sicil
    Print personel.ad
    Print personel.soyad
    Print personel.adres
End Sub
```

```
Private Sub Form_Load()
    personel.sicil = 123
    personel.ad = "serhat"
    personel.soyad = "özekes"
    personel.adres = "Marmara Çağdaş Bilimler Vakfı - Hasanpaşa"
End Sub
```

SABİTLER (CONSTANTS)

Değişkenlerin aksine sabitler, sahip oldukları değerler ile kullandığımız bilgi tanımlamalarıdır. Özellikle bilinen değerlerin kullanımını sağlar. Kodlamayı ve kodun okunabilirliğini kolaylaştırır.

[Private|Private] Const adı [As tipi] = ifade yada değer

Örnek:

```
Const pi=3.14
Const katsayi=2.5
```

```
Private Sub Command1_Click()
    Const katsayi = 2.5
    MsgBox katsayi * 3
End Sub
```

OPERATÖRLER

+ (Toplama) Operatörü

sayısal bilgi + sayısal bilgi
karakter sel bilgi + karakter sel bilgi

sayi=10+25
sonuc="Visual"+" "+ "Basic" veya sonuc="Visual"&" "&"Basic"

- (Çıkarma) Operatörü

sayısal bilgi – sayısal bilgi

sayi=20-10

* (Çarpma) Operatörü

sayısal bilgi * sayısal bilgi

sayi=10*20

/ (Bölme) ve \ (Tam Bölme) Operatörü

sayısal bilgi / sayısal bilgi

Print 25 / 10 'çıktı 2.5
Print 25 \ 10 'çıktı 2

^ (üs alma) Operatörü

sayısal bilgi ^ sayısal bilgi

sonuc=3^2 ` 9 olur

AND (ve) Operatörü

sonuc = ifade1 **And** ifade2

If ifade1	And ifade2	Then sonuc
True	True	True
True	False	False
True	Null	Null
False	True	False
False	False	False
False	Null	False

Null	True	Null
Null	False	False
Null	Null	Null

OR (veya) Operatörü

sonuc = ifade1 Or ifade2

If ifade1	Or ifade2	Then sonuc
True	True	True
True	False	True
True	Null	True
False	True	True
False	False	False
False	Null	Null
Null	True	True
Null	False	Null
Null	Null	Null

NOT (Değil) Operatörü

sonuc = Not ifade

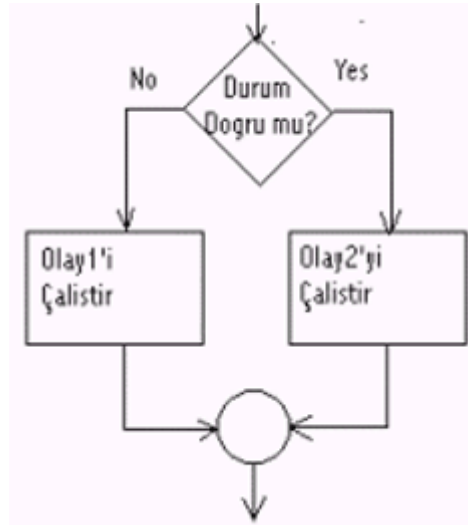
If ifade	Then sonuc
True	False
False	True
Null	Null

Karşılaştırma Operatörleri

= (Eşittir ve Atama) Operatörü
 <> (Eşit Değil) Operatörü
 < (Küçüktür) Operatörü
 > (Büyüktür) Operatörü
 <= (Küçük Eşit) Operatörü
 >= (Büyük Eşit) Operatörü

KARAR YAPILARI

If...Then...Else Deyimi



Yapı 1 :

If koşul Then işlem

Yapı 2 :

If koşul Then

[işlem1]

Else

[işlem2]

End If

Yapı 3 :

If koşul1 Then

[işlem1]

[ElseIf koşul2 Then

[işlem2]

....

[Else

[işlem n]]

End If

Örn: Bir öğrencinin vize ve final notları veriliyor. Bu notlara göre geçme notunu $Vize \cdot 0.40 + Final \cdot 0.60$ formülüne göre hesaplayan, final veya geçme notunun 50'den küçük olması durumunda bütünleme notunu isteyen ve buna göre bir geçme notu hesaplayıp sonuçta geçme notu ile 'GEÇTİ' veya 'KALDI' mesajı veren program:

Option Explicit

Dim vize As Integer, final As Integer

Dim ortalama As Integer, butunleme As Integer

Private Sub Command1_Click()

Cls

Form1.FontSize = 15

vize = InputBox("Vize Notunuzu Giriniz", "Öğrencinin Vize Notu")

final = InputBox("Final Notunuzu Giriniz", "Öğrencinin Final Notu")

*ortalama = vize * 0.4 + final * 0.6*


```

Show
If ortalama >= 50 And final >= 50 Then
    Print "Bu öğrenci " & ortalama & " ortalaması ile geçmiştir"
Else
    butunleme = InputBox("Öğrencinin bütünleme notunu giriniz", "Öğrencinin bütünleme notu")
    ortalama = vize * 0.4 + butunleme * 0.6
    If ortalama >= 50 And butunleme >= 50 Then
        Print "Bu öğrenci " & ortalama & " ortalaması ile bütünlemede geçmiştir"
    Else
        Print "Bu öğrenci bütünlemeye girmesine rağmen " & ortalama & " ortalaması ile kalmıştır"
    End If
End If
End Sub

```

Ödev: Bir kişiye ait cinsiyet, boy ve kilo bilgilerini alarak aşağıdaki şartlara göre, kişinin ideal kiloda olup olmadığını, kaç kilo alması veya kaç kilo vermesi gerektiğini bulan programı yazınız?

İdeal kilo; bayanlarda:Boy-110
 erkeklerde:Boy-108

IIF() Fonksiyonu:

Yapısı: IIF(Koşul,birinci ifade,ikinci ifade)

Bir bakıma , IF deyiminin kısıtlı bir şeklidir. Yalnızca iki olasılığın söz konusu olduğu durumlarda IF deyimi yerine IIF fonksiyonu kullanılabilir.

```

Private Sub Command2_Click()
    sonuc = Iif(ortalama >= 50, "Geçer Not", "Zayıf Not")
    MsgBox sonuc
End Sub

```

Select Case Deyimi

Yapısı:

```

Select Case testifadesi
    [Case ifade listesi-1
        [ifade bloğu-1] ]
    [Case ifade listesi-2
        [ifade bloğu-2] ]
    [Case Else
        [ifade bloğu-n] ]
End Select

```

Bir önceki örnekte şu ifade de kullanılabilir:

```

Private Sub Command2_Click()
    Select Case ortalama
        Case Is < 50
            MsgBox "Zayıf Not"

```

```
Case Else
    MsgBox "Geçer Not"
End Select
End Sub
```

Örn: 34 ise İSTANBUL, 35 ise İZMİR, değilse FARKLI BİR İL:

```
Private Sub Command1_Click()
    il = InputBox("Hangi ilde oturuyorsunuz")
    Select Case il
        Case 34
            MsgBox "İSTANBUL"
        Case 35
            MsgBox "İZMİR"
        Case Else
            MsgBox "Farklı Bir İl"
    End Select
End Sub
```

Örn: Klavyeden girilen harfi test edelim:

```
Private Sub Command2_Click()
    Dim deger
    deger = InputBox("Bir harf yada rakam giriniz")
    Select Case deger
        Case "A" To "Z"
            MsgBox "A ile Z arasında bir büyük harf girdiniz"
        Case "a" To "z"
            MsgBox "a ile z arasında bir küçük harf girdiniz"
        Case 0 To 9
            MsgBox "0 ile 9 arasında bir rakam girdiniz"
        Case Else
            MsgBox "İstenmeyen bir tuşa bastınız"
    End Select
End Sub
```

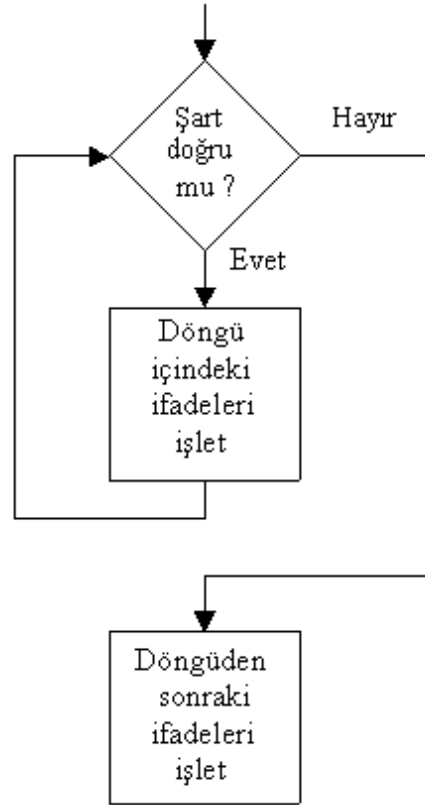
DÖNGÜ YAPILARI

Do ... Loop Deyimi

Yapı-1:

Bir blok deyimi verilen koşul doğru (True) olduğu sürece işletir.

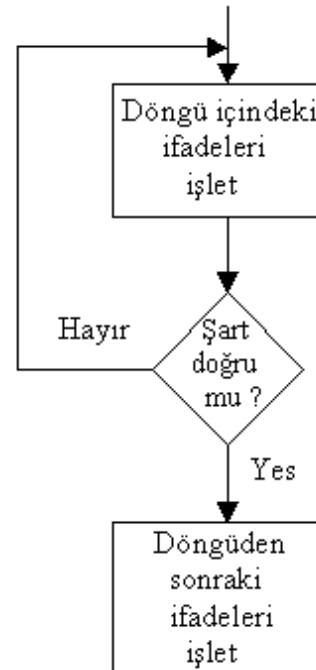
Do [{While | Until} koşul]
 [ifade bloğu]
 [Exit Do]
 [ifade bloğu]
 Loop



Yapı-2:

Bir blok deyimi verilen koşul sağlanıncaya kadar işletilir.

Do
 [ifade bloğu]
 [Exit Do]
 [ifade bloğu]
 Loop [{While | Until} koşul]



Yapı1-Örn:

```

Private Sub Command2_Click()
    Dim num
    Cls
    num = 1
    Do While num <= 10
        Print num; ' ; yan yana basılmasını sağlar
        num = num + 1
    Loop
End Sub

```

veya

```

Private Sub Command2_Click()
    Dim num
    Cls
    num = 1
    Do Until num > 10
        Print num; ' ; yan yana basılmasını sağlar
        num = num + 1
    Loop
End Sub

```

Yapı-2 Örn:

```

Private Sub Command2_Click()
    Dim num
    Cls
    num = 1
    Do
        Print num; ' ; yan yana basılmasını sağlar
        num = num + 1
    Loop While num <> 11 ' veya <
End Sub

```

veya

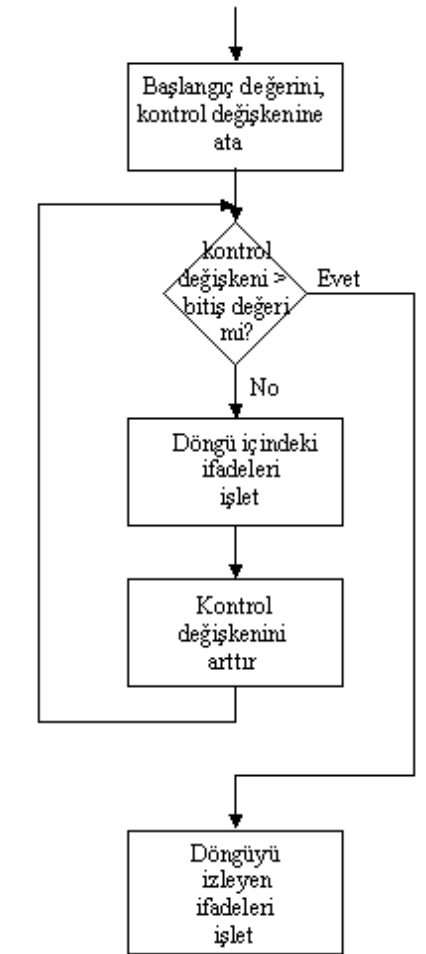
```

Private Sub Command2_Click()
    Dim num
    Cls
    num = 1
    Do
        Print num; ' ; yan yana basılmasını sağlar
        num = num + 1
    Loop Until num = 11 ' veya num > 10
End Sub

```

For...Next Deyimi**Yapısı:**

For sayaç=başlangıç to bitiş [Step arttırım]
 [ifade bloğu]
 [Exit For]
 [ifade bloğu]
 Next [sayaç]



Örn: 1-10 arasındaki sayıları bas.

```

Private Sub Command2_Click()
    Dim i
    Cls
    For i = 1 To 10
        Print i;
    Next i
End Sub

```

```

Private Sub Command2_Click()
    Dim i
    Cls
    For i = 0 To 10 Step 2
        Print i;
    Next i
End Sub

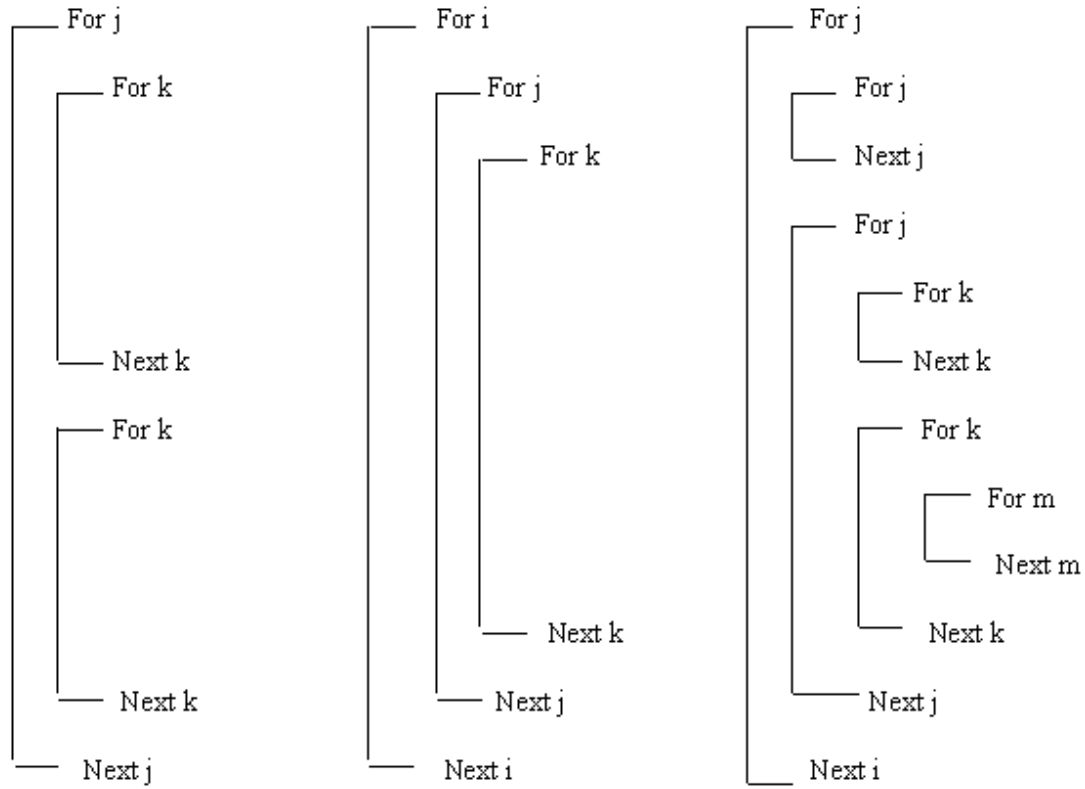
```

```

Private Sub Command2_Click()
    Dim i
    Cls
    For i = 10 To 0 Step -2
        Print i;
    Next i
End Sub

```

İç içe Döngüler



Örn: Çarpım Tablosu

```

Private Sub Command1_Click()
    Dim j As Integer, k As Integer
    Picture1.Cls
    For j = 1 To 10
        For k = 1 To 10
            Picture1.Print k; "x"; j; "="; j * k,
        Next k
        Picture1.Print
    Next j
End Sub
  
```

Form2									
Çarpım Tablosunu Göster									
1x1=1	2x1=2	3x1=3	4x1=4	5x1=5	6x1=6	7x1=7	8x1=8	9x1=9	10x1=10
1x2=2	2x2=4	3x2=6	4x2=8	5x2=10	6x2=12	7x2=14	8x2=16	9x2=18	10x2=20
1x3=3	2x3=6	3x3=9	4x3=12	5x3=15	6x3=18	7x3=21	8x3=24	9x3=27	10x3=30
1x4=4	2x4=8	3x4=12	4x4=16	5x4=20	6x4=24	7x4=28	8x4=32	9x4=36	10x4=40
1x5=5	2x5=10	3x5=15	4x5=20	5x5=25	6x5=30	7x5=35	8x5=40	9x5=45	10x5=50
1x6=6	2x6=12	3x6=18	4x6=24	5x6=30	6x6=36	7x6=42	8x6=48	9x6=54	10x6=60
1x7=7	2x7=14	3x7=21	4x7=28	5x7=35	6x7=42	7x7=49	8x7=56	9x7=63	10x7=70
1x8=8	2x8=16	3x8=24	4x8=32	5x8=40	6x8=48	7x8=56	8x8=64	9x8=72	10x8=80
1x9=9	2x9=18	3x9=27	4x9=36	5x9=45	6x9=54	7x9=63	8x9=72	9x9=81	10x9=90
1x10=10	2x10=20	3x10=30	4x10=40	5x10=50	6x10=60	7x10=70	8x10=80	9x10=90	10x10=100

Örn: Kaçan buton. Command1(height=600)

```
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Dim i
    If Command1.Top = 600 Then
        For i = 600 To form1.Height - 2 * Command1.Height Step 3
            Command1.Top = i
        Next i

    Else
        For i = form1.Height - 2 * Command1.Height To 600 Step -3
            Command1.Top = i
        Next i

    End If
End Sub
```

EXIT Deyimi

Do...Loop, For...Next döngüleri , bir Function procedure ya da Sub procedure'dan çıkışı sağlar.

Yapısı: Exit { Do | For | Function | Sub }

END Deyimi

Yapısı : End [Function | If | Select | Type}]

DİZİLER (ARRAYS)

Aynı tipte ve birbirleriyle ilgili bilgilerin oluşturduğu bütüne dizi denir. Program içerisinde aynı anda aynı tür bilgiden çok sayıda mevcut olması ve bu bilgiler üzerinde toplu işlem yapılmasının gerekmesi durumunda diziler kullanılır. Bir dizi aynı tipte ve aynı adı paylaşan bir grup değişken demektir. Diziler birçok değişkene aynı adla ulaşmayı sağlayan bir grup veri yapısıdır. Bir indeks numarası ile dizi içindeki elemanlara ulaşılır. Dizi içindeki elemanlar aynı tipteki verilerdir; örneğin haftanın günleri ya da iller gibi.

Dizilerin gerekliliğini iki örnek ile inceleyelim:

Örnek 1: 100 adet isim ve telefon bilgisini saklamak için,

İsim : 100 adet

Telefon : 100 adet

Toplam 200 adet değişken kullanılmalıdır.

Örnek 2: bir sınıfta okuyan 40 öğrencinin isim ve her öğrencinin 5'er farklı dersi ve bu derslerden aldıkları 3 farklı not bilgisini aynı anda bellekte tutmak için,

İsim : 40 adet

Dersler : 40 öğrenci x 5 ders = 200 adet

Notlar : 40 öğrenci x 5 ders x 3 not = 600 adet

Toplam 840 adet değişken kullanılmalıdır. Bu durum imkansızdır.

Şimdi bu örnekleri dizilerle inceleyelim.

Örnek 1 :

İsim(x) ---100 elemanlı
Telefon(x) --- 100 elemanlı

Örnek 2 :

40 elemanlı İsim(I) ---isim dizisinin (I+1). elemanı
40,5 elemanlı Dersler(I,J) --- (I+1). Öğrencinin, (J+1). dersi
40,5,3 el. Notlar(I,J,K)--- (I+1). Öğrencinin, (J+1). Dersinin, (K+1). notu

Dizi Tanımlama : Dim dizideğişkeni (eleman sayısı) As tipi

Dizilerin eleman sayıları sıfırdan başladığı için 5 elemanlı bir dizinin eleman sayısı 4 olarak belirtilir.

Örn: Haftanın günleri

```
Private Sub Command1_Click()  
    Dim gunler(6) As String, i As Integer  
    gunler(0) = "Pazartesi" 'dizinin 1. elemanı  
    gunler(1) = "Salı"  
    gunler(2) = "Çarşamba"  
    gunler(3) = "Perşembe"  
    gunler(4) = "Cuma"  
    gunler(5) = "Cumartesi"  
    gunler(6) = "Pazar" 'dizinin 7. elemanı  
    For i = 0 To 6  
        Print "Haftanın " & i + 1 & " . günü " & gunler(i)  
    Next i  
End Sub
```

Visual Basic’de iki tür dizi vardır:

- Sabit eleman sayılı (Statik) dizi
- Dinamik dizi

Sabit Eleman Sayılı Dizi:

Eleman sayıları sabittir. Daha sonra eleman sayıları değiştirilemez.

Dim gunler(6) As String

Dim aylar(11) As String

Dinamik Diziler

Bazen bir diziyi yaratırken eleman sayısını tam olarak kestiremeyiz. Bir dizinin eleman sayısının sonradan değiştirebilme özelliğine dinamik diziler denir. Dinamik dizinin boyutları istenildiği zaman değiştirilebilir. Şu şekilde kullanılır:

```
Dim dinamikdizi( )  
şeklinde tanımlandıktan sonra, program akışı içinde  
ReDim DinamikDizi(10)  
yada daha sonra  
ReDim DinamikDizi(10+x)  
olarak tanımlanır.
```

ReDim deyiminin her kullanılışında dizideki bütün değerler silinir. Bu değerleri korumak için Preserve deyimi kullanılır.

ReDim Preserve DinamikDizi (10)

Örn:

```
Private Sub Command1_Click()  
    Dim gunler(), i As Integer  
    ReDim gunler(6)  
    gunler(0) = "Pazartesi"  
    gunler(1) = "Salı"  
    gunler(2) = "Çarşamba"  
    gunler(3) = "Perşembe"  
    gunler(4) = "Cuma"  
    gunler(5) = "Cumartesi"  
    gunler(6) = "Pazar"  
    For i = 0 To 6  
        Print "Haftanın " & i + 1 & " . günü " & gunler(i)  
    Next i  
    ReDim Preserve gunler(8)  
    gunler(7) = "8. eleman"  
    Print gunler(7)  
    Print gunler(4)  
End Sub
```

Çok Boyutlu Diziler

Diziler, çok boyutlu olarak kullanılabilir.
Dim aylar_satışlar (11,4) As String

Dizilerin Alt ve Üst Sınırlarıyla Tanımlanması

Dim aylar_satışlar (5 to 12) as String

Dim Matris(9,9) As Double olarak tanımlanan bir dizi ,
Dim Matris(1 To 10, 1 To 10) As Double olarak da tanımlanabilir.

Dizilerde Option Base Deyimi Kullanımı

General Declarations kısmında tanımlanır.

Yapısı : Option Base{0 | 1}

Bir dizinin indeksinin alt sınırı varsayım olarak 0 dır. Eğer bu değer değiştirilecekse 1 değeri kullanılır.

Option Explicit

Option Base 1

```
Private Sub Command1_Click()  
    Dim alt, ust  
    Dim dizi(10, -5 To 15, 20 To 30)  
    ust = UBound(dizi, 1): Print ust,  
    ust = UBound(dizi, 2): Print ust,  
    ust = UBound(dizi, 3): Print ust  
    alt = LBound(dizi, 1): Print alt,  
    alt = LBound(dizi, 2): Print alt,  
    alt = LBound(dizi, 3): Print alt  
End Sub
```

KARAKTERSEL VE SAYISAL FONKSİYONLAR

LEN() FONKSİYONU

Kendisine parametre olarak verilen karaktersel veya başka tip bilginin uzunluğunu bulur. Karakter sel sabitler LEN() fonksiyonuna doğrudan veya bir değişken aracılığıyla verilebilir.

Genel yazılış: LEN(Bilgi veya Bilgi İçeren Değişken)

Örn: *ad="serhat"*
Uzunluk=Len(ad) 'uzunluk 6 olur
Uzunluk=Len("serhat") 'uzunluk 6 olur

Bunun dışında Len() fonksiyonu ile başka tipteki bilgiler kullanılırsa , o bilginin bellekte kapladığı değer sonuç olarak döner.

LEFT() FONKSİYONU

Bu fonksiyon dışarıdan parametre olarak karakter sel bir bilgiyi, karakter sel bilgi içeren değişkeni veya Variant tipli değişkeni alabilmektedir. Kendisine verilen bilginin soldan istenilen kadarını ayırmak , çıkarmak amacıyla kullanılmaktadır.

Genel yazılış: LEFT(ifade, uzunluk)

Örn: *sehir="adana"*
Sonuc=Left(sehir,3) 'Sonuc "ada" olur
Sonuc=Left("Microsoft Visual Basic",9) 'Sonuc "Microsoft" olur

RIGHT() FONKSİYONU

Çalışma mantığı olarak Left() fonksiyonuna benzer. Kendisine verilen bilginin sağdan istenilen kadarını ayırmak , çıkarmak amacıyla kullanılmaktadır.

Genel yazılış: RIGHT(İfade,uzunluk)

Örn: *sehir="adana"*
Sonuc=right(sehir,3) 'Sonuc "ana" olur
Sonuc=right("Visual Basic",5) 'Sonuc "Basic" olur

LTrim() Fonksiyonu

Kendisine parametre olarak verilen bilginin baş tarafında bulunan boşlukları çıkarmak amacıyla kullanılır.

Genel yazılış: LTRIM(Karakter sel Bilgi)

Örn: *sehir=" Adana"*
Sonuc=LTRIM(sehir) 'sonuc "Adana" olur

Rtrim() Fonksiyonu

Kendisine parametre olarak verilen bilginin sağ tarafında bulunan boşlukları çıkarmak amacıyla kullanılır.

Genel yazılış: RTRIM(Karakter sel Bilgi)

Örn: *sehir="Adana "*
Sonuc=RTRIM(sehir) 'sonuc "Adana" olur

```
Private Sub Command1_Click()
    Dim ad As String * 20, soyad As String * 30
    ad = "Serhat"
    soyad = "Özekes"
    MsgBox ad & soyad
End Sub
```

Çıktı: Serhat Özekes

olurdu. Oysa

```
Private Sub Command1_Click()
    Dim ad As String * 20, soyad As String * 30
    ad = "serhat"
    soyad = "Özekes"
    MsgBox RTrim(ad) & " " & RTrim(soyad)
End Sub
```

Çıktı: Serhat Özekes

olur.

Trim() Fonksiyonu

Kendisine parametre olarak verilen bilginin hem başındaki, hem de sonundaki boşlukları çıkarmak amacıyla kullanılır.

Genel yazılış: TRIM(Karaktersel Bilgi)

Mid() Fonksiyonu

Kendisine parametre olarak verilen bilginin belirlenen bir kısmını ayırıp çıkarmak amacıyla kullanılır.

Genel yazılış: MID(Karaktersel Bilgi, Başlama Yeri, Alınacak Karakter Sayısı)

Örn: *a=MID("İstanbul",2,4)* *'stan olur.*

Instr() Fonksiyonu

Kendisine parametre olarak verilen karaktersel bilgide başka bir karaktersel bilgiyi aramak amacıyla kullanılır.

Genel yazılış: INSTR(Başlama Yeri, Karaktersel Bilgi, Aranacak Bilgi)

Örn: *sonuc=Instr(1,"adana","a")* *'sonuc 1 olur*
 sonuc=Instr(1,"adana","n") *'sonuc 4 olur*
 sonuc=Instr(2,"adana","a") *'sonuc 3 olur*
 sonuc=Instr(4,"adana","a") *'sonuc 5 olur*

LCase() Fonksiyonu

Kendisine parametre olarak verilen bilginin içinde bulunan alfabetik karakterlerin büyük olanlarını küçük harfe çevirmektedir.

Genel yazılış: LCase(Karaktersel Bilgi)

Visual Basic, Türkçe'ye özgü olan "İ" ve "ı" harflerini LCase() fonksiyonu ile küçük harfe çeviremez. Bunun için:

```

Private Sub Command1_Click()
    Dim i As Byte, s As String, ss As String, soyad As String, h As String
    soyad = "IAaaFGGII-İİggFFDSGİ"
    For i = 1 To Len(soyad)
        h = Mid(soyad, i, 1)
        If h = "I" Then h = "i"
        If h = "İ" Then h = "i"
        s = s + h
    Next i
    MsgBox s
    ss = LCase(s)
    MsgBox ss
End Sub

```

UCase() Fonksiyonu

Kendisine parametre olarak verilen bilginin içinde bulunan alfabetik karakterlerin küçük olanlarını büyük harfe çevirmektedir.

Genel yazılış: UCASE(Karaktersel Bilgi)

Visual Basic, Türkçe'ye özgü olan "ı" ve "i" harflerini LCASE() fonksiyonu ile küçük harfe çeviremez.

Str() Fonksiyonu

Str() fonksiyonu ile sayısal bir bilgi karaktersel bilgi tipine çevrilir.

Genel yazılış: STR(Sayısal Bilgi)

Örn:

```

Dim no As Long, ad As String
no = 9616538
ad = "Serhat"
MsgBox Str(no) + ad

```

String() Fonksiyonu

Bir karakterin istenilen sayıda ekrana arka arkaya tekrarlanarak yazılmasını sağlar.

Genel yazılış: STRING(Sayı,Karakter)

Örn: Print String(20, "-")

Chr() Fonksiyonu

0 ile 255 arasındaki herhangi bir sayıya karşılık gelen ASCII karakterini elde etmek amacıyla kullanılır.

Genel yazılış: CHR(Sayı)

Sonuc=CHR(65) 'sonuc "A" olur

ASCII tablosu:

```
Private Sub Form_Load()  
    Dim i As Integer  
    Cls  
    Show  
    For i = 0 To 49  
  
        Print i & " = " & Chr(i) & " ",  
        Print i + 50 & " = " & Chr(i + 50) & " ",  
        Print i + 100 & " = " & Chr(i + 100) & " ",  
        Print i + 150 & " = " & Chr(i + 150) & " ",  
        Print i + 200 & " = " & Chr(i + 200) & " "  
    Next i  
    Print 250 & " = " & Chr(250) & " ",  
    Print 251 & " = " & Chr(251) & " ",  
    Print 252 & " = " & Chr(252) & " ",  
    Print 253 & " = " & Chr(253) & " ",  
    Print 254 & " = " & Chr(254) & " ",  
    Print 255 & " = " & Chr(255) & " "  
  
End Sub
```

Tab() Fonksiyonu

Genel yazılış: Tab(Ekranın Bilgi Yazılacak Sütunu)

```
For i = 1 To 10  
    Print Tab(10 + i); "+"  
Next
```

Spc() Fonksiyonu

Boşluk oluşturmak amacıyla kullanılır.

Genel yazılış: SPC(Boşluk Sayısı)

Örn: *Print "Serhat"; Spc(5); "Özekes"*

Asc() Fonksiyonu

Herhangi bir karakterin ilk karakterinin ASCII kodunu bulmak amacıyla kullanılmaktadır.

Örn: *sonuc=ASC("A")* *'sonuc 65 olur.*
 Sonuc=ASC("Adana") *'sonuc 65 olur.*

```
Private Sub Form_Load()  
    Dim i, mesaj  
    For i = Asc("A") To Asc("Z")  
        mesaj = mesaj + Chr(i)  
    Next i  
    MsgBox mesaj  
End Sub
```

Val() Fonksiyonu

Bu fonksiyon karakterel bir bilginin sayısal bilgiye dönüştürülmesi amacıyla kullanılmaktadır.

Genel yazılış: VAL(Karakterel Bilgi)

Sonuc=VAL("123.456") 'sonuc 123.456 olur
Sonuc=VAL("123adana") 'sonuc 123 olur
Sonuc=VAL("A15") 'sonuc 0 olur
Sonuc=VAL("11A21") 'sonuc 11 olur

Abs() Fonksiyonu

Bir sayının mutlak değerini bulmak için kullanılır.

Genel yazılış: ABS(Sayı)

Sonuc=ABS(50) 'sonuc 50
Sonuc=ABS(-50) 'sonuc 50

Int() Fonksiyonu

Ondalık nokta içeren bir sayıyı tam sayıya çevirmek amacıyla kullanılır.

Sonuc= Int(10.25) 'sonuc 10 olur
Sonuc=Int(10.75) 'sonuc 10 olur
Sonuc= Int(-10.75) 'sonuc -11 olur
Sonuc= Int(-10.25) 'sonuc -11 olur

Rnd Fonksiyonu

0 ile 1 arasında rastgele sayı üretir. ($0 \leq \text{sayı} < 1$)

Belirli aralıklarda rastgele bir sayı üretmek için şu formül uygulanır.

$\text{Int}((\text{üst sınır}-\text{altsınır}+1)*\text{Rnd}+1)$

Örneğin 1-49 arasında rastgele sayı üretmek için:

*sayi = Int((49 - 1 + 1) * Rnd + 1)*
MsgBox sayi

Sqr() Fonksiyonu

Pozitif bir sayısal değerin karekökünü bulmak için kullanılır.

Genel yazılış: SQR(Karekökü Alınacak Sayı)

Sign() Fonksiyonu

Sayısal bir sabit veya sayısal bilgi içeren değişkenin pozitif veya negatif olup olmadığını öğrenmek için kullanılır.

Genel yazılış: SIGN(Sayısal Bilgi)

Sonuc=SIGN(10) 'sonuc 1 olur
Sonuc=SIGN(-10) 'sonuc -1 olur
Sonuc=SIGN(0) 'sonuc 0 olur

Date(), Time() ve Now() Fonksiyonları

Date() fonksiyonu ile sistem tarihi öğrenilirken, Time() fonksiyonu ile sistem saati elde edilmektedir. Now() fonksiyonu ile sistem tarihini aynı anda edebiliriz.

Text1.Text = Date
Text2.Text = Time

Text3.Text = Now

Shell() Fonksiyonu

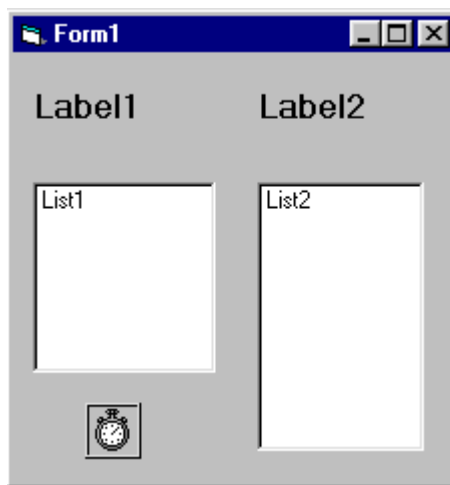
Bilgisayara kurulu olan herhangi bir programı VB projesi dahilinde çalıştırabilmemize imkan sağlar.

Örneğin hesap makinesi ve WordPad programlarını açalım.

```
Private Sub Command1_Click()  
    Shell ("c:\windows\calc.exe")  
End Sub
```

```
Private Sub Command2_Click()  
    Shell ("c:\windows\Write.exe")  
End Sub
```

ÖDEV: Saat örneği.



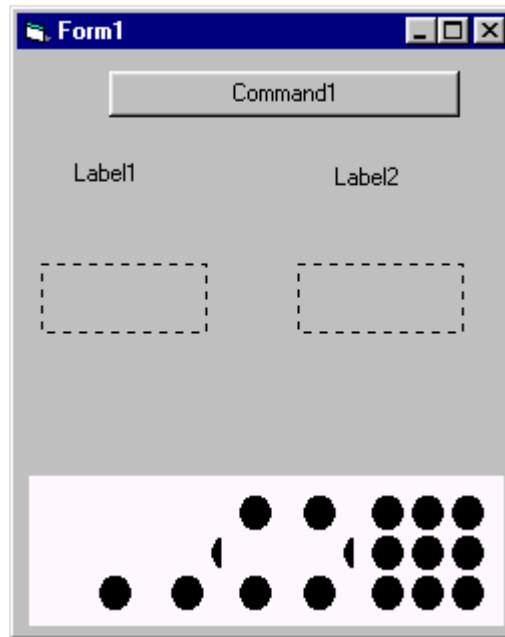
```
Private Sub Form_Load()  
    List1.AddItem " "  
    List1.AddItem "On"  
    List1.AddItem "Yirmi"  
    List1.AddItem "Otuz"  
    List1.AddItem "Kırk"  
    List1.AddItem "Elli"  
  
    List2.AddItem ""  
    List2.AddItem "Bir"  
    List2.AddItem "İki"  
    List2.AddItem "Üç"  
    List2.AddItem "Dört"  
    List2.AddItem "Beş"  
    List2.AddItem "Altı"  
    List2.AddItem "Yedi"  
    List2.AddItem "Sekiz"  
    List2.AddItem "Dokuz"  
  
    Timer1.Enabled = True  
    Timer1.Interval = 1000  
End Sub
```

```

Private Sub Timer1_Timer()
    Dim birler As Byte, onlar As Byte
    Label1.Caption = Time
    birler = Val(Mid(Label1.Caption, 8, 1))
    onlar = Val(Mid(Label1.Caption, 7, 1))
    List2.ListIndex = birler
    List1.ListIndex = onlar
    Label2.Caption = List1.Text + " " + List2.Text
End Sub

```

ÖDEV: Zar örneği



```

Private Sub Command1_Click()
    Dim sayi1 As Byte, sayi2 As Byte, t As Byte
    sayi1 = Int(6 * Rnd) + 1
    sayi2 = Int(6 * Rnd) + 1
    If sayi1 < sayi2 Then t = sayi1: sayi1 = sayi2: sayi2 = t
    Label1.Caption = sayi1
    Label2.Caption = sayi2
    Image2.Picture = Image1(sayi1 - 1)
    Image3.Picture = Image1(sayi2 - 1)
End Sub

```

ÖDEV: At Yarışı

```

Option Explicit
Dim atsayisi As Integer, f As Integer, p As Byte
Dim mesafe(50) As Integer

```

```

Private Sub Command1_Click()
    For f = 1 To atsayisi
        Image2(f).Left = 0
        mesafe(f) = 0
    Next f

```



```

        Image2(f).BorderStyle = 0
    Next f
    Cls
    Show
    For f = 9000 To 9050
        Line (f, 0)-(f, Form1.Height), vbRed
    Next f
End Sub

Private Sub Command2_Click()
    Timer1.Enabled = True
End Sub

Private Sub Command3_Click()
    Timer1.Enabled = False
End Sub

Private Sub Command4_Click()
    End
End Sub

Private Sub Form_Load()
    atsayisi = InputBox("At sayısını giriniz...")
    For f = 1 To atsayisi
        Load Image2(f)
        Image2(f).Top = f * 600
        Image2(f).Picture = Image1(0).Picture
        Image2(f).Visible = True
    Next f
    Timer1.Interval = 100
    Show
    For f = 9000 To 9050
        Line (f, 0)-(f, Form1.Height), vbRed
    Next f
End Sub

Private Sub Timer1_Timer()
    For f = 1 To atsayisi
        Image2(f).Picture = Image1(p).Picture
        mesafe(f) = mesafe(f) + Rnd * 100
        Image2(f).Left = mesafe(f)
        If Image2(f).Left + Image2(f).Width >= 8970 Then
            Timer1.Enabled = False
            Image2(f).BorderStyle = 1
            CurrentX = 1000
            CurrentY = 7000
            FontSize = 20
            FontBold = True
            Print "KAZANAN " & f & " NUMARA"
        End If
    Next f
    p = p + 1
    If p = 5 Then p = 0
End Sub

```

PROCEDURE VE FUNCTION HAZIRLAMAK VE KULLANMAK

Bir formun üzerine buton ekleyip, üzerine çift tıkladığımızda, program çalıştığında ve butona tıklandığında çalışacak kodların yazıldığı kod penceresi ekrana gelir. Bu ekranda

```
Private Sub Form_Load()
```

```
End Sub
```

Kodu görmektedir. Bu butona tıklandığında çalışacak kodun yazılabildiği bir yordamdır. Visual Basic projeleri dahilinde kullanılan nesnelere uygulama olasılığı olan olaylar için yordam kalıbı otomatik olarak uygulanmaktadır.

Forma nesne ekleme işlemi ile birlikte, projeye dahil edilen yordamlar en çok kullanılan yordamlar olmakla birlikte yeterli değildir. Programcı da kendi yordamlarını hazırlayabilir.

VB projelerinin ikinci önemli bileşeni olan BAS uzantılı module dosyalarıdır. Bu dosyalarda değişken tanımlamaları yapılan program kodları hariç bütün kodlar programcı tarafından hazırlanan yordamlardan oluşur.

Modül veya forma yordam eklemek için önce söz konusu modül veya formun kod penceresi açılır. Ardından Tools menüsündeki Add Procedure komutu kullanılır.

Sub tipli yordamlar geriye bir değer döndürmezlerken Function tipli yordamlar geriye mutlaka bir değer döndürürler. Sub ve Function arasındaki farkı anlatmak için MsgBox hazır deyimi ve MsgBox() fonksiyonundan söz edelim.

Hatırlayacağınız gibi VB de hem MsgBox deyimi:

```
MsgBox mesaj [, simgeler, [başlık]]
```

Hem de MsgBox() fonksiyonu vardı:

```
Cevap=MsgBox (mesaj [,tip] , [başlık] , [yardım dosyası,konu no]))
```

MsgBox deyimi şu şekilde kullanılabilir.

```
Private Sub Form_Click()
```

```
MsgBox "Değişiklikler Kaydedilsin mi?", 4
```

```
End Sub
```

Fakat geriye bir değer döndürmediği için bir etkisi olmaz.

Eğer kullanıcının MsgBox deyimi ile ekrana gelen diyalog kutusunda yaptığı seçimi temsil eden sayısal bir değeri bir değişkene aktarmak suretiyle yakalamak istiyorsanız, MsgBox deyimi yerine MsgBox() fonksiyonu kullanılır:

```
Private Sub Form_Click()
```

```
cevap = MsgBox("Değişiklikler Kaydedilsin mi?", 4)
```

```
End Sub
```

Şimdi kolay bir Sub tipli yordam yapalım. Form1 in kodu açılmış iken Add Procedure – MesajYaz - Sub – Public

Otomatik olarak General kısmına

```
Public Sub MesajYaz()
```

```
End Sub
```

Şeklinde yordam eklendi.

Şimdide yordam çağırıldığında çalışacak kodu yazalım:

```
Public Sub MesajYaz()
```

```
cevap = MsgBox("Değişiklikler Kaydedilsin mi?", 4)
```

```
End Sub
```

Yordamlar çalıştırılmaları için bir olay tarafından çağırılmaları gerekir.

```
Private Sub Form_Click()
```

```
MesajYaz
```

```
End Sub
```

Görüldüğü gibi Sub tipli yordamlar Sub bildiri deyimi ile başlayıp End Sub bildiri deyimi ile sonlanır. Tools menüsünden Add Procedure yi kullanmadan da bu iki bildiri deyimini kendimiz yazarak ta bir yordam oluşturabiliriz.

Yordam da bulunan bütün satırlar işletilmeden de yordamdan çıkılabilir. Bunun için Exit Sub deyimi kullanılır:

```
Private Sub Form_Click()
```

```
Topla
```

```
End Sub
```

```
Sub Topla()
```

```
Dim sayi1 As Integer, sayi2 As Integer, toplam As Integer
```

```
sayi1 = InputBox("1. sayıyı giriniz:")
```

```
If sayi1 = 0 Then Exit Sub
```

```
sayi2 = InputBox("2. sayıyı giriniz:")
```

```
If sayi2 = 0 Then Exit Sub
```

```
toplam = sayi1 + sayi2
```

```
MsgBox toplam
```

```
End Sub
```

Nesnelerle ilgili oldukları için önceden tanımlı olan bazı yordamlar parametreler içerir. Örneğin:

```
Private Sub Form_KeyPress(KeyAscii As Integer)
```

```
End Sub
```

Buradaki KeyAscii parametresi, Keypress olayında hangi tuşa basılmışsa, o tuşun ASCII kodunu içerir. Biz de bu parametre üzerinde bazı işlemler yaparak hangi koda basıldığını öğreniyoruz. Kendi hazırladığımız yordamlara da bu tip parametreler ekleyebiliriz.

Şimdi iki sayıyı parametreler aracılığı ile alıp toplayan yordamı yazalım.

```
Option Explicit
Dim sayi_bir As Long, sayi_iki As Long, sonuc As Long
```

```
Private Sub Form_Click()
    sayi_bir = 4
    sayi_iki = 5
    Topla sayi_bir, sayi_iki
End Sub
```

```
Sub Topla(sayi1 As Long, sayi2 As Long)
    sonuc = sayi1 + sayi2
    MsgBox ("Sonuç :" & sonuc)
End Sub
```

Not: parametre olarak kullanılan değişkenler ile bu değişkenlere gönderilen değerleri tutan değişkenler aynı veri tipinde tanımlanmalıdır.

Veya

```
Private Sub Form_Click()
    Topla 5, 6
End Sub
```

```
Sub Topla(sayi1 As Long, sayi2 As Long)
    sonuc = sayi1 + sayi2
    MsgBox ("Sonuç :" & sonuc)
End Sub
```

veya

```
Private Sub Form_Click()
    Call Topla(5, 6)
End Sub
```

```
Sub Topla(sayi1 As Long, sayi2 As Long)
    sonuc = sayi1 + sayi2
    MsgBox ("Sonuç :" & sonuc)
End Sub
```

Fonksiyon Hazırlamak

Fonksiyonlar ile Sub tipli yordamlar arasındaki en temel fark fonksiyonların geriye değer döndürmeleridir.

Add Procedure diyalog kutusundan bu kez Function'ı seçelim ve KokAl ismini verelim. Bu kez yordamımız şu şekilde oldu:

```
Public Function KokAl()
```

```
End Function
```

Girilen bir sayının karekökünü bulan prog:

Option Explicit

Dim sayi As Double, kok As Double

Private Sub Form_Click()

sayi = InputBox("Karekökü bulunacak sayıyı giriniz:")

kok = KokAl(sayi)

If kok = -1 Then Exit Sub

MsgBox kok, , "Kök Alma İşleminin Sonucu"

End Sub

Function KokAl(sayi As Double) As Double

If sayi < 0 Then

MsgBox "Girilen sayı negatif"

KokAl = -1

Exit Function

Else

KokAl = Sqr(sayi)

End If

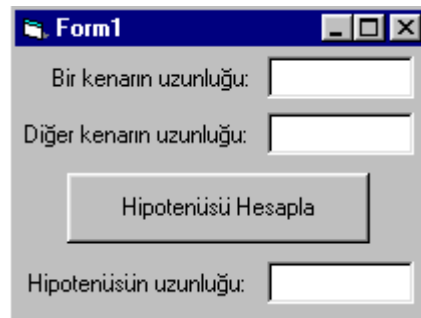
End Function

Oluşturduğumuz Function veya Sub tipli yordamın tüm proje tarafından kullanılmasını sağlamak için, sözü geçen Function veya Sub'ı bir Modülün General kısmına yazmamız gerekir.

Eğer Function veya Sub başına hiçbir şey yazmazsak veya Public yazarsak o yordam projedeki tüm formlar ve modüller tarafından kullanılabilir.

Eğer yordamın başına Private yazarsak o yordam sadece o modül içinde kullanılabilir.

ÖDEV : İki kenarı verilen dik üçgenin hipotenüsünü hesapla.
Sub ve Function kullanarak ayrı ayrı yapın.



Function ile:

```
Option Explicit
Dim a As Integer, b As Integer

Private Sub Command1_Click()
    a = Val(Text1.Text)
    b = Val(Text2.Text)
    Text3.Text = HipHesapla(a, b)
End Sub

Private Sub Form_Load()
    Show
    Text1.SetFocus
End Sub

Function HipHesapla(a As Integer, b As Integer)
    HipHesapla = Sqr((a ^ 2) + (b ^ 2))
End Function
```

Sub ile:

```
Option Explicit
Dim a As Integer, b As Integer

Private Sub Command1_Click()
    a = Val(Text1.Text)
    b = Val(Text2.Text)
    HipHesapla a, b
End Sub

Private Sub Form_Load()
    Show
    Text1.SetFocus
End Sub

Sub HipHesapla(a As Integer, b As Integer)
    Dim c
    c = Sqr((a ^ 2) + (b ^ 2))
    Text3.Text = c
End Sub
```

SPLASH FORMLARIN KULLANIMI

Eğer programımızın yüklenmesi uzun sürüyorsa, kullanıcıyı oyalayacak bir açılış formu tasarlanabilir. Örneğin Windows işletim sistemi yüklenirken bulutlar içindeki Windows logosu görüntülenir. Bütün yazılımcılar da aynı taktiği kullanarak programlarının tümü belleğe yüklenene kadar kullanıcıyı oyalayacak logo, uygulama adı ve lisans bilgileri gibi bilgiler içeren bir ekran kullanırlar.

Bunun için yapılması gereken ilk işlem bir Modül içerisinde Sub Main adında bir yordam oluşturmaktır. Ardından da görüntülenecek form tasarlanır. Biz bu formun adına Acilis diyelim. Acilis formuna şimdilik bir resim koyalım. Acilis formu tasarlandıktan sonra Sub Main yordamına şu kodu yazalım.

Option Explicit
Dim x, sayi

```
Public Sub Main()  
    Acilis.Show  
    Acilis.Refresh ' resmin yüklenmesi için kullanılır.  
    For x = 1 To 10000000  
  
        Next  
        Form1.Show  
        Unload Acilis  
End Sub
```

Hazırladığımız Acilis formu yerine VB de bulunan ve daha gerçekçi olan frmSplash 'ı kullanalım. Bunun için Project – Add Form – Splash Screen menüsü kullanılır. O zaman kodumuz şu şekilde olur:

```
Public Sub Main()  
    frmSplash.Show  
    frmSplash.Refresh  
    For x = 1 To 10000000  
  
        Next  
        Form1.Show  
        Unload frmSplash  
End Sub
```

MDI (Multiple Document Interface)

VB uygulamalarında aynı anda birden fazla pencereyi açık tutmak amacıyla MDI formlar kullanılır. MDI arabirimi tek bir form ve içinde yer alan çok sayıda alt formdan oluşur. Alt formlara MDIChild denir. Bir projeye sadece bir MDI form eklenir. Ancak bir MDI form içinde çok sayıda alt form açılabilir. MDI üzerinde komut düğmesi vb. kontroller kullanılamaz. Bu nedenle MDI formlar veri girişi için kullanılamaz.

Bir MDI uygulama yaratmak:

1. Bir proje başlatılır.
2. Project menüsünden Add MDIForm komutu seçilir.
3. Projeye yeni bir form eklenir.
4. Yeni formun MDI Child özelliği True yapılır.
Project Explorer penceresinde formların simgeleri gözlenebilir.

```
Private Sub MDIForm_Load()  
    Form1.Show  
    Form2.Show  
End Sub
```

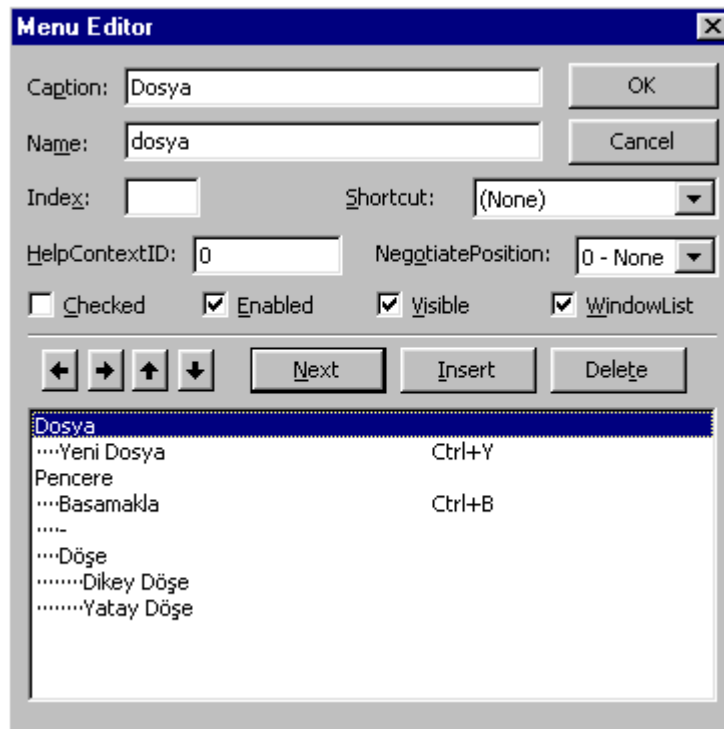
MDI Form içine PictureBox dışında hiçbir kontrolü ekleyemediğimizi gözlemleyiniz. PictureBox içine diğer tüm nesneler eklenebilir. Ve bir araç çubuğu tasarlanabilir.

MDI Form ile Notepad benzeri bir uygulama yapmaya çalışalım.

1. File menüsünden New Project tıklanır.
2. Bir MDI Form eklenir.
3. Şu anda proje dahilinde bir normal form (Form1) ve bir MDI Form (MDIForm1) olmak üzere iki form var.
4. Form1'in MDIChild özelliği True yapılır.
5. Form1 üzerine bir TextBox yerleştirilir. Sol üst kenara yanaştırılır.
6. TextBox'un MultiLine özelliği True yapılır. ScrollBar özelliği 2- Vertical yapılır.
7. Form1'in koduna şu satırlar yazılır.

```
Private Sub Form_Resize()  
    Text1.Height = ScaleHeight  
    Text1.Width = ScaleWidth  
End Sub
```

8. MDI formda bir Dosya menüsü yaratmak için Tools – Menu Editor komutunu seç.



9.

```
Private Sub yenidooya_Click()  
    Dim yeniform As New Form1  
    yeniform.Show  
End Sub
```

```
Private Sub basamakla_Click()  
    MDIForm1.Arrange vbCascade  
End Sub
```

```
Private Sub ddose_Click()  
    MDIForm1.Arrange vbTileVertical  
End Sub
```

```
Private Sub ydose_Click()  
    MDIForm1.Arrange vbTileHorizontal  
End Sub
```


FARE OLAYLARI

Private Sub Form_**MouseDown**(Button As Integer, Shift As Integer, X As Single, Y As Single)

Private Sub Form_**MouseUp**(Button As Integer, Shift As Integer, X As Single, Y As Single)

Private Sub Form_**MouseMove**(Button As Integer, Shift As Integer, X As Single, Y As Single)

Fare tuşları (düğmeleri)

Fare tuşu (Buton)

	Değer	Açıklama
VbLeftButton	1	Sol tuşa basıldı
VbRightButton	2	Sağ tuşa basıldı
VbMiddleButton	4	Orta tuşa basıldı

Fare ile birlikte klavye shift tuşları

Tuş (Shift)

	Değer	Açıklama
VbShiftMask	1	SHIFT tuşuna basıldı
VbCtrlMask	2	CTRL tuşuna basıldı
VbAltMask	4	ALT tuşuna basıldı

ÖRN:

Mouse işlemleri

MOUSE

SOL SAĞ SHIFT CONTROL ALTERNATE

Text3 Text4 Text1 Text2

☐ Kesikli/düz çizgi

CLS END

Dim x1 As Integer

Dim y1 As Integer

```
Private Sub Command1_Click()  
    Cls  
End Sub
```

```
Private Sub Command2_Click()  
    End  
End Sub
```

```
Private Sub Form_MouseDown(button As Integer, shift As Integer, x As Single, y As  
Single)  
    Line (x1, y1)-(x, y), vbRed  
    x1 = x: y1 = y  
    TEMIZ  
    If shift = 1 Then Label4.BackColor = vbRed  
    If shift = 2 Then Label5.BackColor = vbRed  
    If shift = 4 Then Label6.BackColor = vbRed  
    If button = 1 Then Label1.BackColor = vbRed  
    If button = 2 Then Label2.BackColor = vbRed  
    Text1 = x  
    Text2 = y  
    Text3 = button  
    Text4 = shift  
End Sub
```

```
Private Sub Form_MouseMove(button As Integer, shift As Integer, x As Single, y As  
Single)  
    If Check1 = 1 Then Exit Sub  
    Line (x1, y1)-(x, y)  
    x1 = x: y1 = y  
    Text1 = x  
    Text2 = y  
End Sub
```

```
Private Sub TEMIZ()  
    Label1.BackColor = vbWhite  
    Label2.BackColor = vbWhite  
    Label3.BackColor = vbWhite  
    Label4.BackColor = vbWhite  
    Label5.BackColor = vbWhite  
    Label6.BackColor = vbWhite  
End Sub
```

API FONKSİYONLARI

Windows işletim sisteminin bazı fonksiyonlarının Visual Basic uygulaması içerisinde kullanılması istendiğinde API fonksiyonlarından yararlanılır. Böylece kullanıcı işletim sistemi ile karşı karşıya gelmeden sistemin bazı fonksiyonlarını VB uygulaması içerisinde yapabilmektedir. API (Application Programming Interface)'ler Windows'un dahili fonksiyonlarıdır.

API fonksiyonları hakkında bilgi almak için API Text Viewer kullanılır.
Başlat – Programlar – MS Visual Studio 6.0 – MS Visual Studio 6.0 Tools - API Text Viewer kullanılır.

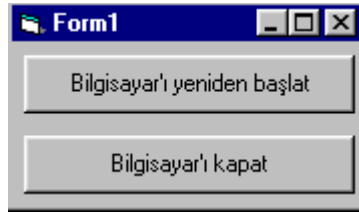
veya

VB de Add-Ins menüsünden Add-In Manager kullanılabilir. VB 6 API Viewer seçilir ve Load Behaviors çerçevesinden Loaded/Unloaded ve Load On StartUp işaretlenirse VB her açıldığında Add-Ins menüsünden de erişilebilir.

API Viewer da File – Load Text File dan Win32api.txt dosyası seçilir.

Windows'u yeniden başlatma ve kapama

- 1- API Viewer dan *ExitWindowsEx* fonksiyonu seçilir.
- 2- Form tasarlanır.



- 3- Projeye bir modül eklenir.
- 4- Kullanacağımız *ExitWindowsEx* API fonksiyonunu tanımlamak amacıyla Declare satırı modüle yapıştırılır:

Option Explicit

```
Public Declare Function ExitWindowsEx Lib "user32" _  
(ByVal uFlags As Long, ByVal dwReserved As Long) As Long
```

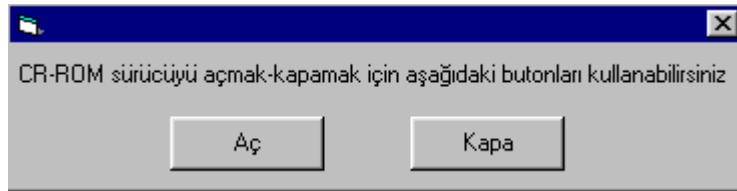
- 5- *Private Sub Command1_Click()*
 Dim sonuc As Long
 sonuc = ExitWindowsEx(2, 0)

End Sub

```
Private Sub Command2_Click()  
    Dim sonuc As Long  
    sonuc = ExitWindowsEx(1, 0)  
End Sub
```

CD-ROM sürücüyü açıp-kapatan program

- 1- API Viewer dan *mciExecute* fonksiyonu seçilir.
- 2- Form tasarlanır.



- 3- Projeye bir modül eklenir.
- 4- Kullanacağımız *mciExecute* API fonksiyonunu tanımlamak amacıyla Declare satırı modüle yapıştırılır:

```
Public Declare Function mciExecute Lib "winmm.dll" _  
    (ByVal lpstrCommand As String) As Long
```

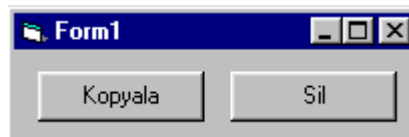
- 5- Program kodu yazılır:

```
Private Sub Command1_Click()  
    Dim sonuc As Long  
    sonuc = mciExecute("Set CDAudio door Open")  
End Sub
```

```
Private Sub Command2_Click()  
    Dim sonuc As Long  
    sonuc = mciExecute("Set CDAudio door closed")  
End Sub
```

Dosya kopyalama

- 1- API Viewer dan *CopyFile* fonksiyonu seçilir.
- 2- Form tasarlanır.



- 3- Projeye bir modül eklenir.
- 4- Kullanacağımız *CopyFile* ve *DeleteFile* API fonksiyonunu tanımlamak amacıyla Declare satırı modüle yapıştırılır:

*Declare Function CopyFile Lib "kernel32" Alias "CopyFileA" _
(ByVal lpExistingFileName As String, ByVal lpNewFileName As String, ByVal bFailIfExists
As Long) As Long*

*Declare Function DeleteFile Lib "kernel32" Alias "DeleteFileA" _
(ByVal lpFileName As String) As Long*

5- Program kodu yazılır:

```
Private Sub Command1_Click()  
    Dim sonuc As Long, kaynakdosya As String, hedefdosya As String  
    kaynakdosya = InputBox("Kopyalanacak dosyanın adını ve yolunu giriniz")  
    hedefdosya = InputBox("Yeni dosyanın adını ve kopyalanacağı yeri giriniz")  
    sonuc = CopyFile(kaynakdosya, hedefdosya, 0)  
End Sub
```

```
Private Sub Command2_Click()  
    Dim sonuc As Long, silinecekdosya As String  
    silinecekdosya = InputBox("Silinecek dosyanın adını ve yolunu giriniz")  
    sonuc = DeleteFile(silinecekdosya)  
End Sub
```

COMMON (ORTAK) DİYALOG KUTULARI

Toolbox'a eklenmelidir. Components – Controls - Microsoft Common Dialog Control 6.0

Action değişkeni	Diyalog Kutusu türü
0	Common Denetimi işlevsizdir.
1	Dosya Açma
2	Yeni Adla Kaydet
3	Renk Paleti
4	Fontlar
5	Yazdırma
6	Windows Help Penceresi

```
Private Sub Command1_Click()  
    On Error GoTo hata  
    CommonDialog1.Filter = "Gif dosyaları{*.gif}|*.gif|Jpg dosyaları{*.jpg}|*.jpg|Bmp  
dosyaları{*.bmp}|*.bmp|Ico dosyaları{*.ico}|*.ico|Tüm Dosyalar{*.}*|*.*"  
    CommonDialog1.FilterIndex = 3  
    CommonDialog1.Action = 1  
    MsgBox CommonDialog1.Flags  
    Picture1.Picture = LoadPicture(CommonDialog1.FileName)  
hata: Exit Sub  
End Sub
```

```
Private Sub Command2_Click()  
    CommonDialog1.Action = 4  
    Label1.FontName = CommonDialog1.FontName  
    Label1.FontBold = CommonDialog1.FontBold  
    Label1.FontItalic = CommonDialog1.FontItalic  
    Label1.FontSize = CommonDialog1.FontSize  
End Sub
```

```

Private Sub Command3_Click()
    CommonDialog1.Action = 3
    Form1.BackColor = CommonDialog1.Color
End Sub

```

ÖRNEK: Metin arama



```

Option Explicit
Dim ArananPoz As Integer

```

```

Private Sub cmdBul_Click()
    Ara 1
End Sub

```

```

Sub Ara(Baslangic As Integer)
    Dim poz As Integer, aranacak As String
    aranacak = txtAranacak.Text
    poz = InStr(Baslangic, txtMetin.Text, aranacak)
    If poz > 0 Then
        ArananPoz = poz
        txtMetin.SelStart = ArananPoz - 1
        txtMetin.SelLength = Len(aranacak)
        txtMetin.SetFocus
    Else
        MsgBox "Aranan Kelime Bulunamadı"
        txtMetin.SetFocus
    End If
End Sub

```

```
Private Sub cmdSonraki_Click()  
    Ara ArananPoz + 1  
End Sub
```

ÖRNEK: POPUP Menu

Menü editörden mnuFile adında bir menü tasarlanır. Mutlaka alt menüleri de olmalı.

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    If Button = 2 Then PopupMenu mnuFile, 2  
End Sub
```

```
Private Sub menu1_Click()  
    MsgBox "Menu1 e tıklandı"  
End Sub
```

```
Private Sub menu2_Click()  
    MsgBox "Menu2 ye tıklandı"  
End Sub
```

```
Private Sub menu3_Click()  
    MsgBox "Menu3 e tıklandı"  
End Sub
```

VERİ ERİŞİM YÖNTEMLERİ

VB veri erişimi için çok sayıda teknoloji geliştirmiştir. Geliştirilen her teknoloji daha kolay ve daha etkin bir biçimde verilere ulaşmayı amaçlamaktadır. Bu yöntemler şunlardır:

DAO (Data Access Objects)
RDO (Remote Data Objects)
ADO (ActiveX Data Objects)

ADO, RDO ve DAO nun gelişmiş halidir. Özellikle Web Uygulamalarında be Client/Server uygulamaları geliştirmek için kullanılır. ADO'nun amacı kolay kullanım, veri kaynaklarının esnekliği, daha iyi performans sağlamanın yanı sıra daha az sistem kaynağına gereksinim duymaktır.

SQL YAPILARI

SELECT

```
SELECT ogrencino, Isim, soyisim, dogumtarihi, dogumyeri, adresi,  
       askerlikdurumu, cinsiyeti  
FROM genelbilgiler
```

```
SELECT *  
FROM genelbilgiler
```

```
SELECT ogrencino, Isim, soyisim  
FROM genelbilgiler
```

```
SELECT sinif  
FROM okulbilgileri
```

```
SELECT DISTINCT (sinif)  
FROM okulbilgileri
```

```
SELECT ogrencino, ogrencino * 2 AS yeni  
FROM okulbilgileri
```

```
SELECT ogrencino  
FROM okulbilgileri  
WHERE ogrencino > '0131006'
```

```
SELECT ogrencino, Isim, soyisim  
FROM genelbilgiler  
WHERE Isim = 'ALİ' OR  
       Isim = 'AHMET'
```

```
SELECT genelbilgiler.ogrencino, genelbilgiler.Isim,  
       genelbilgiler.soyisim, okulbilgileri.sinif,  
       okulbilgileri.bolumprogram  
FROM genelbilgiler, okulbilgileri  
WHERE genelbilgiler.ogrencino = okulbilgileri.ogrencino
```

```
SELECT genelbilgiler.ogrencino, Isim, soyisim, sinif,  
       bolumprogram  
FROM genelbilgiler, okulbilgileri  
WHERE genelbilgiler.ogrencino = okulbilgileri.ogrencino
```

```
SELECT g.ogrencino, Isim, soyisim, sinif, bolumprogram  
FROM genelbilgiler g, okulbilgileri o  
WHERE g.ogrencino = o.ogrencino
```

```
SELECT okulbilgileri.sinif, genelbilgiler.Isim  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND  
       genelbilgiler.ogrencino BETWEEN '0131012' AND '0132013'
```

```
SELECT okulbilgileri.sinif, genelbilgiler.Isim  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND  
       genelbilgiler.ogrencino IN ('0131012', '0132013')
```

```
SELECT okulbilgileri.sinif, genelbilgiler.Isim  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND
```


genelbilgiler.ogrencino NOT IN ('0131012', '0132013')

```
SELECT okulbilgileri.bolumprogram, okulbilgileri.sinif,  
       genelbilgiler.Isim  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND  
       (genelbilgiler.Isim LIKE 'H%')
```

```
SELECT okulbilgileri.bolumprogram, okulbilgileri.sinif,  
       genelbilgiler.Isim  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND  
       (genelbilgiler.Isim LIKE 'H____') 4
```

```
SELECT okulbilgileri.bolumprogram, okulbilgileri.sinif,  
       genelbilgiler.Isim  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND  
       (genelbilgiler.Isim LIKE '%e')
```

```
SELECT okulbilgileri.bolumprogram, okulbilgileri.sinif,  
       genelbilgiler.Isim  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND  
       (genelbilgiler.Isim LIKE '%e%')
```

```
SELECT okulbilgileri.bolumprogram, okulbilgileri.sinif,  
       genelbilgiler.Isim  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND  
       (genelbilgiler.Isim LIKE '_e%')
```

```
SELECT okulbilgileri.ogrencino, okulbilgileri.bolumprogram,  
       okulbilgileri.sinif, genelbilgiler.Isim, genelbilgiler.soyisim,  
       genelbilgiler.dogumyeri  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND  
       (genelbilgiler.dogumyeri LIKE 'İS%')  
ORDER BY okulbilgileri.ogrencino
```

```
SELECT okulbilgileri.ogrencino, okulbilgileri.bolumprogram,  
       okulbilgileri.sinif, genelbilgiler.Isim, genelbilgiler.soyisim,  
       genelbilgiler.dogumyeri  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino AND  
       (genelbilgiler.dogumyeri LIKE 'İS%')  
ORDER BY okulbilgileri.ogrencino DESC
```

```
SELECT okulbilgileri.ogrencino, okulbilgileri.bolumprogram,  
       okulbilgileri.sinif, genelbilgiler.Isim, genelbilgiler.soyisim,  
       genelbilgiler.dogumyeri  
FROM okulbilgileri, genelbilgiler  
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino  
ORDER BY dogumyeri, Isim
```

```
SELECT okulbilgileri.ogrencino, okulbilgileri.bolumprogram,  
       okulbilgileri.sinif, genelbilgiler.Isim, genelbilgiler.soyisim,
```

```
        genelbilgiler.dogumyeri
FROM okulbilgileri, genelbilgiler
WHERE okulbilgileri.ogrencino = genelbilgiler.ogrencino
ORDER BY dogumyeri, Isim DESC
```

```
SELECT COUNT(*) AS COUNT
FROM genelbilgiler
```

```
SELECT COUNT(ogrencino) AS C, SUM(ogrencino) AS s
FROM okulbilgileri
```

```
SELECT MIN(ogrencino) AS k, MAX(ogrencino) AS b, AVG(ogrencino)
        AS ort
FROM okulbilgileri
```

DATABASE UPDATES

INSERT

```
INSERT INTO genelbilgiler
VALUES ('111', 'SERHAT', 'ÖZEKES ', ' 010101 ', ' İSTANBUL ',
        ' ÇOK UZUN SOK. ', 0, 1)
```

```
INSERT INTO genelbilgiler
        (Isim, soyisim)
VALUES ('SERHAT', 'ÖZEKES ')
```

UPDATE

```
UPDATE genelbilgiler
SET Isim = 'SERHAT'
WHERE ogrencino = '0111005'
```

```
UPDATE genelbilgiler
SET Isim = 'SERHAT', soyisim = 'ÖZEKES'
WHERE ogrencino = '0111005'
```

DELETE

```
DELETE FROM genelbilgiler
WHERE (ogrencino = '0111005')
```

```
DELETE FROM genelbilgiler
```

DATABASE KODLARI

MODULE1.BAS

Option Explicit

```
Global conn As ADODB.Connection
Global rs As ADODB.Recordset
Global sqlstr As String
```

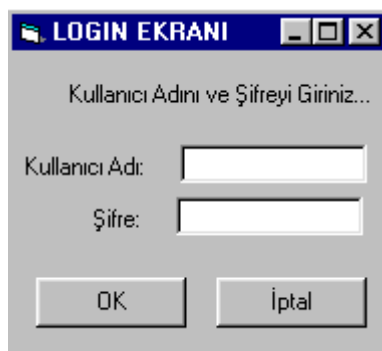
```
Sub BAGLAN()
    Set conn = New ADODB.Connection
    conn.ConnectionString = "dsn=kurs"
    conn.Open
End Sub
```

```
Sub KOP()
    conn.Close
    Set conn = Nothing
End Sub
```

```
Sub RSAC()
    Set rs = New ADODB.Recordset
    rs.CursorLocation = adUseClient
    rs.CursorType = adOpenDynamic
    rs.LockType = adLockOptimistic
End Sub
```

```
Sub RSKAPA()
    rs.Close
    Set rs = Nothing
End Sub
```

LOGIN EKRANI



Option Explicit

```
Private Sub cmdİptal_Click()
    Unload Me
End Sub
```

```
Private Sub cmdOK_Click()
    Dim SONUC
```

```

If txtLogin = "" Then
    MsgBox "Kullanici girisi eksik!", vbCritical, "Uyari"
    cmdOK.SetFocus
    Exit Sub
End If

BAGLAN
RSAC
rs.Open "SELECT * FROM kullanicilar WHERE kullanici=" + txtLogin + " AND sifre="
+ txtPass + "", conn, , , adCmdText
SONUC = rs.RecordCount
RSKAPA
KOP
If SONUC = 0 Then
    MsgBox "Yanlış Kullanıcı adı ve/veya Şifre", 16, "Uyari"
Else
    Unload LOGIN
    AnaForm.Show
End If
End Sub

Private Sub txtLogin_GotFocus()
    cmdOK.Default = True
End Sub

Private Sub txtLogin_KeyPress(KeyAscii As Integer)
    KeyAscii = Asc(UCCase(Chr(KeyAscii)))
End Sub

Private Sub txtPass_GotFocus()
    cmdOK.Default = True
End Sub

Private Sub txtPass_KeyPress(KeyAscii As Integer)
    KeyAscii = Asc(UCCase(Chr(KeyAscii)))
End Sub

```

ANA FORM

Ana Form

Öğrenci Bilgileri

Ad

Soyad

Doğum Yeri Cinsiyeti

Doğum Tarihi Askerlik Durumu

Adresi:

Öğrenci No Sınıfı

Fakülte

Bölümü / Programı

Arama Değiştir Yeni Kayıt Kaydet

Vazgeç

Sil

Öğrencilerin Listesi Sınıf Listesi

Option Explicit
Dim i, b As String

```
Private Sub cinsiyet_Click()  
    askerlikdurumu.ListIndex = IIf(cinsiyet.ListIndex = 1, 1, 2)  
    askerlikdurumu.Enabled = IIf(cinsiyet.ListIndex = 1, False, True)  
End Sub
```

```
Private Sub cmdArama_Click()  
    formarama.Show vbModal  
End Sub
```

```
Private Sub cmdDegistir_Click()  
    Frame1.Enabled = True  
    ogrencino.Enabled = False  
    cmdKaydet.Enabled = True
```

```

cmdVazgec.Enabled = True
b = "D"
End Sub

```

```

Private Sub cmdKaydet_Click()
    If b = "D" Then
        BAGLAN
        conn.Execute "Update genelbilgiler SET Isim = '" + AnaForm.isim + "',soyisim='" + AnaForm.soyisim + "',dogumyeri='" + AnaForm.dogumyeri + "',dogumtarihi='" + AnaForm.dogumtarihi + "',adresi='" + AnaForm.adresi + "',cinsiyeti='" + CStr(AnaForm.cinsiyet.ListIndex) + "',askerlikdurumu='" + CStr(AnaForm.askerlikdurumu.ListIndex) + "' WHERE (ogrencino = '" + ogrencino + "')"
        conn.Execute "update okulbilgileri set sinif='" + AnaForm.sinif + "',fakulte='" + AnaForm.fakulte + "',bolumprogram='" + AnaForm.bolumprogram + "' WHERE (ogrencino = '" + ogrencino + "')"
        MsgBox "UPDATE İŞLEMİ BAŞARIYLA TAMAMLANMIŞTIR", vbInformation, "MESAJ"
        KOP
        Frame1.Enabled = False
    ElseIf b = "Y" Then
        BAGLAN
        conn.Execute "insert into genelbilgiler (Isim,soyisim,ogrencino,dogumyeri,dogumtarihi,adresi,askerlikdurumu,cinsiyeti) values('" + AnaForm.isim + "','" + AnaForm.soyisim + "','" + AnaForm.ogrencino + "','" + AnaForm.dogumyeri + "','" + AnaForm.dogumtarihi + "','" + AnaForm.adresi + "','" + CStr(AnaForm.askerlikdurumu.ListIndex) + "','" + CStr(AnaForm.cinsiyet.ListIndex) + "')"
        conn.Execute "insert into okulbilgileri (ogrencino,sinif,fakulte,bolumprogram) values('" + AnaForm.ogrencino + "','" + AnaForm.sinif + "','" + AnaForm.fakulte + "','" + AnaForm.bolumprogram + "')"
        KOP
        MsgBox "INSERT İŞLEMİ BAŞARIYLA TAMAMLANMIŞTIR", vbInformation, "MESAJ"
    End If
    cmdKaydet.Enabled = False
End Sub

```

```

Private Sub cmdOgrenciler_Click()
    sqlstr = "select ogrencino,Isim, Soyisim from genelbilgiler order by ogrencino"
    liste.Show
End Sub

```

```

Private Sub cmdSil_Click()
    BAGLAN
    conn.Execute "delete from genelbilgiler where ogrencino='" + AnaForm.ogrencino + "'"
    conn.Execute "delete from okulbilgileri where ogrencino='" + AnaForm.ogrencino + "'"
    KOP
    With AnaForm
        .isim = ""
        .soyisim = ""
        .ogrencino = ""
        .dogumyeri.ListIndex = -1
        .cinsiyet.ListIndex = -1
        .dogumtarihi = ""
    End With

```

```

        .askerlikdurumu.ListIndex = -1
        .adresi = ""
        .sinif = ""
        .fakulte.ListIndex = -1
        .bolumprogram.ListIndex = -1
        .Image1.Picture = Nothing
    End With
    MsgBox "DELETE İŞLEMİ BAŞARIYLA TAMAMLANMIŞTIR", vbInformation, "MESAJ"
End Sub

```

```

Private Sub cmdSinifListesi_Click()
    SinifListesi.Show vbModal
End Sub

```

```

Private Sub cmdVazgec_Click()
    If b = "D" Then
        sqlstr = "select
genelbilgiler.ogrencino,Isim,soyisim,dogumtarihi,dogumyeri,adresi,askerlikdurumu,cinsiy
eti,sinif,bolumprogram,fakulte from genelbilgiler,okulbilgileri,fakulteler,bolumprogram
where genelbilgiler.ogrencino=okulbilgileri.ogrencino and okulbilgileri.ogrencino=" +
AnaForm.ogrencino + ""
        BAGLAN
        RSAC
        rs.Open sqlstr, conn, , , adCmdText
        With AnaForm
            .isim = rs.Fields("Isim")
            .soyisim = rs.Fields("soyisim")
            .ogrencino = rs.Fields("ogrencino")
            .dogumyeri = rs.Fields("dogumyeri")
            .cinsiyet.ListIndex = rs.Fields("cinsiyeti")
            .dogumtarihi = rs.Fields("dogumtarihi")
            .askerlikdurumu.ListIndex = rs.Fields("askerlikdurumu")
            .adresi = rs.Fields("adresi")
            .sinif = rs.Fields("sinif")
            .fakulte = rs.Fields("fakulte")
            .bolumprogram = rs.Fields("bolumprogram")
        End With
        RSKAPA
        KOP
        ElseIf b = "Y" Then
            With AnaForm
                .isim = ""
                .soyisim = ""
                .ogrencino = ""
                .dogumyeri.ListIndex = -1
                .cinsiyet.ListIndex = -1
                .dogumtarihi = ""
                .askerlikdurumu.ListIndex = -1
                .adresi = ""
                .sinif = ""
                .fakulte.ListIndex = -1
                .bolumprogram.ListIndex = -1
            End With

```

```

End If
Frame1.Enabled = False
cmdVazgec.Enabled = False
cmdKaydet.Enabled = False
End Sub

```

```

Private Sub cmdYeniKayit_Click()
    With AnaForm
        .isim = ""
        .soyisim = ""
        .ogrencino = ""
        .dogumyeri.ListIndex = -1
        .cinsiyet.ListIndex = -1
        .dogumtarihi = ""
        .askerlikdurumu.ListIndex = -1
        .adresi = ""
        .sinif = ""
        .fakulte.ListIndex = -1
        .bolumprogram.ListIndex = -1
    End With
    b = "Y"
    cmdKaydet.Enabled = True
    Frame1.Enabled = True
    cmdVazgec.Enabled = True
End Sub

```

```

Private Sub Form_Activate()
    cmdDegistir.Enabled = IIf(Len(Trim(ogrencino)) = 0, False, True)
    cmdSil.Enabled = IIf(Len(Trim(ogrencino)) = 0, False, True)
End Sub

```

```

Private Sub Form_Load()
    cmdKaydet.Enabled = False
    cmdVazgec.Enabled = False
    cmdSil.Enabled = False
    BAGLAN
    RSAC
    rs.Open "select * from iller", conn, , , adCmdText
    For i = 0 To rs.RecordCount - 1
        dogumyeri.AddItem rs.Fields("iladi")
        rs.MoveNext
    Next i
    RSKAPA
    RSAC
    rs.Open "select * from fakulteler", conn, , , adCmdText
    For i = 0 To rs.RecordCount - 1
        fakulte.AddItem rs.Fields("fakultead")
        rs.MoveNext
    Next i
    RSKAPA
    RSAC
    rs.Open "select bolumprogramadi from bolumprogram", conn, , , adCmdText
    For i = 0 To rs.RecordCount - 1
        bolumprogram.AddItem rs.Fields(0)
    Next i

```

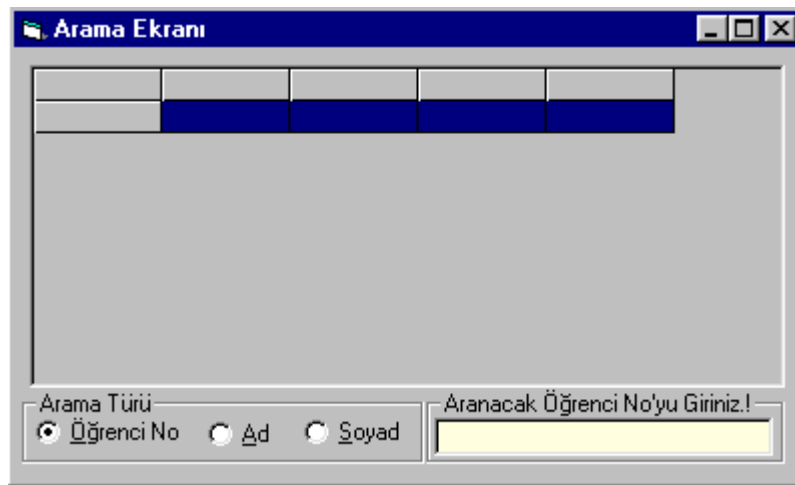


```

rs.MoveNext
Next i
RSKAPA
KOP
End Sub

```

ARAMA EKRANI



Option Explicit

```

Private Sub aranacak_KeyDown(KeyCode As Integer, Shift As Integer)
Dim kayitsayisi
If KeyCode = 27 Then Unload formarama
If KeyCode = 13 Then
If Len(aranacak.Text) = 0 Then
sqlstr = "select genelbilgiler.Ogrencino,Isim,Soyisim,Sinif from
genelbilgiler,okulbilgileri where okulbilgileri.ogrencino=genelbilgiler.ogrencino order by
genelbilgiler.ogrencino asc "
ElseIf Option1 = True Then
sqlstr = "select genelbilgiler.Ogrencino,Isim,Soyisim,Sinif from
genelbilgiler,okulbilgileri where okulbilgileri.ogrencino=genelbilgiler.ogrencino and
genelbilgiler.ogrencino=" + aranacak + ""
ElseIf Option2.Value = True Then
sqlstr = "select genelbilgiler.Ogrencino,Isim,Soyisim,Sinif from
genelbilgiler,okulbilgileri where okulbilgileri.ogrencino=genelbilgiler.ogrencino and
genelbilgiler.Isim like " + aranacak + "%""
ElseIf Option3.Value = True Then
sqlstr = "select genelbilgiler.Ogrencino,Isim,Soyisim,Sinif from
genelbilgiler,okulbilgileri where okulbilgileri.ogrencino=genelbilgiler.ogrencino and
genelbilgiler.soyisim=" + aranacak + ""
End If
BAGLAN
RSAC
rs.Open sqlstr, conn, , , adCmdText

```

```

        If rs.RecordCount = 0 Then
            MsgBox "Aradığınız Kayıt Bulunamadı"
            Set liste.DataSource = rs
        Else
            Set liste.DataSource = rs
            Set AnaForm.Image1.Picture = LoadPicture("c:\kursresimler\" &
liste.TextMatrix(liste.Row, 1) & ".jpg")
            liste.SetFocus
        End If
    RSKAPA
    KOP
End If
End Sub

Private Sub liste_DblClick()
    sqlstr = "select
genelbilgiler.ogrencino,Isim,soyisim,dogumtarihi,dogumyeri,adresi,askerlikdurumu,cinsiy
eti,sinif,bolumprogram,fakulte from genelbilgiler,okulbilgileri,fakulteler,bolumprogram
where genelbilgiler.ogrencino=okulbilgileri.ogrencino and okulbilgileri.ogrencino=" +
liste.TextMatrix(liste.Row, 1) + ""
    BAGLAN
    RSAC
    rs.Open sqlstr, conn, , , adCmdText
    With AnaForm
        .isim = rs.Fields("Isim")
        .soyisim = rs.Fields("soyisim")
        .ogrencino = rs.Fields("ogrencino")
        .dogumyeri = rs.Fields("dogumyeri")
        .cinsiyet.ListIndex = rs.Fields("cinsiyeti")
        .dogumtarihi = rs.Fields("dogumtarihi")
        .askerlikdurumu.ListIndex = rs.Fields("askerlikdurumu")
        .adresi = rs.Fields("adresi")
        .sinif = rs.Fields("sinif")
        .fakulte = rs.Fields("fakulte")
        .bolumprogram = rs.Fields("bolumprogram")
    End With
    RSKAPA
    KOP
    Unload formarama
End Sub

Private Sub liste_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = 27 Then Unload formarama: Set AnaForm.Image1.Picture = Nothing
    If KeyCode = 13 Then liste_DblClick
End Sub

Private Sub liste_RowColChange()
    Set AnaForm.Image1.Picture = LoadPicture("c:\kursresimler\" &
liste.TextMatrix(liste.Row, 1) & ".jpg")
    formarama.Caption = liste.Row & " / " & liste.Rows - 1
End Sub

Private Sub Option1_Click()
    aramafrm.Caption = "Aranacak Öğrenci No'yu Giriniz.!"

```

End Sub

Private Sub Option2_Click()

aramafrm.Caption = "Aranacak Öğrenci Adını Giriniz.!"

End Sub

Private Sub Option3_Click()

aramafrm.Caption = "Aranacak Öğrenci Soyadını Giriniz.!"

End Sub

ÖĞRENCİLERİN LİSTESİ

The screenshot shows a DataReport1 window with a report layout. The report is titled "MARMARA ÇAĞDAŞ BİLİMLER VAKFI". It consists of several sections: Report Header (Section4), Page Header (Section2), Detail (Section1), Page Footer (Section3), and Report Footer (Section5). The Detail section contains three unbound text boxes. The Page Footer section contains a page number field labeled "%p". The Report Footer section contains a label "Toplam Öğrenci Sayısı:" and a label "Label2".

Option Explicit

Private Sub DataReport_Initialize()

BAGLAN

RSAC

rs.Open sqlstr, conn, , , adCmdText

Set liste.DataSource = rs

liste.Sections("section1").Controls("isim").DataField = "Isim"

liste.Sections("section1").Controls("soyisim").DataField = "soyisim"

liste.Sections("section1").Controls("ogrencino").DataField = "ogrencino"

liste.Sections("section5").Controls("toplam").Caption = rs.RecordCount

End Sub

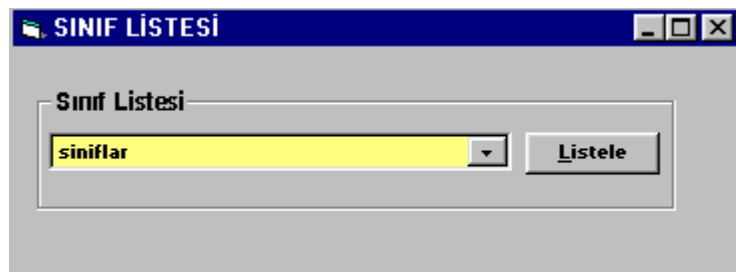
Private Sub DataReport_Terminate()

Set liste.DataSource = Nothing

RSKAPA

KOP
End Sub

SINIF LİSTESİ



Option Explicit

```
Private Sub Command1_Click()  
    Dim sube  
    If siniflar.ListIndex <> -1 Then  
        sube = Left(SinifListesi.siniflar, InStr(1, SinifListesi.siniflar, "-") - 1)  
        sqlstr = "SELECT Isim+' '+soyisim as Isimsoyisim,genelbilgiler.ogrencino FROM  
genelbilgiler,okulbilgileri WHERE genelbilgiler.ogrencino=okulbilgileri.ogrencino and  
okulbilgileri.sinif='" + sube + "' order by genelbilgiler.ogrencino "  
    ElseIf siniflar.ListIndex = -1 Then  
        MsgBox "Lütfen Bir Sınıf Seçiniz..."  
        Exit Sub  
    End If  
    Unload Me  
    rptSinifListesi.Show  
    rptSinifListesi.Sections("section2").Controls("sinif").Caption = sube  
End Sub  
  
Private Sub Form_Load()  
    Dim sqlstr, a  
    BAGLAN  
    RSAC  
    sqlstr = "SELECT distinct(sinif+'-'+bolumprogramadi) from okulbilgileri,bolumprogram  
where okulbilgileri.bolumprogram=bolumprogram.bolumprogramadi"  
    rs.Open sqlstr, conn, , , adCmdText  
    For a = 0 To rs.RecordCount - 1
```

```

    If rs.Fields(0) <> "" Then siniflar.AddItem rs.Fields(0)
    rs.MoveNext
Next
RSKAPA
KOP
End Sub

```

RPTSINIFLISTESİ

The screenshot shows the design view of a report titled "DataReport1". The report is divided into several sections:

- Report Header (Section4):** Contains the text "MARMARA ÇAĞDAŞ BİLİMLER VAKFI".
- Page Header (Section2):** Contains two labels: "Label3" and "Sınıfı Öğrenci Listesi".
- Detail (Section1):** Contains two unbound text boxes: "Öğrenci No" and "Ad - Soyad".
- Page Footer (Section3):** Contains two unbound text boxes: "Unbound" and "Unbound".
- Report Footer (Section5):** Contains two labels: "Sınıf Mecudu:" and "Label6".

Option Explicit

```

Private Sub DataReport_Initialize()
    BAGLAN
    RSAC
    rs.Open sqlstr, conn, , , adCmdText
    Set rptSinifListesi.DataSource = rs
    rptSinifListesi.Sections("section1").Controls("isimsoyisim").DataField = "Isimsoyisim"
    rptSinifListesi.Sections("section1").Controls("ogrencino").DataField = "ogrencino"
    rptSinifListesi.Sections("section5").Controls("mevcut").Caption = rs.RecordCount
End Sub

```

```

Private Sub DataReport_Terminate()
    Set rptSinifListesi.DataSource = Nothing

```

RSKAPA
KOP
End Sub

Microsoft Visual Basic Nedir?	Componentler (Kontrol Elemanları)
DEĞİŞKENLER	Textbox (Metin Kutusu)
MsgBox (Mesaj Kutusu)	Label (Etiket)
InputBox (Bilgi Giriş Kutusu)	Command Button (Komut Düğmesi)
Döngüler	CheckBox (İşaret Kutusu)
If Then Else End if	Option Button (Seçenek Düğmesi)
Select Case	Horizontal & Vertical ScrollBar (Kaydırma Çubuğu)
For Next	Shape (Şekil Kontrol Elemanı)
QBColor ()	Timer Control (Zamanlayıcı)
RGB ()	Frame Control (Çerçeve)
Like	UpDown(Yukarı aşağı arttırma)
Disk ve Dosya Komutları	Line (Çizgi)
CurDir	ComboBox (Açılan Liste)
ChDir	ListBox (Listeleme Kutusu)
ChDrive	DriveListBox (Sürücü Listeleme Kutusu)
MkDir	Directory List Box (Dizin Listeleme Kutusu)
RmDir	FileList Box (Dosya Listeleme Kutusu)
FileCopy	PictureBox (Resim Kutusu)
FileLen	Image (Resim Gösterme Kontrolü)
FileDateTime	Common Dialog (Diyalog Pencereleleri)
GetAttr	MSMasked (Formatlı Giriş)
SetAttr	SSTab
Name	Richtextbox
Dir	Popup Menü
Kill	MsFlexGrid
End	ProgressBar (İlerleme Çubuğu)
Çizim Yapmak	Slider
Line	DateTimePicker
Circle	MonthView
Pset	Toolbar
Random Dosyalar	Coolbar
Fonksiyonlar	Statusbar
Abs Fonksiyonu	Veri Tabanı Uygulamaları
Asc Fonksiyonu	DBGrid
Chr Fonksiyonu	ADO ile Veri Tabanı
Date Fonksiyonu	DataGrid
DateDiff Fonksiyonu	Data Report
Int Fonksiyonu	SQL Komutları
LCase Fonksiyonu	Formlar
Left Fonksiyonu	MDIFormlar
Len Fonksiyonu	
Ltrim, Rtrim ve Trim Fonksiyonu	
Mid Fonksiyonu	
Now Fonksiyonu	
Round Fonksiyonu	
Space Fonksiyonu	
Shell Fonksiyonu	
Sqr Fonksiyonu	
String Fonksiyonu	

Time Fonksiyonu	
UCase Fonksiyonu	
Val Fonksiyonu	
Dizi Değişkenler	
Dinamik Diziler	
Function Oluşturma	

Microsoft Visual Basic Nedir?

Visual Basic bir programlama dilidir. Visual Basic'in bir programın yazılıp çalıştırılmasından öte, program geliştirmek için çok sayıda araca sahip olması; kullanıcı arabiriminin tasarlanması, hataların giderilmesi, veritabanı yaratmak, farklı türde programlar üretmek gibi gelişmiş özellikleri, onu bir "programlama ortamı" ya da "program geliştirme ortamı" olarak tanımlamamıza neden olur.

Visual Basic, çok amaçlı ve aynı zamanda kolay bir şekilde program geliştirmenizi sağlayan bir programlama ortamıdır. Visual olmak (görsellik), simgelerle yapılan iletişimi ifade eder. Visual Basic programlarında ortamın tasarımı öncelikli olarak yapılan işlemdir. Kod yazmak ikincil kalır.

Visual Basic ile küçük işlemler için bir program geliştirilebileceği gibi, bütün dünyaya uzanabilecek bir internet uygulaması yada büyük firma düzeyinde kullanılacak gelişmiş bir enterprise (kurumsal) uygulaması geliştirilebilir. Örneğin bir cari hesap, stok, satış, imalat vb. uygulamaları, oyun programı vb.

Visual Basic'in sahip olduğu araçlar:

1. Kullanıcı ara birimi geliştirmek için çok sayıda kontrol,
2. Programı derleme ve hata bulmaya yönelik çok sayıda araç,
3. Değişik veritabanlarına erişim,
4. SQL Server ve diğer enterprise düzeyindeki veritabanlarına erişim,
5. Diğer uygulamaların fonksiyonlarını kullanmayı sağlayan ActiveX teknolojisi,
6. Uygulamanın EXE dosya olarak üretilmesi ve dağıtılması,
7. Raporlama araçları,
8. Veri yapısı araçları,
9. Web uygulaması geliştirme araçları,
10. Çok katmanlı programlama ve Microsoft Transaction Server.

Visual Basic'in Özellikleri

Özellikle Windows ortamında programlama geliştirme aracı olarak kullanılan Visual Basic, diğer klasik programlama dillerine göre farklı özelliklere sahiptir. Visual Basic ile program geliştirmek, öncelikle Windows arabiriminin kullanıcı etkileşiminde kullanılan araçların kullanılması anlamına gelir. Klasik programlama dilleri ise genellikle karakter tabanlı (DOS ve UNIX) işletim sistemlerinde kullanılan özellikle klavye ağırlıklı kullanıcı arabirimine sahiptir. Gelişmiş programlama ortamı olan Visual Basic, Windows uygulaması gibi mönülere, düğmelere sahiptir.

Visual Basic sadece standart kontroller sağlamaz. Aynı zamanda çok sayıda üçüncü parti yazılımcıların da hazırladığı kontrolleri de kullanabilir. Bu kontroller satın alınarak kolayca programlara eklenebilirler. Visual Basic, birçok OCX(OLE Control Extensions) kontrol ve ActiveX kontrolleri destekler.

Bir Visual Basic Programının Kısımları

Bir program belli kısımlardan oluşur:

- Kullanıcı arabirimi
- Bilgi işleme (hesaplamalar)
- Bilgi saklama (veritabanı işlemleri)
- Raporlama işlemleri

Kullanıcı arabirimi bir programın ekranını oluşturur. Diğer bir deyişle kullanıcıların bilgi girdiği etkileşimli ekranlardır. İyi bir kullanıcı arabirim geliştirmenin yöntemleri şunlardır.:

- Anlaşılabilir, kolay ve çok yönlü kullanılabilir olması
- Veriyi net bir biçimde ifade etmeleri
- Komutları açıkça sergilemeleri,
- Menüler, komutlar ve düğmeleriyle programın diğer bölümleriyle uyum içinde olmaları,
- Menü, araç çubukları, fare ve klavye aracılığıyla kullanılabilir olması.

İyi Bir Programın Nitelikleri

Programlar belli bir işin bilgisayar tarafından yerine getirilmesini sağlarken; programların belli niteliklere sahip olması gerekir. İyi program niteliklerine sahip olmak, aynı zamanda program geliştirme sürecinin de amaçlarındandır. İyi bir programın nitelikleri:

- Görsellik
- Kolaylık
- Doğruluk
- Hızlı
- İyi bir belgeleme
- Kolayca değiştirilebilme, güncellenebilme
- Etkin bir kodlama
- Etkin bir işletim (daha az sistem kaynağına gereksinim duyması)

Bunun dışında programın analiz çalışmalarının ve kodlarının iyi bir şekilde belgelenmesi gerekir (söz uçar, yazı kalır). Belgeleme belli bir sistem içinde bir kitapçık olarak hazırlanmalı ve kod içinde de açıklamalara çok önem verilmelidir.

Çalışma Ortamı

Visual Basic 6.0 ortamı özellikle program birimlerinin düzenlenmesi için çok sayıda bölüme sahiptir. Bu bölümler aracılığıyla programın tasarımı yapılır ve kod yazılır. Çalışma ortamının hemen her alanında farenin sağ tuşuna basılarak içerik menüleri elde edilir. Bu menüler bulunulan yere göre içeriği değişebilen özel menülerdir. Bütün olanaklar grafik yardımı ile programın arabiriminin geliştirilmesini, çalıştırılmasını ve hata düzeltmelerinin yapılmasını sağlar.

Visual Basic programı bir proje olarak geliştirilir. Bu proje içinde formlar, kontroller, modüller ve komponent' ler bulunur. Formlar üzerine kontroller yerleştirilir. Ardından kontrollerle ilişkili olarak olay yordamları yazılır. İstenirse genel yordamların yer aldığı modüller yazılır ve proje çalıştırılır.

Project ve Formlar

Yeni bir Visual Basic programına bir proje olarak başlanır. Varsayılan proje türü standart EXE' dir. Proje penceresiyle birlikte bir form ekrana gelir. Projenin başlangıç adı Project1, formun başlangıç adı ise Form1'dir. Her Visual Basic uygulamasında bir form bulunur. Form üzerine denetimler ve grafikler eklenerek uygulama geliştirilir. Her bir form aynı zamanda form tasarımının yapıldığı bir alandır. Formlar uygulamanın kullanıcı arabirimini oluşturur. Visual Basic kontrollerinden istenilen kontroller form üzerine fare ile yerleştirilerek uygulamanın arabirimi oluşturulur. Bu arabirim üzerinde metin kutuları (textbox), komut düğmeleri(command buttons) gibi veri işlem elemanları yer alır. Ardından istenilen olaylara göre kodlar yazılarak programın yazılması sağlanır.

Araç Kutusu (Toolbox)

Görsel programlama ortamlarında görsel kontrollerin çizilmesi için zengin bir araç kutusu kullanılır. Visual Basic araç kutusunda programın tasarımı sırasında kullanılmak üzere çok sayıda aracı içerir. Bu araçlar kontrol (control) olarak adlandırılır. Standart olarak Visual Basic açıldığında ekrana gelen kontroller aşağıda gösterilmiştir. Ayrıca diğer kontrollerden de eklemek isterseniz; toolbox üzerinde farenin sağ tuşuna basarak Component menüsünden seçebilirsiniz.

Standart Kontrol İşlevi

PictureBox	Resim kutusu. Bitmap, Icon, GIF resimlerin gösterilmesini sağlar
Label	Etiket. Sadece bilgi vermek yada görüntü olarak form üzerinde durur. Genellikle metin kutularının adlarını göstermek için kullanılır.
TextBox	Metin Kutusu. Veri girişi için kullanılan bir kontroldür. Hemen hemen bütün programlarda kullanıcılardan veri girişi, değiştirme yapması gereken alanlar gerekir. İşte metin kutusu bu amaçla kullanılır.
Frame	Çerçeve. Kontrolleri gruplamak için kullanılır.
CommandButton	Komut düğmesi. Bir işlemin başlatılmasını sağlar. Genellikle fare üzerine tıklanarak kullanılır.
CheckBox	İşaret Kutusu. Bir seçim yapmayı sağlar.
OptionBox	Seçenek Kutusu. Seçim yapmayı sağlar.
ComboBox	Açılan Kutu.
ListBox	Liste. Birçok elemanı liste biçiminde göstermek için kullanılır.
HScrollBar	Yatay kaydırma çubuğu. Yatay olarak kullanılan kaydırma çubuğu
VScrollBar	Dikey kaydırma çubuğu. Dikey olarak kullanılan kaydırma çubuğu
Timer	Zaman saati. Zamanın takip edilmesini sağlar.
DriveListBox	Sürücü listesi. Sistemdeki sürücülerini listeler.
DirListBox	Dizin listesi. Sistemdeki dizinleri (klasörleri) listeler.
FileListBox	Dosya listesi. Bir dizin içindeki dosyaları listeler.
Shape	Şekil. Şekil çizmeyi sağlar.
Line	Çizgi. Çizgi çizmek için kullanılır.
Image	Resim gösterme. Sadece resimleri göstermek için kullanılır.
Data	Data. Verileri göstermek için kullanılır.
OLE	OLE

MÖNÜLER

File Menüsü

New Project	Yeni bir projeye başlamayı sağlar.
Open Project	Daha önce kaydedilmiş bir projeyi açar.
Add Project	Mevcut projeye yeni projeler ekler.
Remove Project	Projeyi ve elemanlarını ortadan kaldırır.
Save Project Group	Projeleri grup kayıt eder.
Save Project Group As	Projelere yeni ad vererek kaydeder.
Save Form1	Aktif durumdaki formu kaydeder.
Save Form1 As	Aktif durumdaki formu değişik adla kaydeder.
Save Selection	Seçili kaydet.
Save Chance Script	Transact-SQL deyimlerinin kayıt edilmesini sağlar.
Print	Form görüntülerinin veya kod sayfalarının yazdırılmasını sağlar.
Print Setup	Yazıcı ayarları.
Make Project1.exe	Projenin EXE dosyası haline getirilmesini sağlar.

Make Project Group	Projeleri bir grup olarak birleştirir.
Exit	Visual Basic'ten çıkış.
Edit Mönüsü	
Undo	Son yapılan işlemi geri alır.
Redo	Geri alınan son işlemi tekrarlar.
Cut	Seçili olanları keserek panoya atar.
Copy	Seçili olanları kopyalayarak panoya atar.
Paste	Daha önce kesilmiş yada kopyalanmış olanları yapıştırır.
Paste Link	Daha önce kopyalanmış olanları bağlantılarını oluşturur.
Remove	Seçilen elemanları siler.
Delete	Seçilen elemanları siler.
Delete table from Database	Tabloyu veritabanından siler.
Select All	Form üzerindeki yada kod penceresindeki bütün elemanlar seçilir.
Select All Columns	Bütün kolonları seçer.
Table	Tablo komutlarını gösteren bir alt mönü.
Find	Kod penceresinde istenilen bilgiyi bulur.
Find Next	Bir sonrakini bulur.
Replace	Bulunan bir sözcüğü değiştirmeyi sağlar.
Indent	Satırı içerden başlatır.
Outdent	Dışarı alır.
Insert File	Koda bir dosyadan ekleme sağlar.
List properties / Methods	Yazılan nesnelerin özelliklerinin açılan bir liste olarak ekrana getirilmesini sağlar.
List Constants	Bir özelliğin alacağı değerleri listeler.
Quick Info	Bir deyimin kullanım biçimini gösterir.
Parameter Info	Bir deyimin parametrelerini gösterir.

View Mönüsü

Code	Kod penceresini görüntüler.
Object	Formu görüntüler.
Definition	Kod.
Last Position	Bir önceki konuma gidilmesini sağlar.
Object Browser	Mevcut proje yada Visual Basic nesnelerinin seçilmesini sağlayan görsel yardımcı.
Immediate Window	Programın kesilmesi yada hata bulunduğu (Debug durumunda) istenilen değişkenlerin değerlerinin izlenebildiği pencere.
Locals Window	Yine debug durumunda yerel değişkenlerin değerlerinin izlenmesini sağlar
Watch Window	Bir ifadenin durumunun izlenmesini sağlar.
Call Stacks	Çağırılmış ama tamamlanmamış procedure'leri gösterir.
Project Explorer	Proje içinde yer alan elemanlar görüntülenir.
Properties Window	Özellikler penceresi görüntülenir.
Form Layout Window	Formun ekrandaki görüntüsünü gösterir.
Property Pages	Özellikler tablosunu gösterir.
Table	Veritabanı, query gibi ortamlarda seçilen penceresinin açılıp kapanmasını sağlar.
Zoom	Büyütme.
Show Panes	Bölmeleri göster.

Toolbox	Araç kutusunun görüntülenmesi / görüntülenmemesi sağlanır.
Data View Window	
Color Palette	Form üzerinde renklendirme yapmak için kullanılan renk paleti ekrana gelir.
Toolbars	Araç çubuklarının görüntülenmesi için kullanılır.
Visual Component Manager ..	

Project Mönüsü

Add Form	Projeye yeni bir form ekler.
Add MDI Form	Projeye yeni bir MDI form ekler.
Add Module	Projeye yeni bir modül ekler.
Add Class Module	Projeye yeni bir modul ekler.
Add User Control	Projeye yeni bir kullanıcı kontrolü ekler.
Add Property Page	Projeye yeni bir property ekler.
Add User Document	Projeye yeni bir user document ekler.
Add ActiveX Desingner	Projeye yeni bir ActiveX Designer ekler.
Add File	Projeye dosya ekler.
Remove <form>	İstenilen eleman kaldırılır.
References	Çeşitli kütüphanelerin yüklenmesi sağlanır.
Components	Araç kutusuna eklenecek bileşenleri gösterir.
<Proje> Properties	Projenin özellikleri düzenlenir.

Format Mönüsü

Align	Form üzerindeki birden çok elemanın birbirine hizalanmasını sağlar.
Make Same Size	Form üzerindeki birden çok elemanın birbirine göre aynı boyuta getirilmesini sağlar.
Size to Grid	Form üzerindeki elemanların form üzerindeki ızgaraya göre boyutlarını ayarlar.
Horizontal Spacing	Form üzerinde birden çok eleman arasındaki yatay boşluğu ayarlamak için kullanılır.
Vertical Spacing	Form üzerinde birden çok eleman arasındaki dikey boşluğu ayarlamak için kullanılır.
Center in Form	Form üzerindeki elemanların forma dikey yada yatay olarak ortalanmasını sağlar.
Order	Form üzerinde üst üste duran elemanların sırasını belirler.
Lock Controls	Form üzerindeki elemanların kilitlenerek yerlerinin oynamamasını sağlar.

Debug Mönüsü

Step Into	Programın satır satır çalışmasını sağlar. F8 tuşu.
Step Over	Programın procedure olarak çalışmasını sağlar.
Step Out	Programın
Run to Cursor	Programı bulunulan satıra kadar çalıştırır.
Add Watch	Program çalışırken değişkenlerin değerlerini kayıt eder ve izlenmesini sağlar.
Edit Watch	İstenilen izleme satırını düzenler.
Quick Watch	Değişkenlerin seçildikleri andaki değerleri görüntülenir.
Toggle Breakpoint	Bir satıra kesme noktası işareti vererek programın bu satıra durması sağlanır.
Clear All Breakpoints	Bütün kesme noktalarını kaldırır.
Set Next Statements	jkh
Show Next Statements	

Run Mönüsü

Start	Programı çalıştırır. F5 tuşu.
Start With Full Compile	Programı derler / çalıştırır.
Break	Programı keser.
End	Programı sonlandırır.
Restart	Programı yeniden başlatır.

Query Mönüsü

Run	Sorguyu çalıştırır.
Clear Results	Sorguyu temizler.
Verify SQL Syntax	SQL deyimini denetler.
Group By	Guruplar.
Change Type	Sorgu tipini değiştirir.
Add to Output	Seçilen kolonu giriş bilgisine ekler.
Sort Ascending	Artan sırada sıralar.
Sort Descending	Azalan sırada sıralar.
Remove Filter	Filtreyi kaldırır.
Select All Rows From	Bütün satırları seçer.

Diagram Mönüsü

New Text Annotation	Metin açıklama ekler.
Set Text Font	Metin yazı tiplerini düzenler.
Add related Tables	İlişkili tablolar ekler.
Show Relationship Labels	İlişki etiketlerini ekler.
Modify Custom View	Özel görünümü değiştirir.
View Pages Breaks	Sayfa aralarını (sonlarını) göster.
Recalculate Page Breaks	Sayfa aralarını (sonlarını) yeniden hesaplar.
Arrange Selection	Yeniden düzenler.
Arrange Tables	Tabloları düzenler.
Autosize Selected Tables	Tabloları yeniden boyutlandırır.

Tools Mönüsü

Add Procedure	Kod penceresinde yeni bir prodecure yaratmak için kullanılır.
Procedure Attribute	Özellik ve metotları düzenler.
Menu Editör	Bir mönüyü tasarlayan iletişim kutusu.,
Options	Visual Basic ortamını düzenlemek için kullanılır.

Add-Ins Mönüsü

Visual Data Manager	Veritabanı düzenleme aracı.
Add-In Manager	Ek programları ekleme/kaldırma.

Window Mönüsü

Tile Horizontally	Yatay döşeme.
Tile Vertical	Dikey döşeme.
Cascade	Basamaklama.
Arrange Icons	

Help Mönüsü

Contents	İçerikler.
Indx	İndeks.
Search	Arama.
Technical Support	Teknik destek seçenekleri.
Microsoft on the Web	Microsoft 'un Web sitesine bağlantı.
About Microsoft Visual Basic.	Visual Basic programınız hakkında bilgiler.

Options Mönüsü

Editör Sekmesi

Auto Syntax Check	Yazılan kodların otomatik olarak yazım denetiminin yapılmasını sağlar.
Require Variable Declaration	Kod içinde kullanılan bütün değişkenlerin tanımlanmasını zorunlu kılar.
Auto List Members	Nesnelerin özelliklerinin otomatik olarak listelenmesini sağlar.
Auto Quick Info	Bir deyimın yazılması durumunda parametrelerinin ekrana gelmesini sağlar.
Auto Data Tips	Program kesildikten sonra (debug modda) fare ile üzerine gidilen değişkenin değerini gösterir.
Auto Indent	İçerden başlamayı sağlar.
Drag-and-Drop Text Editing	Kod satırları üzerinde fare ile seçip sürüklemek işleminin yapılmasını sağlar.
Default to Full Module View	Bir procedure yada bütün procedure'ların görülmesini sağlar.
Procedure Separator	Procedure'lar arası ayraç.
Editör Format Sekmesi	
Normal Text	Normal kod satırlarının rengini düzenler.
Selection Text	Seçilen kod satırlarının rengini düzenler.
Syntax Error Text	Sözdizimi hatası olan kod satırlarının rengini düzenler.
Foreground	Yazı rengi.
Background	Zemin rengi.
Indicator	Göstergenin rengi.
General Sekmesi	
Show Grid	Form üzerinde kareleme yapılarak kontrollerin hizalanmasına yardımcı olur.
Align Controls to Grid	Form üzerindeki kontrollerin hizalarının gridlere dayanmasını sağlar.
Break on All Errors	Derleyici herhangi bir hata ile karşılaştığında programı keser.
Break in Class Module	Class modulünde programın kesilmesi.
Break on Unhandled Errors ...	Bulunamayan hatalar oluştuğunda da programın kesilmesini sağlar.
Compile on Demand	İstendiğinde derleme.
Background Compile	Derleme arka planda yapılır.
Show Tooltips	Fare işaretçisinin herhangi bir kontrol elemanı üzerine gelmesi ile o kontrol hakkında bilgi verici bir metnin görüntülenmesi sağlanır.
Collapse Proj.Hide Window ...	Pencerelerin diğerleri arasına yuvalanarak açılmasını sağlar. İşaretlenmezse pencere diğerlerinin üzerinde bir yerde açılır.
Docking Sekmesi	
Immediate Window	Anlık görünüm penceresi.
Locals Window	Yerel değişkenleri penceresi
Watch Window	İzleme penceresi.
Project Explorer	projenin elemanlarını gösteren pencere.
Properties Window	Özellikler penceresi.
Object Browser	Nesne gösterici.
Form Layout	Formun ekrandaki görünümü.
Toolbox	Araç kutusu.
Color Palette	Renk Paleti.
Environment Sekmesi	
Visual Basic başladığında	
Prompt for project	Visual Basic'in açılmasının ardından hangi tip proje yaratılacağı sorulur.

Create default project AutoLoad dosyasına kayıt edilmiş dosyalardan oluşan bir proje başlatılır.

Bir program başladığında

Save Changes Program çalıştırılmadan önce yapılan değişiklikler kayıt edilir.

Prompt to Save Changes Program çalıştırılmadan önce yapılan değişiklikler kayıt edilmesi için soru sorulur.

Don't Save Changes Program çalıştırılmadan önce yapılan değişiklikler kayıt edilmez.

Şablonları göster

Forms Formlar.

MDI Forms MDI Formlar.

Modules Modüller.

Class Modules Class Modüller.

User Controls Kullanıcı Kontrol Modülleri.

Property Pages Özellik Sayfaları.

User Document Kullanıcı Dokument Modülleri.

Template Directory Şablonların saklanacağı klasör.

Matematiksel İşaretler

^	Üs alma
*	Çarpma
/	Bölme
\	Tam Bölme (Bölme sonucunun tam kısmını alır)
+	Toplama
-	Çıkarma
=	Eşittir
< , <=	Küçük, küçük eşit
> , >=	Büyük, büyük eşit
<>	Eşitdeğil

İlk Programım

Bir form açılır. Kod sayfasını açmak için formun üzerine çift tıklanır, Project Explorer penceresinden View Code tıklanır veya F7 tuşuna basılır.

Bir yordamın başlama ve bitirme deyimleri vardır.

Private Sub Form_Load()

End Sub

SHOW

Sadece Form_Load yordamında kullanılır ve form yüklendiği anda ekrana yazdığımız yazının gelmesi için kullanılır.

PRINT

Ekrana sabit bir bilgiyi yazmak için kullanılır.

ÖRNEK

- 1- Show
Print "Microsoft Visual Basic Dersine Hoş Geldiniz"
 - a. Fontbold=False
 - b. Fontitalic=True
2. Print "Visual Basic Öğretmeni"
3. Print "Oguz KUTLU"
 - a. Fontsize=24
 - b. Fontname="Times New Roman"
4. Print "PRESTIGE EDUCATION CENTER"
 - a. Fontunderline=True
 - b. Fontname="Arial"
5. Print "Bilgisayar Teknik Eğitim Merkezi"
ForeColor=RGB(214,124,35)
1. Print "İngilizce Eğitim Merkezi"

NOT : Dikkat edileceği gibi kodlar yordam içerisinde yukarıdan aşağıya doğru ilerler. Yukarıda yapılan bir yazı büyüklüğü veya yazının rengi komutu aşağı kodlarda değiştirilmezse aynen uygulanır. Aynı kodları bir Command butonunun Click yordamına yazarsak baştaki Show komutunu yazmamıza gerek kalmaz.

Yazdırdığımız yazıları formumuzun herhangi bir noktasına yazdırabiliriz. Bunun için kullanılacak deyim **TWIP** komutudur. Twip komutu aslında Formun bir özelliği olan **Form1.scalewidth** ve **Form1.scaleheight** komutlarını kullanarak yapar.

- 1 inch = 1440 Twip
- 1 inch = 2,54 cm.
- 1 Twip = 0,0018 cm.

Şimdi yukarıdaki örneğimize şu satırları ekleyelim.

- a. CurrentX = 2500
- b. CurrentY = 2500

Formun en üstteki mavi bant hariç twip cinsinden ekran boyutları yaklaşık olarak

	640 X 480	800 x 600	1024 X 768
Form1.scalewidth	9600	12000	16000
Form1.scaleheight	7200	9000	12000

DEĞİŞKENLER

Visual Basic program içersine yazılacak olan değişkenleri tanımlama zorunluluğu vermemektedir, ama büyük programlarda tanımlanmayan değişkenler bir sorun yaratabilirler. Bunun için Visual Basic'in tanıdığı değişkenler aşağıda gösterilmiştir. Bir procedure içinde tanımlanan değişkene local (yerel) değişken denir. Bu değişken bu procedure aktif olduğu zaman aktif olur. Bir değişkenin tüm modüllerde de geçerli kılmak için değişken tanımı Declarations kesiminde tanımlanır. Bunun için değişkeni PUBLIC deyimi kullanılarak tanımlamak gerekir. Yerel değişkenleri de STATIC deyimi ile tanımlarsak uygulama boyunca aktif olurlar.

Public (Değişken) As (Değişken Tipi) `Yalnız Declarations kısmında tanımlanabilir.

Static (Değişken) As (Değişken Tipi)

Dim (Değişken) As (Değişken Tipi)

Değişkenlerin özellikleri

1. Bir harf ile başlar
2. En çok 255 karakterden oluşur.
3. İşaretler içermez.
4. Yazıldığı yordam içinde tek olmalı
5. Değişken adı olarak Visual Basic in kullandığı herhangi bir kod kullanılamaz.
6. Eğer değişken Declarations yordamına yazılırsa tüm procedure içinde geçerli olur.

Değişken Tipi	Veri Tipi	Alabileceği En Büyük Değer	Boyutu (Byte)	Değişken Tipi Karakter
Byte	İşaretsiz tam sayı	0 - 255	1	Yok
Integer	Tamsayı	± 32.767	2	%
Long	Uzun tamsayı	$\pm 2.147.483.647$	4	&
Single	Sayısal değişken	$\pm 3,4 \times 10^{38}$	4	!
Double	Sayısal değişken	$\pm 1,7 \times 10^{308}$	8	#
Currency	Para birimi	$\pm 922..337.203.685.447$ kuruş bölümünde 4 hane gösterir	8	@
Boolean	Evet-Hayır Olumlu-Olumsuz		2	Yok
Date	Tarih formatlarında	01/01/100 – 31/12/9999	8	Yok
String	Sayısal içeriği olmayan değişken	65.000 karakter	Her karakter 1 Byte	\$
Variant	Sayısal içeriği olmayan değişken		16 + Her karakter 1 Byte	Yok
Object	Resim ve image'ler için kullanılır.		4	Yok

Sabit Değişken

Değişkenlerin aksine sabitler (constant), sahip oldukları değerler ile kullandığımız bilgi tanımlamalardır. Özellikle belli değerlerin kullanımını sağlar. Kodlamayı ve kodun okunabilirliğini kolaylaştırır. Bir sabitin uygulama boyunca geçerli olması için Declarations kesiminde tanımlanmalı ve Const deyiminden önce Public deyimini kullanılır.

ÖRNEK

```
Const Pi = 3.14159265
Const Katsayı = 2,5
Const KDV = 1,17
```

String

Karakterlerden oluşan veriler String olarak tanımlanır. Örneğin müşteri adı gibi bir bilgi String olarak tanımlanmalıdır. String veriler sayıları içerebilir ancak sayısal işleme giremezler. String değişkenlere değer atamak için çift tırnak karakteri kullanılır. Bir String değişkene belli bir uzunluk da verilebilir. Bunun için asteriks (*) karakterinden sonra uzunluk birimi girilir.

```
Dim s As String
```

```
S = "Oguz KUTLU"
```

```
Dim Adı As String * 20
```

ÖRNEK

1.

```
Show
Dim a As String
Dim b As String
Dim c As String
```

```
a = 168
b = 122
c = a + b
Print c
```

```
Sonuç = 168122 olur.
```

2.

```
Show
Dim a As Integer
Dim b As Integer
Dim c As Integer
```

```
a = 168
b = 122
c = a + b
```

1.1

```
Show
```

```
a$ = 168
b$ = 122
```

```
c$ = a + b
Print c
```

```
Sonuç = 168122 olur.
```

2.1

```
Show
```

```
a% = 168
b% = 122
```

```
c% = a + b
Print c
```

Print c

Sonuç = 290 olur.

Sonuç = 290 olur.

Bazen programlarımızda değişkenlerin tiplerini değiştirmemiz gerekebilir. Bunun için aşağıdaki kodlar kullanılır.

CByte Değişkeni Byte tipine çevirir
CInt Değişkeni İnteger tipine çevirir
CLng Değişkeni Long tipine çevirir
CSng Değişkeni Single tipine çevirir
CDBl Değişkeni Doble tipine çevirir
CCur Değişkeni Currency tipine çevirir
CBool Değişkeni Boolean tipine çevirir. Bu özellik pek kullanılmaz.
CDate Değişkeni Date tipine çevirir
String Değişkeni String tipine çevirir
CVar Değişkeni Varriant tipine çevirir

Kullanım şekli

CCur (Değişken adı)

ÖRNEK

Dim a As Single,

....

....

a = CDBl (a)

NOT : Değişken tiplerini çevirirken dikkat edilecek en önemli nokta özellikle büyük değişkenlerin küçüklere çevrilmesi sırasında veri kaybının olmaması veya virgüllü sayıların tam sayılara dönüştürülmesi sırasında oluşabilecek veri kayıpları.

VarType () Fonksiyonu

Tipi belirtilmeyen değişkenlere programın işletimi sırasında Variant değişken tipini alır. VarType fonksiyonu ile Variant bir değişkenin hangi tipte bilgi içerdiği öğrenilir.

VarType (Variant Değişken Adı)

Veri Tipi	Sayısal Değer
Boş – Empty	0
Null	1
Integer	2
Long	3
Single	4
Double	5
Currency	6
Date	7
String	8
Object	9
Error	10
Boolean	11
Variant	12
Data Object	13

ÖRNEK

Private Sub Command1_Click()

```
Dim ad
ad = "Oguz KUTLU"
FontSize = 20
ForeColor = RGB(12, 35, 142)
CurrentX = 800
CurrentY = 800
veritipi = VarType(ad)
Print veritipi

ad = 3.254
FontSize = 30
ForeColor = RGB(101, 13, 142)
CurrentX = 800
veritipi = VarType(ad)
Print veritipi
```

End Sub

MsgBox (Mesaj Kutusu)

Bu deyim özellikle kullanıcıya belli bir mesaj verme ve kullanıcıdan belli bir yanıt almak için kullanılır. MsgBox deyimi ile bir mesaj görüntülenir. Bu mesaj sadece mesajda olabilir veya üzerinde onay düğmeleri bulunabilir. Örneğin kullanıcının bir dosyayı kayıt etmesi gerektiğinde "Dosya Kaydedilsin mi?" gibi bir soru olabilir ve msgbox üzerinde de Yes No gibi yanıtlar alınabilir.

MsgBox ("mesaj", simgeler, "başlık")

ÖRNEK

1. MsgBox "Visual Basic Dersine Hoşgeldiniz"
2. MsgBox "Prestige Educations Center"

NOT

MsgBox deyimini Form Load yordamına yazılırsa form ekrana yüklendiği anda mesaj penceresi çıkar.

Windows uyumlu programlarda kullanıcıyı uyarmak veya mesaj vermek üzere ekrana getirilen mesaj kutularında Tamam düğmesinden başka düğmeler ve işaretler bulunur. Örneğin Excel ile çalışırken aktif belge kaydedilmeden kapatılmak istenirse, ekrana bir uyarı mesajı çıkar ve size seçme olanağı sunar. Bunun için iki seçenek vardır ve bilgisayarın sizin hangi tuşa bastığınızı anlaması için bir değişkene atamanız gerekmektedir.

ÖRNEK

Dim a

a = MsgBox("Visual Basic Dersine Hoşgeldiniz")

Görüldüğü gibi herhangi bir simge belirtilmezse TAMAM düğmesi belirir. Diğer düğmeleri çıkartmak için aşağıdaki kodlar kullanılır.

Simgesi	Değeri	Anlamı
vbOKOnly	0	Sadece OK düğmesini gösterir.
vbOKCancel	1	OK ve Cancel düğmesini gösterir.
vbAbortRetryIgnore	2	Durdur, Yeniden dene, Yoksay
vbYesNoCancel	3	Yes, No, Cancel düğmelerini gösterir.
vbYesNo	4	Yes ve No düğmelerini gösterir.
vbRetryCancel	5	Retry ve Cancel düğmelerini gösterir.

ÖRNEK

Dim a

a = MsgBox("Visual Basic Dersine Hoşgeldiniz", vbAbortRetryIgnore)

veya

a = MsgBox("Visual Basic Dersine Hoşgeldiniz", 2)

NOT

Mesaj kutusunun içine iki satır mesaj yazmak için Enter tuşunun karşılığına gelen ANSI 10 (Chr(10)) karakterini yazmamız gerekir.

ÖRNEK

Dim a

a = MsgBox("Visual Basic Dersine Hoşgeldiniz" & Chr(10) & "ikinci satıra geçtim", vbYesNo)

Mesaj pencerelerinde bir de uyarı veya dikkat çekici ek ikonlar mevcuttur. Bunlar da aşağıda gösterilmiştir.

Simgesi	Değeri	Anlamı
vbCritical	16	X işareti
vbQuestion	32	Soru işareti
vbExclamation	48	Ünlem
vbInformation	64	i enformasyon işareti

ÖRNEK

Dim a

a = MsgBox("Değişiklikler kaydedilsin mi?", vbYesNo + vbCritical)

Mesaj pencereleri üzerinde ki düğmelerin bir özelliği de aktif butonu seçme özelliğidir. Bunun için aşağıdaki kodlar yazılır.

Simge	Değeri	Anlamı
vbDefaultButton1	0	1.düğme aktif olacak
vbDefaultButton2	256	2.düğme aktif olacak
vbDefaultButton3	512	3.düğme aktif olacak
vbDefaultButton4	7682	4.düğme aktif olacak

ÖRNEK

```
Dim a
a = MsgBox("Degisiklikler kaydedilsin mi?", vbYesNo + vbCritical + vbDefaultButton2)
```

Son olarak mesaj penceresinin başlığını da değiştirebiliriz. Bunun için yazdığımız kodun en sonuna ekleriz.

ÖRNEK

```
Dim a
a = MsgBox("Degisiklikler kaydedilsin mi?", vbYesNo + vbCritical + vbDefaultButton2, "Kaydet Butonu")
```

Mesaj düğmesinde hangi düğmenin seçildiğine ilişkin bir değer döndürdüğünü belirlemek için aşağıdaki tablodaki değerler kullanılır.

Sabit Bilgi	Değeri	Anlamı
VbOK	1	OK düğmesi seçilmiş.
vbCancel	2	Cancel düğmesi seçilmiş.
vbAbort	3	Abort düğmesi seçilmiş.
vbRetry	4	Retry düğmesi seçilmiş.
vbIgnore	5	Ignore düğmesi seçilmiş.
vbYes	6	Yes düğmesi seçilmiş.
vbNo	7	No düğmesi seçilmiş.

ÖRNEK

```
Private Sub Command1_Click()
    Dim a, b, c
    a = MsgBox("Degisiklikler kaydedilsin mi?", vbYesNo + vbCritical + vbDefaultButton2, "kaydet butonu")
    c = 10
    If a = vbYes Then
        b = c * 10
    Else
        b = c * 2
    End If
    Print b
End Sub
```

Sonuç = yes butonuna basınca 100,
no butonuna basınca 20 olur.
veya,

```

Dim a, b, c
a = MsgBox("Degisiklikler kaydedilsin mi?", 4 + vbCritical + vbDefaultButton2, "kaydet butonu")
c = 10
If a = 6 Then
b = c * 10
Else
b = c * 2
End If
Print b

```

Sonuç = yes butonuna basınca 100,
no butonuna basınca 20 olur.

InputBox (Bilgi Giriş Kutusu)

Giriş kutusu kullanıcıdan bir değeri almayı sağlar. InputBox fonksiyonu bir iletişim kutusu görüntüleyerek kullanıcının bir değer girmesini sağlar. InputBox fonksiyonu kullanıcıdan aldığı değeri programa geri döndürür.

Değişken = InputBox ("mesaj", başlık, varsayım, xpoz, ypoz)

ÖRNEK 1

```

Private Sub Command1_Click()
Dim ad As String
ad = InputBox("Şehir Adını Giriniz : ?")
Print ad
End Sub

```

Görüldüğü gibi ekranın tam ortasında bir bilgi giriş ekranı çıktı ve sol üst köşesinde projenin adı yazıyor. Giriş kutusunun adını değiştirmek için aşağıdaki örnek yazılır.

ÖRNEK 2

```

Private Sub Command1_Click()
Dim ad As String
ad = InputBox("Şehir Adını Giriniz : ?", "Giriş Kutusu Başlığı")
Print ad
End Sub

```

Sol üst köşesinde bu sefer bizim yazdığımız "Giriş Kutusu Başlığı" yazıyor. Giriş kutusunun üzerinde bazen default olarak bir şey yazmasını isteriz, örneğin şehir adlarını giriyoruz ve en çok girilen şehir İstanbul olduğu için onun default olmasını isteyebiliriz. Bunun için aşağıdaki örneği yazarız.

ÖRNEK 3

```

Private Sub Command1_Click()
Dim ad As String
ad = InputBox("Şehir Adını Giriniz : ?", "Giriş Kutusu Başlığı", "İstanbul")
Print ad
End Sub

```

Eğer aksi bir durum belirtmediyse açılan InputBox kutusu ekranın tam ortasında açılır. Bazen açılan bu kutuları kullanıcıların ekranı daha iyi görebilmesi için değişik yerlere koyabiliriz. Bunun için aşağıdaki örneği yazarız.

ÖRNEK 4

```
Private Sub Command1_Click()
    Dim ad As String
    ad = InputBox("Şehir Adını Giriniz : ?", " Giriş Kutusu Başlığı", " İstanbul ", 3500, 500)
    Print ad
End Sub
```

ÖRNEK 5

İki tane arka arkaya sayı girilerek girilen iki sayının toplamı MsgBox ile bize bildirilsin. MsgBox penceresinde information da olsun.

```
Private Sub Command1_Click()
    Dim a As Integer
    Dim b As Integer
    Dim c As Integer
    Dim d As Integer
    a = InputBox("1. sayıyı giriniz")
    b = InputBox("2. sayıyı giriniz")
    c = a + b
    MsgBox("iki sayının toplamı = " & c & ",vbinformation)
End Sub
```

NOT : MesajBox yazımında kullanılan & (ampersant) işareti girilen değeri mesajbox kutusuna taşır.

Döngüler

If Then Else End if

İf deyimi belli bir ifadenin değerine göre bir deyimin işletilmesini sağlar. İf deyiminin basitten karmaşığa doğru değişik kalıpları vardır.

1. **if (koşul) Then (işlem)**
2. **if (koşul) Then**
 (işlem1)
 Else
 (işlem2)
 End if
3. **if (koşul1) Then**
 (işlem1)
 ElseIf (koşul2) Then
 (işlem2)
 Else
 (işlem-n)
 End if

ÖRNEK 1

İki sayı girin ve Ok basınca iki sayının çarpımını Cancel 'e basınca toplamını alsın ve sonucu MsgBox ile ekranda göstereyin.

```
Private Sub Command1_Click()
    Dim a As Integer, b As Integer
    Dim c As Integer
    Dim d As Integer
```

```

a = InputBox("1.sayıyı giriniz")
b = InputBox("2.sayıyı giriniz")
c = MsgBox("Sayılar toplansın mı? yoksa çarpılsın mı?", vbOKCancel)
    If c = 1 Then
        d = a * b
        MsgBox ("İki sayının çarpımı =" & d)
    End If
    If c = 2 Then
        d = a + b
        MsgBox ("İki sayının toplamı =" & d)
    End If
End Sub

```

ÖRNEK 2

Bir öğrencinin aldığı notu girelim ve aldığı not < 50 ise ekrana zayıf not aldı , >=50 ise ekrana geçer not aldı yazsın.

```

Private Sub Command1_Click()
Dim a As Integer
a = InputBox("Öğrencinin notunu giriniz")
    If a < 50 Then
        MsgBox ("Bu öğrenci zayıf not aldı")
    End If
    If a >= 50 Then
        MsgBox ("Bu öğrenci geçer not aldı")
    End If
End Sub

```

ÖRNEK 2.1

```

Private Sub Command1_Click()
Dim a As Integer

a = InputBox("Öğrencinin notunu giriniz")

    If a < 50 Then
        MsgBox ("Bu öğrenci zayıf not aldı")
    Else
        MsgBox ("Bu öğrenci geçer not aldı")
    End If

```

End Sub

ÖRNEK 3

Klavyeden girilen 3 sayının en büyük olanını bulmak.

```

Private Sub Command1_Click()
Dim a As Double
Dim b As Double
Dim c As Double

a = InputBox("birinci sayıyı giriniz")
b = InputBox("ikinci sayıyı giriniz")
c = InputBox("üçüncü sayıyı giriniz")

    If a > b Then

```



```

    If a > c Then MsgBox "En Büyük sayı birinci sayıdır."
    ElseIf c > b Then MsgBox "En Büyük sayı üçüncü sayıdır"
    Else
        MsgBox "En büyük sayı ikinci sayıdır."
    End If
End Sub

```

ÖRNEK 4

Elektrik faturası hesaplayan bir program 150 kw kadar 44.000 TL. 150 – 300 kw arası için 66.000 TL. 300 den sonrası için 75.000 TL. olsun.

```

Private Sub Command1_Click()
Dim tüketim As Double
Dim para As Double

tüketim = InputBox("Tüketim Değerini Girin")
    If tüketim > 0 And tüketim < 151 Then
        para = tüketim * 44000
    ElseIf tüketim > 150 And tüketim < 301 Then
        para = (150 * 44000) + ((tüketim - 30) * 66000)
    ElseIf tüketim > 301 Then
        para = (150 * 44000) + (150 * 66000) + ((tüketim - 300) * 75000)
    End If
    MsgBox ("ödeyeceğiniz para =" & para)
End Sub

```

ÖRNEK 5

3 bu örnek de 1.vize ,2. vize ve final notları girilecek ve puanı 50'den büyükse msgbox la geçtiniz değilse bütünleme notunu istet ve sonuc yine 50 den küçükse kalsın değilse geçsin. Vizenin %40 Finalin % 60

```

Dim a As Integer ,b As Integer ,c As Integer, d As Double, e As Double, f As Double, g As Integer, but
As Double
a:
a = InputBox("Birinci vize notu giriniz")
    If a < 0 Or a > 100 Then
        MsgBox ("Lütfen 0 - 100 arasında bir sayı giriniz")
        GoTo a
    Else
b:
b = InputBox("ikinci vize notu giriniz")
    If b < 0 Or b > 100 Then
        MsgBox ("Lütfen 0 - 100 arasında bir sayı giriniz")
        GoTo b
    Else
c:
c = InputBox("Final notunu giriniz")
    If c < 0 Or c > 100 Then
        MsgBox ("Lütfen 0 - 100 arasında bir sayı giriniz")
        GoTo c
    Else

```

```

        d = ((a + b) / 2) * 0.4
        e = (c * 0.6) + d
If e > 50 Then
MsgBox ("Geçtiniz")
FontBold = True
FontSize = 12
FontName = "tahoma"
ForeColor = RGB(15, 65, 85)
Print "Ortalama Notunuz =" & e
Else

```

' her if muhakkak end if le kapatılmalı iç içe çok sayıda if konabilir. if tek satırda bitirilebilir.

```

MsgBox ("Kaldınız")
f:
    f = InputBox("Bütünleme notunu giriniz")
    If f < 0 Or f > 100 Then
        MsgBox ("Lütfen 0 - 100 arasında bir sayı giriniz")
        GoTo f
    Else
        but = (f * 0.6) + d
        If but > 50 Then
            MsgBox ("geçtiniz")
        Else
            MsgBox ("artık sizi kimse kurtaramaz")
        End If
        Print "Son notunuz =" & but
    End If
End If
End If
End If
End If
End If

```

ÖRNEK 6

```
Private Sub Command1_Click()
```

'or ve and kullanma

```

Dim a As Integer
Dim b As Integer
Dim c As Integer
Dim d As Double
Dim e As Double
Dim f As Double
Dim but As Double

a = InputBox("Birinci vize notu giriniz")
b = InputBox("ikinci vize notu giriniz")
c = InputBox("Final notunu giriniz")

d = ((a + b) / 2) * 0.4
e = (c * 0.6) + d

If e < 50 Then
MsgBox ("Kaldınız")

    ElseIf e >= 50 And e < 65 Then

```

```
MsgBox ("Birinci gruptasınız")
```

```
ElseIf e >= 65 And e < 80 Then  
MsgBox ("ikinci gruptasınız")
```

```
ElseIf e >= 70 And e < 100 Then  
MsgBox ("Ben birtaneyim")
```

```
End If  
Print "Son notunuz  ="; e  
End Sub
```

Select Case

Kullanımı Kalıbı

```
'select case değişken
```

```
' case
```

```
' case
```

```
'end select
```

ÖRNEK 1

Klavyede 0 - 5 arasında bir rakam girilecek ve ne girildiğini msgbox la bize göstereceğiz

'birinci bölüm

```
Dim a As Integer
```

```
a = InputBox("1 - 5 arasında bir sayı giriniz")
```

```
Select Case a
```

```
Case 1
```

```
MsgBox ("girilen sayı 1'dir")
```

```
Case 2
```

```
MsgBox ("girilen sayı 2'dir")
```

```
Case 3
```

```
MsgBox ("girilen sayı 3'dür")
```

```
Case 4
```

```
MsgBox ("girilen sayı 4'dür")
```

```
Case 5
```

```
MsgBox ("girilen sayı 5'dir")
```

'ikinci bölüm

```
'mesala 8 rakamini girin
```

```
Case Else
```

```
MsgBox ("böyle bir sayı girilemez")
```

'üçüncü bölüm

'5-100 arasındaki tüm sayılar için

Case 5 To 100

'yazılır

End Select

ÖRNEK 2

Dim a As Integer ,b As Integer, c As Integer, d As Double, e As Double, f As Double, but As Double

a = InputBox("Birinci vize notu giriniz")

b = InputBox("ikinci vize notu giriniz")

c = InputBox("Final notunu giriniz")

d = ((a + b) / 2) * 0.4

e = (c * 0.6) + d

Select Case e

'Case e >= 0, e <= 100: (: case in bittiğini anlatır.)

'case e="ali",e="mehmet": (kullanılabilir.)

Case 0 To 49

MsgBox ("kaldınız")

Case 50 To 59

MsgBox ("geçer not aldınız")

Case 60 To 84

MsgBox ("sıralamaya girmek üzeresiniz")

Case 85 To 94

MsgBox ("az kaldı")

Case 95 To 100

MsgBox ("iste bu")

'case else kullanılabilir.

Case Is < 0 > 100

MsgBox ("girdiğin not yanlışdır lütfen kontrol ediniz.")

End Select

Print e

ÖRNEK 3

Private Sub Command13_Click()

Dim a

Dim b

Dim c

a = InputBox("bir harfe basınız")

Select Case a

Case "A" To "G"

c = "A ile G arasında bir harfe bastınız"

Case "H" To "K"

c = "H ile K arasında bir harfe bastınız"

Case "L" To "Z"

c = "L ile G arasında bir harfe bastınız"

Case Else

```
c = "yanlış bir tuşa bastınız"  
End Select
```

```
MsgBox c
```

```
End Sub
```

For Next

Bir grup deyimi belli sayıda çalıştırarak bir döngü oluşturur. Döngü bir kez işledikten sonra artırım değeri eklenir. Eğer herhangi bir artırım değeri kullanılmadıysa o zaman bir arttırılır. Sayaç değeri bitiş değerinden büyük olduğu zaman döngü sona erdirilir ve program Next deyiminden itibaren devam eder.

Döngü değişimleri amaç değişkenlerin değerlerini belli aralıklarla yaptırmak. Belli aralıklardaki sayıların toplamı , çıkartmak,çarpmak

Kalıp

```
FOR değişken=1 TO 70
```

```
,
```

```
,
```

```
,
```

```
,
```

```
NEXT
```

ÖRNEK 1

```
Private Sub Command1_Click()
```

'1 den 10 a kadar olan sayıları toplatalım

```
Dim a As Integer
```

```
Dim b As Integer
```

```
For a = 1 To 10
```

```
b = b + a 'buradaki yeni b= kavramı mat. = değil
```

```
Next
```

```
Print b
```

```
End Sub
```

ÖRNEK 2

```
Private Sub Command1_Click()
```

'1 den 10 a kadar olan sayıları toplatalım print e dikkat

```
Dim a As Integer
```

```
Dim b As Integer
```

```
For a = 1 To 10
```

```
b = b + a 'buradaki yeni b= kavramı mat. = değil
```

```
Print b
```

Next

' olayın farkını anlat

End Sub

ÖRNEK 3

Private Sub Command1_Click()

'klavyeden bir sayı girilecek ve
'makina o sayıya kadar olan sayıların toplamını alacak

Dim a As Double
Dim b As Double
Dim c As Double

a = InputBox("birinci sayıyı girin")

For b = 1 To a
c = c + b

Next

Print ("1 den ") & a & ("ya kadar olan sayıların toplamı"), c

End Sub

ÖRNEK 4

Private Sub Command1_Click()

' Belli bir sayıya kadar toplama yapmak
' Bir atlatacak toplatmak için
' step komutu kullanılarak

Dim a As Double
Dim b As Double
Dim c As Double

a = InputBox("birinci sayıyı girin")

For b = 1 To a Step 2

c = c + b

Next

Print c

End Sub

ÖRNEK 5

Private Sub Command11_Click()

' 1 den girilen bir sayıya kadar olan sayıların çarpımı
' en fazla 170 ! e kadar yapabilirsiniz.

Dim a As Double, b As Double, c As Double

```

a = InputBox("birinci sayıyı girin")
c = 1
  For b = 1 To a
    c = c * b
  Next
Print a; "! ="; c
End Sub

```

ÖRNEK 6

```

Private Sub Command12_Click()

' üç boyutlu bir yazı yazılması

Dim t, i, x1, y1, x, y
x = 200
y = 200
t = "Oguz Kemalettin KUTLU"
FontName = "tahoma"
FontSize = 30

  For i = 0 To 15
    x1 = x + i * 5 ' 10 birim sağa kayar
    y1 = y + i * 10 ' 10 birim aşağıya kayar
    ForeColor = QBColor(i) 'QBColor (0 - 15 ) ARASI ÖZEL RENKLER
                        'forecolor=rgb(red,green,blue)
                        ' qbcolor kullanılacaksa eğer to 16 olamaz

    CurrentX = x1
    CurrentY = y1
  Print t
  Next ' döngüyü anlatmayı unutma
End Sub

```

ÖRNEK 7

```

'Private Sub Form_Resize() ' formun boyutu değişse bile yazı ona uygun hale gelir.
'Cls ' önce ekranın temizlnmesi lazım
'Show
'Dim t, i
't = "Oguz KUTLU – Visual Basic Öğretmeni"
'FontName = "tahoma"
'For i = 1 To 10
'FontSize = i * 5
'CurrentX = (ScaleWidth - TextWidth(t)) / 2 ' yazıyı tam merkezden başlatmak için yazıyı 10 kere
büyütterek aşağıya doğru yazarak iner
'Print t
'Next
'End Sub

```

QBColor ()

0	Siyah	8	Gri
1	Mavi	9	Açık Mavi
2	Yeşil	10	Açık Yeşil
3	Cyan	11	Açık Cyan
4	Kırmızı	12	Açık Kırmızı
5	Eflatun	13	Açık Eflatun
6	Sarı	14	Açık Sarı
7	Beyaz	15	Açık Beyaz

RGB ()

Renk	Kırmızı	Yeşil	Mavi
Siyah	0	0	0
Mavi	0	0	255
Yeşil	0	255	0
Cyan	0	255	255
Kırmızı	255	0	0
Eflatun	255	0	255
Sarı	255	255	0
Beyaz	255	255	255

Disk ve Dosya Komutları

Program içerisinde kullanacağımız bazı kodlar sayesinde çalışma anında geçerli klasör veya dizinin adını öğrenebilir, başka bir dizine geçebilir, istediğiniz dizini veya dosyayı silebilir, yeni bir dizin oluşturabiliriz.

CurDir

Aktif dizini gösterir.

```
Private Sub Command1_Click()
    MsgBox CurDir
End Sub
```

ChDir

Aktif dizini değiştirir.

```
Private Sub Command1_Click()
    ChDir "c:\\"
    MsgBox CurDir
End Sub
```

ChDrive

Geçerli sürücüyü değiştirir.

```
Private Sub Command1_Click()
    ChDrive "c"
    ChDir "c:\program files"
    MsgBox CurDir
End Sub
```

MkDir

Herhangi bir sürücüde dizin oluşturmak için kullanılır.

```
Private Sub Command1_Click()
```



```
MkDir "c:\deneme klasör"  
End Sub
```

Rmdir

Herhangi bir sürücüde ki içi boş olan bir dizini silmek için kullanılır.

```
Private Sub Command1_Click()
```

```
    Rmdir "c:\deneme klasör"  
    'Dikkat edileceği gibi silinen klasör çöp kutusuna gitmedi. 'eğer dizinin içinde bir dosya kaldıysa  
    yine hata verir. Hatayı önlemek için
```

```
    On Local Error GoTo hata  
        Rmdir "c:\deneme klasör"  
        Exit Sub  
hata:  
    MsgBox ("Dizin silinemedi " & Error)  
End Sub
```

FileCopy

Bir klasörden başka bir klasöre kopyalamak için kullanılır.

'öncelikle c:\belgelerim 'e bir tane kopya.txt dosyası jazırlanır.

```
Private Sub Command4_Click()  
    FileCopy "c:\belgelerim\kopya.txt", "c:\deneme klasör\kopya.txt" : End Sub
```

FileLen

Bir dosyanın byte olarak büyüklüğünü verir.

```
Private Sub Command5_Click()  
    MsgBox (FileLen("c:\deneme klasör\kopya.txt"))  
    'ikinci olarak dosyaya bir şeyler yazıp tekrar deneyin  
End Sub
```

FileDateTime

Bir dosyanın en son değiştirildiği tarihi öğrenmek için kullanılır.

```
Private Sub Command6_Click()  
    MsgBox (FileDateTime("c:\deneme klasör\kopya.txt"))  
End Sub
```

GetAttr

Bir dosyanın özelliklerini öğrenmek için kullanılır. Bu komut normalde geriye bir rakam döndürür.

Örnek 1

```
Private Sub Command7_Click()  
    MsgBox (GetAttr("c:\deneme klasör\kopya.txt"))  
End Sub
```

VbNormal	0	Özelliği olmayan dosya
vbReadOnl	1	Salt okunur dosya

y		
VbHidden	2	Gizli dosya
VbSystem	4	Sistem dosyası
VbDirectory	16	Klasör
VbArchive	32	Arşiv dosyası

Örnek 2

```
Private Sub Command8_Click()
Cls
Dim dosya
dosya = GetAttr("c:\deneme klasör\kopya.txt")
Print "kopya.txt"
If dosya And 0 Then Print "Normal Dosya"
If dosya And 1 Then Print "Salt Okunur"
If dosya And 2 Then Print "Gizli Dosya"
If dosya And 4 Then Print "Sistem Dosyası"
If dosya And 16 Then Print "Klasör"
If dosya And 32 Then Print "Arşiv Dosyası"
End Sub
```

SetAttr

Bir dosyanın özelliklerini değiştirmek için kullanılır.

Örnek 1

```
Private Sub Command9_Click()
SetAttr "c:\deneme klasör\kopya.txt", 1 + 2 + 4 + 32
End Sub
```

Örnek 2

```
Private Sub Command9_Click()
SetAttr "c:\deneme klasör\kopya.txt", vbNormal + vbReadOnly + vbHidden + vbSystem +
vbArchive
End Sub
```

Name

İstediğimiz bir klasörün veya bir dosyanın adını değiştirmek için kullanılır.

Örnek 1

```
Private Sub Command11_Click()
Name "c:\belgelerim\kopya.txt" As "c:\deneme klasör\kopya2.txt"
End Sub
```

NOT: Program ikinci kez çalıştırıldığında dikkat edileceği gibi hata komutu verir. Bunu önlemek için,

Örnek 2

```
Private Sub Command12_Click()
On Error GoTo k
Name "c:\belgelerim\kopya.txt" As "c:\deneme klasör\kopya2.txt"
Exit Sub
```

```
k:
MsgBox ("Böyle bir dosya bulunamadı")
End Sub
```

Dir

İstediğimiz bir klasördeki dosyaların listesini göstermek için kullanılır.

Örnek 1

```
Private Sub Command10_Click()
Cls
Dim dosya
dosya = Dir("c:\*.*.*)
Print dosya: End Sub
```

Örnek 2

```
Private Sub Command10_Click()
Cls
Dim dosya
dosya = Dir("c:\*.*.*)
While dosya <> "" ` Başka dosya kalmayıncaya kadar devam et.
Print dosya
dosya = Dir
Wend

End Sub
```

Kill

İstediğimiz bir klasörün veya bir dosyanın silinmesi için kullanılır.
Bu işlemi yaparken çok dikkatli olmamız lazım çünkü silinen dosya çöp kutusuna gitmez.

Örnek 1

```
Private Sub Command13_Click()
Kill "c:\deneme klasör\kopya.txt"
End Sub
```

Örnek 2

```
Private Sub Command14_Click()
On Error GoTo k
Kill "c:\deneme klasör\kopya.txt"
Exit Sub
k:
MsgBox ("Böyle bir dosya bulunamadı")
End Sub
```

Örnek 3

```
Private Sub Command15_Click()
On Error GoTo k
Dim c
c = MsgBox("dosya silinecek mi ?", vbYesNo)
If c = vbYes Then
Kill "c:\deneme klasör\kopya.txt"
MsgBox ("dosya silindi")
Exit Sub
```

```
k:
    MsgBox ("Böyle bir dosya bulunamadı")
End If
End Sub
```

End

Bir formu sonlandırmak için kullanılır.

```
Private Sub Command17_Click()
    End
End Sub
```

Çizim Yapmak

Bir form üzerine çizim yapmak için daha önce belirttiğimiz gibi ScaleWidth(X) ve ScaleHeight(Y) koordinatlarından ve twip ölçü birimlerinden yararlanacağız. Form üzerinde çizeceğimiz bir noktanın koordinatını belirtirken formun üst sol köşesi (0,0) dır ve X değeri sağa doğru Y değeri ise aşağıya doğru artar. Formun üstündeki mavi bar bu değerler dışındadır.

Line

İstenilen koordinatlar arasında çizgi çizmek veya kutu çizmek için kullanılır.

Line(X_{başlangıç}, Y_{başlangıç})-(X_{bitis}, Y_{bitis}), çizgi rengi ,BF

BF = Otomatik kutu yapmak için kullanılır.

Örnek 1

```
Private Sub Command1_Click()
    Cls
    Line (500, 500)-(2500, 2500)
End Sub
```

Örnek 2

```
Private Sub Command2_Click()
    Cls
    Line (500, 500)-(2500, 2500), RGB(200, 25, 12)
End Sub
```

Örnek 3

```
Private Sub Command3_Click()
    Cls
    Line (500, 500)-(2500, 2500), QBColor(5)
End Sub
```

Örnek 4

```
Private Sub Command4_Click()
    Line (500, 500)-(2500, 2500), RGB(200, 25, 12), B
End Sub
```

Örnek 5

```
Private Sub Command4_Click()
```

```
Line (500, 500)-(2500, 2500), RGB(200, 25, 12), BF
End Sub
```

Örnek 6

Klavyeden girilen rakamlarla bir kutu çizmek.

```
Private Sub Command5_Click()
    Dim a, b, c, d, e, f, g
    a = InputBox("xbas")
    b = InputBox("ybas")
    c = InputBox("xbit")
    d = InputBox("ybit")
    e = InputBox("R rengi için 255 e kadar bir sayı gir")
    f = InputBox("G rengi için 255 e kadar bir sayı gir")
    g = InputBox("B rengi için 255 e kadar bir sayı gir")
    Line (a, b)-(c, d), RGB(e, f, g), BF
End Sub
```

Örnek 7

BF kodunu kullanmadan bir kutu çizmek.

```
Private Sub Command6_Click()
    Cls
    Line (500, 500)-(500, 2000), RGB(250, 12, 35)
    Line (2000, 500)-(2000, 2000), RGB(250, 12, 135)
    Line (500, 2000)-(2000, 2000), RGB(25, 12, 235)
    Line (500, 500)-(2000, 500), RGB(35, 112, 35)
End Sub
```

NOT= DrawWidth özelliğini anlat

Örnek 8

```
Private Sub Form_Resize()
    Show
    Cls
    Dim i, stp
    Form1.Width = Form1.Height
    stp = Form1.ScaleWidth / 20
    For i = 0 To Form1.ScaleHeight / 2 Step stp
        ForeColor = i * 12
        Line (i, i)-(Form1.ScaleWidth - i, Form1.ScaleHeight - i), , BF
    Next
End Sub
```

Örnek 9

Bir üçgen çizmek için

```
Private Sub Command2_Click()
    Cls
    DrawWidth = 3
    Line (500, 500)-(2000, 2000)
    Line -(500, 2000)
    Line -(500, 500)
End Sub
```

End Sub

Circle

İstenilen koordinatlar arasında daire çizmek için kullanılır.

Circle($X_{\text{başlangıç}}, Y_{\text{başlangıç}}$), Yarı çap, çizgi rengi

Örnek 1

```
Private Sub Command1_Click()  
    DrawWidth = 3  
    Circle (1000, 1000), 1000  
End Sub
```

Örnek 2

```
Private Sub Command1_Click()  
    DrawWidth = 3  
    Circle (1000, 1000), 1000, RGB(125,32,14)  
End Sub
```

Örnek 3

```
Private Sub Command1_Click()  
    DrawWidth = 3  
    Circle (1000, 1000), 1000, QBColor(2)  
End Sub
```

Örnek 4

```
Private Sub Command2_Click()  
Cls  
    Dim i, j  
    For j = 1 To 15  
        For i = 750 To 3550 Step 200  
            Circle (i + 2000, 2500), Qbcolor(j)  
        Next  
    Next  
End Sub
```

Örnek 5

```
Private Sub Command15_Click()  
Dim a, b, xort, yort  
DrawWidth = 3  
xort=scalewidth/2  
yort=scaleheight/2  
    b = 500  
    For a = 0 To 15  
        Circle (xort,yort), b, QBColor(a) '16 ana renk için  
        b = b + 1000  
    Next  
End Sub
```

Örnek 6

```

Private Sub Form_Load()
Show
Cls
Dim x, y
DrawWidth = 2
    x = ScaleWidth / 2
    y = ScaleHeight / 2
Circle (x, y), x / 2
End Sub

```

'Bu örneği Form_Resize yordamına yazarsak formun boyutu değiştiği anda oda değişir.

Örnek 7

Aslında Circle deyiminin alacağı iki üç parametre daha vardır. 4. parametredeki rakam 0 olursa tam çember 6 olursa da çemberin bir kısmı çizilir. 4. parametre 0-6 arasında değer alabilir.

```

Private Sub Command1_Click()
    DrawWidth = 4
    Circle (2500, 2500), 1000, RGB(255, 0, 0), 1 'NOT=rakam değiştirilerek denemeler yapılır.
End Sub

```

Örnek 8

5. parametredeki rakam 6.27 olursa tam çember 0-6.27 arasında rakam alırsa 5.parametrenin tersi işlem yapar.

```

Private Sub Command1_Click()
    DrawWidth = 4
    Circle (2500, 2500), 1000, RGB(255, 0, 0), 0, 6.27
End Sub

```

Örnek 9

7.parametredeki rakam 0-1 arasında olursa çember enine basıklaşır. 1 den büyük rakam verirse bu sefer dikine basıklaşır.

```

Private Sub Command1_Click()
    DrawWidth = 4
    Circle (2500, 2500), 1000, RGB(255, 0, 0), 0, , 0.4 'NOT = denemeler yap
End Sub

```

Örnek 10

```

Private Sub Form_Resize()
    Cls
    Dim x As Integer
    Dim y As Integer
    DrawWidth = 3
    x = ScaleWidth / 2
    y = ScaleHeight / 2
    Circle (x, y), x / 2, RGB(0, 0, 255)
    Line (x - x / 2,y)-(x + x / 2,y)
    Line (x, y+ x / 2)-(x, y- x / 2)
End Sub

```

PSet

Formun herhangi bir noktasına nokta koymak için kullanılır.

PSet ($X_{\text{başlangıç}}, Y_{\text{başlangıç}}$), çizgi rengi

Örnek 1

PSet (500, 500), RGB(35,24,111)

Örnek 2

```
Private Sub Command1_Click()  
Dim i  
DrawWidth = 4  
For i = 500 To 2500 Step 10  
PSet (i, i), QBColor(4)  
Next  
End Sub
```

Örnek 3

```
Private Sub Command1_Click()  
Cls  
Dim a, b, r, t, x, y  
For t = -100 To 100 Step 0.01  
a = ScaleWidth / 2  
b = ScaleHeight / 2  
r = Sin(6 * t) * Cos(6 * t)  
x = r * Cos(t)  
y = r * Sin(t)  
PSet (a + x * 3800, b - y * 3800)  
Next  
End Sub
```

Örnek 4

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)  
If Button = 1 Then  
ForeColor = RGB(255, 0, 0)  
DrawWidth = 3  
PSet (x, y)  
End If  
End Sub
```

Random Dosyalar

- '1. 6 tane label çiz
- '2. 6 tane text çiz
- '3. adlarını yaz. Kayıt No, Adınız, Soyadınız, Telefon, Adres, Şehir
- '3.1 Modül ekle
- '4. Kaydet butonu ekle
- '5. Önceki kayıt
- '6. Sonraki kayıt
- '7. İlk kayıt
- '8. Son kayıt
- '9. Kayıt bul
- '10. Degistir
- '11. Kayıt ekle
- '12. Sil

'13. Toplam ve bulunan kayıt sayısı sırası yine command sırasıyla

'14. Çıkış

Private Sub Form_Load()

Open "oguz.dat" For Random As #1 Len = Len(alanlar) ` form yüklendiği anda bir dosya açmak için

kullanılır. oguz.dat dosya isim

kayitno = LOF(1) / Len(alanlar)

kayitno = kayitno + 1

topkayit.Caption = kayitno - 1

suankayit.Caption = kayitno

kayitno.Enabled = False

End Sub

'For Random dosyanın random dosya olduğunu gösterir. **As 1** dosya numarası 255 e kadar verilebilir.

'başlangıçta verdiğimiz dosya numarası sonra kullanılacak. **Len (alanlar)** alanların uzunluğunu len değişkenine

`atamamız gerekmektedir.

'len alanlar toplam uzunluğu 'lof ise 1 numaraları dosyanın toplam uzunluğu 'toplam uzunluğu alanlar

`uzunluğuna bölersek kayıt no yu gösterir.

'kayitno=kayitno+1 ilk anda kayıt boş olduğu zaman hata vermemek için

'ilk çalıştırdığımızda oguz.dat bulamayacağı için ilk çalıştırma anında kendisi oluşturur.

'Random dosyaların bir özelliği de her kayda aynı oranda uzunluk ayırır.

'örneğin 5 dosya var ve hepsine 10 uzunluk ayırdığını düşünelim

'50 / 10 = 5 çıkar ama +1 yapar ve 6 dosyayı bulur.

'artık aktif dizinde oguz.dat dosyası bulunur.

' Değişkenlerin tüm procedür içinde geçerli olması için modülde bir tip tanımlaması yapmak zorundayız.

' **Type** baslar ve **End Type** biter. Type bilgileri Text Kutuları için yapılır. Oluşturulacak text'leri makineye tanıttık.

General _Declarations

Option Explicit

Type kayitbilgisi

ad As String * 10

soyad As String * 15

telefon As String * 10 'tel. numaralarının (216)4755054 göstermek için maskedit component inde anlatacağız.

adres As String * 30

sehir As String * 10

End Type

Global alanlar As kayitbilgisi

' kayitbilgisi tipini tüm modül ve formlara tanıtmak için kullanılır. **alanlar.** koyduğumuz anda tanımladığımız

type'ler otomatik olarak ekrana gelir modulden global yazısı kaldırılınca ekrana gelmez.

'text kutularına yazdığımız yazılar dikkat ederseniz programdan çıkılıp tekrar girildiği zaman yazdığımız yazıların kaydedilmediğini görürüz. Kaydetmek için bir kaydet butonu hazırlar ve içerisine aşağıdaki kod yazılır.

Private Sub kaydet_Click()

Dim c As Integer

```
alanlar.ad = form1.ad
alanlar.soyad = form1.soyad
alanlar.telefon = form1.telefon
alanlar.adres = form1.adres
alanlar.sehir = form1.sehir
```

'dikkat edilirse form1 den sonra ad,şehir gelmesinin kolaylığı görülmüş oldu

```
    c = MsgBox("Degisiklikler kayddilsin mi ?", vbYesNo)
    If c = vbYes Then
        'yes e basıldığı zaman verilerin dosyaya yazılmasını istiyorum bunun için PUT değişimi
        kullanılır.
```

Put #1, kayitno, alanlar

'bunun anlamı yeni kaydı alanlar içine kayitno ya göre kaydetmeyi sağlar 1 rakamı ilk başta verdiğimiz dosya

noyla aynı olması lazım bu işlemden sonra text kutularını boşaltıp kayitno yu arttırıp ekrana getirsin.

```
    form1.ad = " "
    form1.soyad = " "
    form1.telefon = " "
    form1.adres = " "
    form1.sehir = " "
```

'text kutuları boşaldıktan sonra yeni kayda geçince kayitno artması gerekir

```
kayitno = kayitno + 1
    ad.SetFocus 'imlecin ad dan başlaması için kullanılır.
    End If
    topkayit.Caption = LOF(1) / Len(alanlar)
    suankayit.Caption = kayitno
```

End Sub

Private Sub onceki_Click()

```
If kayitno= 1 Then
    MsgBox ("Zaten İlk Kayıttasınız")
Else
    Kayitno=kayitno-1
    Get #1, kayitno, alanlar
        form1.ad = alanlar.ad
        form1.soyad = alanlar.soyad
        form1.telefon = alanlar.telefon
        form1.adres = alanlar.adres
        form1.sehir = alanlar.sehir
```

End if

Form1.ad.SetFocus

End Sub

Private Sub sonraki_Click()

```
Dim a as integer
a=LOF(1) / Len (alanlar)
If kayitno>= a Then
    MsgBox ("Zaten Son Kayıttasınız")
Else
    Kayitno=kayitno+1
    Get #1, kayitno, alanlar
        form1.ad = alanlar.ad
        form1.soyad = alanlar.soyad
```

```

    form1.telefon = alanlar.telefon
    form1.adres = alanlar.adres
    form1.sehir = alanlar.sehir
End if
Form1.ad.SetFocus
End Sub
Private Sub bul_Click()
Dim bul As Integer
bul = InputBox("Bulmak istediğiniz Kayıt No yu giriniz")
kayitno = LOF(1) / Len(alanlar)
If bul <= kayitno Then
kayitno = bul
    Get #1, kayitno, alanlar 'dosya ya çağırdığım bilgileri ekrana yazar
    form1.ad = alanlar.ad
    form1.soyad = alanlar.soyad
    form1.telefon = alanlar.telefon
    form1.adres = alanlar.adres
    form1.sehir = alanlar.sehir
Else
MsgBox ("Böyle bir kayıt yok")
End If
End Sub

Private Sub kayıtekle_Click()
kayitno = LOF(1) / Len(alanlar)
kayitno = kayitno + 1
    form1.ad = " "
    form1.soyad = " "
    form1.telefon = " "
    form1.adres = " "
    form1.sehir = " "
    form1.ad.SetFocus
End Sub
Private Sub iptal_Click()
Dim c
    c = MsgBox("İptal etmek istiyormusunuz?", vbYesNo, "İptal Penceresi")
    If c = vbYes Then
        form1.ad = " "
        form1.soyad = " "
        form1.telefon = " "
        form1.adres = " "
        form1.sehir = " "
    End If
    ad.SetFocus
End Sub

Private Sub degistir_Click()
Dim c As Integer

If form1.ad <> alanlar.ad Or form1.soyad <> alanlar.soyad Or form1.telefon <>
alanlar.telefon Or form1.adres <> alanlar.adres Or form1.sehir <> alanlar.sehir Then

c = MsgBox("Değişiklikler kaydedilsin mi ?", vbYesNo, "Değişiklik Penceresi")
If c = vbYes Then

    alanlar.ad = form1.ad
    alanlar.soyad = form1.soyad
    alanlar.telefon = form1.telefon

```

```
alanlar.adres = form1.adres
alanlar.sehir = form1.sehir
Put #1, kayitno, alanlar
```

```
End If
End If
```

```
ad.SetFocus
End Sub
```

```
Private Sub sil_Click()
```

```
Dim silsayi, i, c
```

```
    silsayi = LOF(1) / Len(alanlar)
    For i = kayitno To silsayi - 1
        Get #1, i + 1, alanlar
        Put #1, i, alanlar
    Next i
```

```
    Open "oguz.tmp" For Random As #2 Len = Len(alanlar)
    For i = 1 To silsayi - 1
        Get #1, i, alanlar
        Put #2, i, alanlar
    Next i
    Close #1
    Close #2
```

```
FileCopy "oguz.tmp", "oguz.dat"
```

```
Kill "oguz.tmp"
```

```
c = MsgBox("Dosya silindi.")
```

```
Unload form1
```

```
Load form1
```

```
form1.Show
```

```
End Sub
```

Fonksiyonlar

Fonksiyonlar belli işlemleri yaparlar (örneğin bir sayının karakökünü almak yada verilen bir tarih bilgisini yıl cinsine çevirmek gibi) ve bir değeri döndürürler. Bu nedenle genellikle bir değere eşitlenerek kullanılır.

Abs Fonksiyonu

Bir sayının mutlak değerini bulmak için kullanılır.

Örnek

```
Private Sub Command1_Click()
Dim a As Double
Dim b As Double
    a = InputBox("Bir sayı girin")
    b = Abs(a)
Print "Girdiğiniz sayının mutlak değeri = "; b
End Sub
```

Asc Fonksiyonu

Herhangi bir string bilginin ilk karakterine karşılık gelen ASCII kodunun değerini verir.

Örnek

```
Private Sub Command1_Click()  
    Dim say  
    say = Asc("A")  
    Print say  
    say = Asc("a")  
    Print say  
    say = Asc("oguz")  
    Print say  
End Sub
```

Chr Fonksiyonu

Asc fonksiyonunun tersidir yani ASCII kodunun karakter olarak karşılığını verir.

Örnek

```
Private Sub Command1_Click()  
    Dim say  
    say = InputBox("Lütfen bir rakam giriniz")  
    Print Chr(say)  
End Sub
```

Date Fonksiyonu

Sistem tarihini gösterir.

Örnek

```
Private Sub Command1_Click()  
    MsgBox "Bu Günün Tarihi :" & Date, vbOKOnly, "Tarih Bilgisi"  
End Sub
```

DateDiff Fonksiyonu

İki tarih arasındaki sürede geçen zamanı belirtir.

Örnek

```
Private Sub Command1_Click()  
    Dim tarih As Date  
    tarih = InputBox("Doğum tarihinizi giriniz")  
    MsgBox ("siz" & DateDiff("d", Now, tarih) & "gün önce doğdunuz")  
End Sub
```

Int Fonksiyonu

Girilen sayıyı yuvarlayarak yazılmasını sağlar.

Örnek 1

```
Private Sub Command1_Click()
```

```

Dim a, b, c
a = InputBox("bir sayı giriniz")
b = InputBox("bir rakam giriniz")
c = Int(a / b)
Print c
c = a / b
Print c
End Sub

```

LCase Fonksiyonu

Yazdığınız bir yazının bütün harflerini küçük harfe çevirir.

Örnek 1

```

Private Sub Command1_Click()
    Dim a, b
    a = "OGUZ KUTLU"
    b = LCase(a)
    Print b
End Sub

```

Örnek 2

```

Private Sub Command1_Click()
    Dim a, b
    a = "OGUZ"
    b = InputBox("şifreyi giriniz")
    If b = a Then
        Print "şifreyi bildiniz"
    Else
        Print "şifreyi bilemediniz"
    End If
End Sub

```

Örnek 2.1

```

Private Sub Command1_Click()
    Dim a, b
    a = "OGUZ"
    b = InputBox("şifreyi giriniz")
    If LCase(b) = LCase(a) Then
        Print "şifreyi bildiniz"
    Else
        Print "şifreyi bilemediniz"
    End If
End Sub

```

Left Fonksiyonu

Soldan istediğimiz kadar karakteri bulmamızı sağlar.

Örnek 1

```

Private Sub Command1_Click()
    Cls
    Dim ad, b
    ad = InputBox("adınızı giriniz")
    b = Left(ad, 4)

```

```
Print b
End Sub
```

Örnek 2

```
Private Sub Command2_Click()
    Cls
    Dim ad, b, c
    ad = InputBox("adınızı giriniz")
    b = InputBox("kaç karakter")
    c = Left(ad, b)
    Print c
End Sub
```

Örnek 3

```
Private Sub Command1_Click()
    Dim a, b
    Cls
    b = InputBox("adınızı giriniz")
    For a = 1 To Len(b)
        Print Left(b, a)
    Next
End Sub
```

Len Fonksiyonu

Bir bilginin karakter olarak uzunluğunu veya değişken tipinin kapladığı byte alanını verir.

Örnek 1

```
Private Sub Command1_Click()
    Dim ad, uz
    ad = "Oguz KUTLU"
    uz = Len(ad)
    Print uz
End Sub
```

'SONUÇ=10 olur.

Örnek 2

```
Private Sub Command1_Click()
    Dim ad
    Dim a As Integer
    Dim b As Long
    a = 123
    b = 1234567890
    ad = "Oguz KUTLU"
    Print "Tamsayı"; Len(a)
    Print "Uzun tamsayı"; Len(b)
    Print "Karakter"; Len(ad)
End Sub
```

Örnek 3

```
Private Sub Command1_Click()
    Dim ad
    Dim a As Integer
```

```

Dim b As Long
a = InputBox("birinci sayıyı giriniz")
b = InputBox("ikinci sayıyı giriniz")
ad = InputBox("adınızı giriniz")
Print "tamsayı"; Len(a)
Print "uzun tamsayı"; Len(b)
Print "sozluk" & Len(ad); "karakterdir"
End Sub

```

Ltrim, Rtrim ve Trim Fonksiyonu

Bir metnin içindeki boşlukları çıkartarak bitişik metin oluşturur. LTrim bilginin sol tarafındaki boşlukları atar. Rtrim bilginin sağ tarafındaki boşlukları atar. Trim bilginin sol ve sağ tarafındaki boşlukları atar.

Örnek 1

```

Private Sub Command1_Click()
    Dim ad, a, b, c
    ad = " Oguz Kutlu "
    a = LTrim(ad)
    b = RTrim(ad)
    c = Trim(ad)
    Print a
    Print b
    Print c
End Sub

```

Mid Fonksiyonu

Bir metnin içindeki istenilen kısımdan kelime almak için kullanılır.

Mid(değişken,başlangıç,uzunluk)

Örnek 1

```

Private Sub Command1_Click()
    Dim ad
    ad = "Oguz Kemalettin Kutlu"
    FontName = "tahoma"
    FontSize = 15
    Print Mid(ad, 6)
    Print Mid(ad, 1, 23)
    Print Mid(ad, 17)
    Print Mid(ad, 1, 9)
End Sub

```

Örnek 2

```

Private Sub Command1_Click()
    Dim ad As String
    Dim a As Integer
    Dim b As Integer
    ad = "Oguz Kemalettin Kutlu"
    FontName = "tahoma"
    FontSize = 15
    a = InputBox("başlangıç değeri")
    b = InputBox("uzunluğu giriniz")
    Print Mid(ad, a, b)

```


End Sub

Now Fonksiyonu

Sistem tarih ve zamanını verir.

Örnek 1

```
Private Sub Command1_Click()  
    MsgBox "Bu Günün Bilgisi :" & Now, vbOKOnly, "Tarih Bilgisi"  
End Sub
```

Round Fonksiyonu

Belli sayıda ondalık basamağa sahip yuvarlanmış sayı oluşturur.

Round (değişken,ondalık kısım)

Sayı = 123,456789
Yenisayı=Round(sayı,3) ` Sonuç = 123,456

Space Fonksiyonu

İstenilen sayıda boşluk oluşturur.

Text2.Text = "Oğuz" & Space(3) & "KUTLU"

Shell Fonksiyonu

Hazırladığımız program içerisinde başka bir programı çalıştırmak için kullanılır.

Shell(yol[windowtipi])

VbHide	0	Pencere gizli olarak aktifleşir.
vbNormalFocus	1	Pencere normal boyutlarında açılır.
vbMinimizedFocus	2	Pencere simge olarak açılır.
vbMaximizedFocus	3	Pencere tüm ekranı kaplar.
vbNormalNoFocus	4	Pencere geri yüklenir. Focus başka komponentte
vbMinimizedNoFocus	6	Pencere simge olarak yüklenir. Focus başka komponentte.

Örnek 1

```
Private Sub Command1_Click()  
    Dim a  
    a = Shell("c:\program files\microsoft office\excel",3)  
End Sub
```

Sqr Fonksiyonu

Bir sayının karakökünü alır.

Örnek 1

```
Private Sub Command1_Click()
```

```

Dim a As Double
Dim b As Double
Dim c As Double
Dim d As Double
c = InputBox("bir sayı giriniz")
If c < 0 Then
MsgBox ("yanlış sayı girdiniz")
Else
d = Sqr(c)
Print d
End If
End Sub

```

String Fonksiyonu

İstenilen karakterin ilk harfini yanyana istediğimiz adtte yazmamızı sağlar.

Örnek 1

```

Private Sub Command1_Click()
Dim a
Dim b
b = InputBox("bir karakter giriniz.")
FontSize = 10
Print String(10, Chr(65))
Print String(20, "_")
Print String(5, b)
End Sub

```

Time Fonksiyonu

Sistem zamanını gösterir.

Örnek 1

```

Private Sub Command1_Click()
Dim zaman
MsgBox ("sistem tarihi :") & Time
End Sub

```

UCase Fonksiyonu

Bir ifadeyi büyük harfe çevirir.

Örnek 1

```

Private Sub Command1_Click()
Dim a, b
a = "oguz kutlu"
b = UCase(a)
Print b
End Sub

```

Örnek 2

```

Private Sub Command1_Click()
Dim a, b
a = "oguz"

```

```

        b = InputBox("şifreyi giriniz")
    If b = a Then
        Print "şifreyi bildiniz"
    Else
        Print "şifreyi bilemediniz"
    End If
End Sub

```

Örnek 2.1

```

Private Sub Command1_Click()
    Dim a, b
    a = "oguz"
    b = InputBox("şifreyi giriniz")
    If UCase(b) = UCase(a) Then
        Print "şifreyi bildiniz"
    Else
        Print "şifreyi bilemediniz"
    End If
End Sub

```

Val Fonksiyonu

Bir ifadenin içindeki ilk rakam veya rakamları alır.

Örnek 1

```

Private Sub Command1_Click()
    Dim a
    Dim b As Single
    a = InputBox("bir sayı giriniz")
    b = Val(a)
    Print b
End Sub

```

Dizi Değişkenler

Aynı tür bilgileri aynı anda bir değişkene atamak için kullanılan yöntemlerdir. Diziler birçok değişkene aynı adla ulaşmayı sağlayan bir grup veri yapısıdır. Dizilerin eleman sayısı 0 dan başlar. Eğer değişken numarasını 1 den başlamasını istiyorsak ilk satıra **Option Base 1** yazılır. Dizi değişkenlerin üzerinde çalışılan form veya modülün **declarations** kısmında tanımlanması gerekir.

Örnek 1

```

Private Sub Form_Load()
    Dim ad(2) As String
    Dim no(2) As Integer
    ad(0) = "oguz"
    ad(1) = "ahmet"
    ad(2) = "mehmet"
    no(0) = 25
    no(1) = 75
    no(2) = 85
End Sub

```

Örnek 2

Bir sınıftaki öğrencilerin notlarını toplayıp ortalamalarını almak.

General_Declarations

Option Explicit

Option Base 1

Dim ogrenci(30) As Integer

Private Sub Form_Load()

Dim sayi As Integer

Dim a, toplam, ortalama

sayi = InputBox("Öğrenci sayısını giriniz")

For a = 1 To sayi

ogrenci(a) = InputBox(Str(a) + ".Öğrencinin notunu girin")

toplam = toplam + ogrenci(a)

Next a

FontSize = 15

Show

ortalama = toplam / sayi

Print "Not Ortalaması :"; ortalama

End Sub

Örnek 2

Option Explicit

Option Base 1

Dim a(3) As Integer

Private Sub Command1_Click()

Dim hesap

a(1) = Text1.Text

a(2) = Text1.Text

a(3) = Text1.Text

hesap = a(1) + a(2) + a(3)

Print hesap

End Sub

Dinamik Diziler

Bu tip dizilerde değişkene atanacak dizi sayısında bir sınırlama yoktur.

Örnek 1

Option Explicit

Option Base 1

Private Sub Form_Load()

Dim ogrenci() As Integer

Dim sayi As Integer

Dim a, toplam, ortalama

sayi = InputBox("Öğrenci sayısını giriniz")

ReDim ogrenci(sayi) As Integer

For a = 1 To sayi

ogrenci(a) = InputBox(Str(a) + ".Öğrencinin notunu girin")

toplam = toplam + ogrenci(a)

Next a

FontSize = 15

```

Show
ortalama = toplam / sayi
Print "Not Ortalaması :"; ortalama
End Sub

```

Function Oluşturma

Bir procedure olarak düzenlenen fonksiyonlar belli bir işlemi kod olarak içeren ve bir değeri geri döndüren program parçalarıdır. Örneğin bir çok yerde faktöriyel hesaplamanız gerekmektedir, her seferinde faktöriyel formülünü yazmaktansa bu işlemi bir fonksiyonda tanımlayıp sonra bu fonksiyonu çağırırız. Fonksiyonlar **Function** deyimi ile başlayıp **End Function** deyimi ile biter. Bir function oluşturmak için code penceresi açıkken Tools menüsünden Add Procedure seçilir.

Örnek 1

Klavyeden girilen bir sayının faktöriyelini bulan fonksiyon. Önce fak bir function tanımlanır.

```

Public Function fak(x)
Dim a, b
    b = 1
    For a = 1 To x
        b = b * a
    Next
    fak = b
End Function

Private Sub Command1_Click()
    Dim a
    a = InputBox("Lütfen bir sayı giriniz")
    Print a & "! =" & fak(a)
End Sub

```

Örnek 2

Klavyeden girilen bir sayının faktöriyelini bulan ve tüm sayıları ekrana yazan fonksiyon. Önce fak bir function tanımlanır.

```

Public Function fak(x)
Dim a, b
    b = 1
    For a = 1 To x
        b = b * a
    Next
    fak = b
End Function

Private Sub Command1_Click()
    Dim a, b
    a = InputBox("Lütfen bir sayı giriniz")
    For b = 1 To a
        Print b & "! =" & fak(b)
    Next
End Sub

```

Örnek 3

Klavyeden girilen bir sayının fahrenheit cinsinden değerini veren fonksiyon.

```
Public Function fah(x)
    fah = x * 9 / 5 + 32
End Function

Private Sub Command1_Click()
    Cls
    Dim a
    a = InputBox("Lütfen bir sayı giriniz")
    MsgBox a & " °C Fahrenheit cinsinden karşılığı =" & fah(a)
End Sub
```

Örnek 4

Klavyeden girilen bir sayının karesini ve karakökünü bulan fonksiyon.

```
Public Function kare(x)
    kare = x ^ 2
End Function

Public Function karekok(x)
    karekok = Sqr(x)
End Function

Private Sub Command1_Click()
    Cls
    Dim a
    a = InputBox("Lütfen bir sayı giriniz")
    Print a & " sayısının karesi =" & kare(a)
    Print a & " sayısının karakökü =" & karekok(a)
End Sub
```

Örnek 4

bir üçgenin hipotenüsünü bulan function.

```
Public Function hipo(x, y)
    hipo = Sqr(x ^ 2 + y ^ 2)
End Function

Private Sub Command1_Click()
    Cls
    Dim a, b
    a = InputBox("Lütfen birinci kenarı giriniz")
    b = InputBox("Lütfen ikinci kenarı giriniz")
    Print hipo(a, b)
End Sub
```

Componentler (Kontrol Elemanları)

Textbox (Metin Kutusu)

Bilgi girişi için kullanılır. Programlarda kullanılan adı, soyadı, telefonu gibi alanlardır. Metin kutularına girilen bilgiler daha sonra başka amaçlar için kullanılabilirler. Genel bir alışkanlık olarak textbox ların önüne bir label nesnesi konularak kullanıcı bilgilendirilir.

Özellikleri :

Name

Yordam içerisinde text kutusunun adını değiştirmek için kullanılır.

Alignment

Textbox içerisindeki yazının sağa, sola veya ortaya yazılmasını sağlar.

0 = Sola
1 = Sağa
2 = Ortaya

```
Private Sub Command1_Click()  
    Text1.Alignment = 0  
End Sub
```

Appearance

Text kutusunun üç boyutlu olarak görünümünü sağlar.

0 = Düz
1 = Üç boyutlu

```
Private Sub Command1_Click()  
    Text1.Appearance = 0  
End Sub
```

BackColor

Text kutusunun zemin rengini değiştirir. Properties penceresinden değiştirebildiğimiz gibi aşağıdaki kodlarla da değiştirebiliriz.

```
Private Sub Command1_Click()  
    Text1.BackColor = QBColor(4)  
End Sub
```

```
Private Sub Command1_Click()  
    Text1.BackColor = RGB(120, 25, 30)  
End Sub
```

BorderStyle

Text kutusunun çerçeve özelliğini belirler.

0 = Çerçeve yok
1= Çerçeve var

CausesValidation

Text kutusundan çıkarken bir şartın gerçekleşip gerçekleşmeyeceğini belirtmeye yarar. Default olarak true dir.

Örnek 1

```
Private Sub Form_Load()  
    Text1 = ""
```

```

Text2 = ""
End Sub

Private Sub Text1_Change()
    Print Text1
End Sub

Private Sub Text1_Validate(Cancel As Boolean)
    If 100 < Val(Text1) < 0 Then
        MsgBox ("0 - 100 arasında bir sayı giriniz")
        Cancel = True ` Burada Cancel = False olursa kodu işletir ama diğer yordama geçer.
    End If
End Sub

```

Örnek 2

```

Private Sub Text2_Validate(Cancel As Boolean)
    Dim a
    a = Val(Text2.Text)
    If Text2.Text <> a Then
        MsgBox ("Lütfen sadece rakam giriniz")
        Cancel = True
    End If
End Sub

```

Enabled

Text kutusunun aktif veya pasif olmasını sağlar.

```

Private Sub Command1_Click()
    Text1.Enabled = True ` False olursa kullanıcı giriş yapamaz.
End Sub

```

Font

Text kutusunun yazı özelliklerini değiştirir.

```

Private Sub Command1_Click()
    Text1.FontBold = True
    Text1.FontItalic = True
    Text1.FontName = "tahoma"
    Text1.FontSize = 15
    Text1.FontStrikethru = True
    Text1.FontUnderline = True
    Text1.ForeColor = QBColor(4)
End Sub

```

Program esnasında bütün textlerin özelliklerini bir anda değiştirmemizde mümkündür. Bunun için öncelikle bir text çizilir ve sonra bu text copy – paste yapılır. Çıkan soruya YES denir. Bu işlemin sonucunda aynı ad ve özelliklere sahip text kutuları oluşmuş olur. Onları birbirinden ayıran indis numaralarıdır.

```

Private Sub Command1_Click()
    Dim a
    For a = 0 To 5
        Text1(a).FontBold = True
        Text1(a).FontItalic = True
    Next a
End Sub

```



```
Text1(a).FontName = "tahoma"  
Text1(a).FontSize = 15  
Next  
End Sub
```

Height

Text kutusunun yüksekliğini ayarlar.

```
Private Sub Command1_Click()  
Text1.Height = 950  
End Sub
```

Index

Aynı isimli birden fazla nesne varsa VB bunlara bir index numarası vererek ayırır.

Left

Text kutusunun soldan uzaklığını ayarlar.

Locked

Text kutusunu veri girilemeyecek şekilde kitler.

MaxLength

Text kutusuna girilebilecek maksimum karakter sayısını belirler.

```
Private Sub Command1_Click()  
Text1.MaxLength = 5  
End Sub
```

MousePointer, MouseIcon

Text kutusunun üzerine mouse la gelindiği zaman mousenin alacağı şekli gösterir. 0 – 15 arasında bir değer verilebilir. 99 verirse bizim MouseIcon da seçtiğimiz icon görüntülenir.

```
Private Sub Command1_Click()  
Static a  
Text1.Text = Str(a) + "mouse şekilleri"  
Text1.MousePointer = a  
a = a + 1  
If a = 15 Then End  
End Sub
```

MultiLine

Text kutusunun için birden fazla satır açmak için kullanılır.

PasswordChar

Text kutusunu şifre girmek için kullanılacaksa; kullanıcının girdiği karakterleri belli etmemek için kullanılır. Multiline özelliği = True olunca çalışmıyor.

```
Private Sub Form_Load()  
Text1.Text = ""  
Text1.PasswordChar = "o"  
End Sub
```

SelStart, SelLength, SelText

Text kutusunun içinde seçme yapmak için kullanılır.

Örnek 1

```
Private Sub Command1_Click()  
    Text1.SelStart = 0  
    Text1.SelLength = 6  
    Text2.Text = Text1.SelText  
End Sub
```

Text1.SelLength = Len(Text1) 'yazarsak text1 in tamamını kopyalamış oluruz.

Örnek 2

```
Private Sub Command1_Click()  
If Text1.SelLength > 0 Then 'seçili bir kısım varsa  
    Text1.SelText = UCase(Text1.SelText) 'Seçili kısmı büyük harfe çevir.  
Else  
    Text1.Text = UCase(Text1.Text)  
End If  
End Sub
```

```
Private Sub Command2_Click()  
If Text1.SelLength > 0 Then 'seçili bir kısım varsa  
    Text1.SelText = LCase(Text1.SelText) 'Seçili kısmı küçük harfe çevir.  
Else  
    Text1.Text = LCase(Text1.Text)  
End If  
End Sub
```

ScrollBars

Text kutusunun MultiLine özelliği True olması durumunda etkin olur ve text kutusuna kaydırma çubuklarını ekler.

TabIndex

VB form üzerinde kullanıcının ulaşabildiği her nesneye bir tabindeksi verir. Bu indeks numarası yardımıyla tab tuşuna basıldığı zaman kontroller arasında geçiş sağlanır.

TabStop

Bilgi girişinin çok olduğu formlarda bazı kutulara tabstop koyarak tab tuşuyla bu kontrole ulaşmayı engelleyebiliriz.

Text

Text kutusuna girilen metne ulaşmak için veya bir metin atamak için kullanılır.

```
Private Sub Command1_Click()  
    Text1.Text = "prestige"  
End Sub
```

```
Private Sub Form_Load()  
    Dim a, b, c, d  
    a = InputBox("adınızı giriniz")
```

```

b = InputBox("soy adınızı giriniz")
c = InputBox("telefonunuzu giriniz")
d = InputBox("adresinizi giriniz")
Text1.Text = a
Text2.Text = b
Text3.Text = c
Text4.Text = d
End Sub

```

TooltipText

Program içerisinde text kutusunun üzerine gelindiği zaman açılan bir bilgi penceresi oluşturur.

```

Private Sub Form_Load()
    Text1.ToolTipText = "Adı Bilgisi"
    Text2.ToolTipText = "Soyadı Bilgisi"
    Text3.ToolTipText = "Telefon Bilgisi"
    Text4.ToolTipText = "Adres Bilgisi"
End Sub

```

Top

Text kutusunun yukarıdan olan uzaklığını ayarlar.

Visible

Text kutusunun ekranda görünmesi sağlar.

Width

Text kutusunun genişliğini ayarlamak için kullanılır.

Text Change

Text kutusunun içerdiği metin değiştiği anda bir olayın olmasını istiyorsak bu yordama yazarız.

Örnek 1

```

Private Sub Text1_Change()
    Print Text1
End Sub

```

Örnek 2

Rakamlar arasına virgöl koyma.

```

Private Sub Text1_Change()
    Dim a
    a = Text1.SelStart
    Text1 = Format(Text1, "###,###")
    Text1.SelStart = a + 1
End Sub

```

Text Click, DbClick

Text kutusuna fare ile birkez tıklandığı zaman bir olayın olmasını istiyorsak bu yordama yazarız.

Örnek 1

```
Private Sub Text1_Click()
    Text1.BackColor = QBColor(Rnd * 15)
End Sub
```

Örnek 2

```
Private Sub Text1_DblClick()
    Text1.BackColor = QBColor(Rnd * 15)
End Sub
```

Text GotFocus

Text kutusuna gelindiği anda meydana gelmesini istediğimiz bir olayın kodunu bu yordama yazarız.

Örnek 1

```
Private Sub Text1_GotFocus()
    Text1.SelStart = 0
    Text1.SelLength = Len(Text1)
End Sub
```

Text LostFocus

Text kutusundan çıkıldığı anda meydana gelmesini istediğimiz bir olayın kodunu bu yordama yazarız.

Örnek 1

```
Private Sub Text1_LostFocus()
    Text1.Text = UCase(Text1)
End Sub
```

Örnek 2

```
Private Sub Text1_LostFocus()
    If 100 < Val(Text1) < 0 Then
        MsgBox ("0 - 100 arasında bir sayı giriniz")
        Cancel = True ' Burada Cancel = False olursa kodu işletir ama diğer yordama geçer.
        Text1.SetFocus
        Text1.SelStart = 0
        Text1.SelLength = Len(Text1)
    End If
End Sub
```

Text KeyPress, KeyUp, KeyDown

KeyDown : Klavyeden bir tuşa basıldığı zaman istediğimiz bir olayın kodunu bu yordama yazarız.

KeyUp : Klavyeden bir tuş bırakıldığı zaman istediğimiz bir olayın kodunu bu yordama yazarız.

KeyPress : KeyDown ile KeyUp arasında oluşan olayları kontrol eder.

Örnek 1

Ctrl tuşlarıyla beraber ok tuşları kullanılınca text kutusu yön değiştirecek, alt tuşuyla beraber basılınca boyutları değişecek.

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
    If Shift And vbCtrlMask Then
        Select Case KeyCode
            Case vbKeyLeft: Text1.Left = Text1.Left - 25
```

```

        Case vbKeyRight: Text1.Left = Text1.Left + 25
        Case vbKeyUp: Text1.Top = Text1.Top - 25
        Case vbKeyDown: Text1.Top = Text1.Top + 25
    End Select
End If
If Shift And vbAltMask Then
    Select Case KeyCode
        Case vbKeyLeft: Text1.Width = Text1.Width - 25
        Case vbKeyRight: Text1.Width = Text1.Width + 25
        Case vbKeyUp: Text1.Height = Text1.Height - 25
        Case vbKeyDown: Text1.Height = Text1.Height + 25
    End Select
End If

```

Örnek 2

```

Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
    If Shift And vbCtrlMask Then

        Select Case KeyCode
            Case vbKeyLeft:
                Text1.Left = Text1.Left - 80
            Case vbKeyRight:
                Text1.Left = Text1.Left + 80
            Case vbKeyUp:
                Text1.Top = Text1.Top - 80
            Case vbKeyDown:
                Text1.Top = Text1.Top + 80
        End Select
    End If
End Sub

```

Örnek 3

Klavyeden yazı yazarken büyük harfe çevirir.

```

Private Sub Text1_KeyPress(KeyAscii As Integer)
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
End Sub

```

MouseUp, MouseDown

MouseDown : Text kutusu üzerindyeiken mousenin bir tuşuna basıldığı zaman istediğimiz bir olayın kodunu bu yordama yazarız.

MouseUp : Mousenin bir tuşundan parmağımızı kaldırdığımız zaman istediğimiz bir olayın kodunu bu yordama yazarız.

Örnek 1

```

Private Sub Text1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Dim a
    Text1.BackColor = QBColor(4)
End Sub

Private Sub Text1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Dim a
    Text1.BackColor = QBColor(0)

```

End Sub

MouseMove

MouseMove : Text kutusu üzerindeyken mouse ile dolaşırken istediğimiz bir olayın kodunu bu yordama yazarız.

Örnek 1

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Text1 = "Visual Basic"
End Sub
```

```
Private Sub Text1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Text1.Text = UCase(Text1)
End Sub
```

Örnek 2

Text kutusu üzerinde gezinirken mousenin koordinatlarını verir.

```
Private Sub Text1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Text1.Text = "fare şu anda " & " dikey :" & Y & " yatay :" & X & "koordinatlarında"
End Sub
```

Örnek 3

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Static a, b
    a = Int(Rnd * 40) + 1
    b = Int(Rnd * 16000) + 1
    DrawWidth = a
    PSet (X, Y), b
End Sub
```

Örnek 4

Dim a, b

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    a = X
    b = Y
    PSet (X, Y)
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    If Button = vbRightButton Then Circle (a, b), Abs(a - X) ' sağ tuşa basılıyken çember çiz
```

```
    If Button = vbLeftButton Then Line (a, b)-(X, Y), , B 'sol tuşa basılıyken kare çiz
```

```
End Sub
```

Label (Etiket)

Form üzerinde bir bilginin gösterilmesi için veya textbox ın açıklaması olarak kullanılır.

Özellikleri :

AutoSize

Label içindeki bilginin uzunluğuna göre otomatik olarak boyutlarının ayarlanmasını sağlar.

BackStyle

Label transparent veya opak olmasını sağlar.

Caption

Label üzerindeki yazıyı buradan ayarlarız.

Örnek 1

```
Private Sub Command1_Click()  
    Label1.Caption = "Adı Soyadı"  
End Sub
```

Command Button (Komut Düğmesi)

Bir olayın kullanıcı tarafından başlatılması için kullanılır.

Özellikleri :

Cancel

Klavyeden ESC tuşuna basıldığı zaman command buttonun aktif olması durumudur.

```
Private Sub Command1_Click()  
    Command1.Caption = "Esc tuşuna basıldı"  
End Sub
```

```
Private Sub Form_Load()  
    Command1.Cancel = True  
End Sub
```

Caption

Command Buttonun adını belirtir. Yazdığınız kelimelerin arasına & işareti konursa kısayol tuşu görevi görür.

```
Private Sub Form_Load()  
    Command1.Caption = "o&guz"  
    Command2.Caption = "og&uz"  
End Sub
```

Picture, DisabledPicture, DownPicture

Style özelliği 1 verilerek Command Buttonun resimli bir komut düğmesi olarak belirlendikten sonra.

Picture : Komut düğmesi aktif durumdayken

DisabledPicture : Komut düğmesi pasif durumdayken

DownPicture : Komut düğmesi basılı durumdayken

Enabled

Command Buttonun aktif olup olmamasının sağlar.

```
Private Sub Text1_Change()  
    If Text1.Text <> "" And Text2.Text <> "" Then  
        Command1.Enabled = True  
    End If  
End Sub
```

```
Private Sub Text2_Change()  
    If Text1.Text <> "" And Text2.Text <> "" Then  
        Command1.Enabled = True  
    End If  
End Sub
```

MouseMove

```
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    Dim a As Integer, b As Integer  
    a = Int(Rnd * 5000)  
    b = Int(Rnd * 5000)  
    If a > 7000 Then a = 3500  
    If b > 7000 Then b = 2700  
    Command1.Left = a  
    Command1.Top = b  
End Sub
```

OYUN

Sayı bulma oyunu, 4 Label ,1 TextBox, 2 Command Buton

```
Private Sub Command1_Click()  
    Command2.Enabled = True  
    Command1.Enabled = False  
    Label1 = Int(Rnd * 100)  
    Label4 = 1000  
    Label3 = 0  
    Label2 = "0 ile 100 arasında bir sayı girin"  
    Text1.SetFocus  
End Sub
```

```
Private Sub Command2_Click()  
    Label3 = Label3 + 1  
    Label4 = Label4 - Label3 * 10  
  
    If Label1 < Text1 Then  
        Label2 = "Küçült"  
    Else  
    If Label1 > Text1 Then  
        Label2 = "Büyült"  
    Else  
        Label2 = "Tebrikler " + Label3 + " defada bildiniz " + Label4 + " puan aldınız "  
        Command1.Enabled = True  
        Command2.Enabled = False  
    End If  
End If
```



```

Text1 = ""
Text1.SetFocus
End Sub

Private Sub Form_Load()
    Command1.Default = True
    Text1 = ""
    Command2.Enabled = False
    Label1.Visible = False
End Sub

```

PASSWORD

```

Private Sub Command1_Click()
    Text1 = "" : Text2 = ""
End Sub

Private Sub Command2_Click()
    If Text1 = "oguz" And Text2 = "kutlu" Then
        MsgBox ("Merhaba")
        Exit Sub
    End If
    If Text1 <> "oguz" Then
        Text1.SetFocus
        Text1.SelStart = 0
        Text1.SelLength = Len(Text1)
        Exit Sub
    End If
    If Text2 <> "kutlu" Then
        Text2.SetFocus
        Text2.SelStart = 0
        Text2.SelLength = Len(Text2)
        Exit Sub
    End If : End Sub

Private Sub Text2_Change()
    If Text1 <> "" Then Command2.Enabled = True
    If Text2 = "" Then Command2.Enabled = False
End Sub

Private Sub Text1_Change()
    If Text2 <> "" Then Command2.Enabled = True
    If Text1 = "" Then Command2.Enabled = False
End Sub

```

CheckBox (İşaret Kutusu)

Belli seçeneklerin seçilmesini sağlar.

Özellikleri :

Alignment

Check işaretinin solda veya sağda olmasını sağlar.

Value

Check işaretinin özelliklerini belirler.

0 : İşaretsiz

1 : İşaretli
2 : Belirsiz

```
If Check1.Value = 0 Then Text1.Text = "İşaretsiz"  
If Check1.Value = 1 Then Text1.Text = "İşaretli"  
If Check1.Value = 2 Then Text1.Text = "Belirsiz"
```

Örnek 1

1 tane text ve 3 tane checkbox çiz ve style =1 yap sonra \common\graphics\bitmaps\tlbr_w95 dizininden gerekli resimleri üzerine koy (bold, underline, italic)

```
Private Sub Check1_Click()  
    Text1.FontBold = Check1.Value  
End Sub  
  
Private Sub Check2_Click()  
    Text1.FontItalic = Check2.Value  
End Sub  
  
Private Sub Check3_Click()  
    Text1.FontUnderline = Check3.Value  
End Sub
```

Option Button (Seçenek Düğmesi)

Örnek 1

3 tane text ve 2 tane optionbox çizin,

```
Private Sub Form_Load()  
    Show  
    Text1.SetFocus  
    Option1 = False  
    Option2 = False  
End Sub  
  
Private Sub Option1_Click()  
    Text2.Text = Val(Text1.Text) * 0.17  
    Text3.Text = Val(Text1.Text) + Val(Text2.Text)  
    Text3 = Format(Text3, "###,###")  
End Sub  
  
Private Sub Option2_Click()  
    Text2.Text = Val(Text1.Text) * 0.25  
    Text3.Text = Val(Text1.Text) + Val(Text2.Text)  
    Text3 = Format(Text3, "###,###")  
End Sub
```

Horizontal & Vertical ScrollBar (Kaydırma Çubuğu)

Value

Kaydırma çubuğunun aldığı değeri gösterir. Bu değer min. - max. Arasında olabilir.

Min. Max.

Kaydırma çubuğunun alabileceği değerleri gösterir.

Large Change

Kaydırma çubuğunun üzerine tıklandığı zaman scroll.value'nin uğrayacağı değişimi gösterir.

Small Change

Kaydırma çubuğunun üzerine oklara tıklandığı zaman scroll.value'nin uğrayacağı değişimi gösterir.

Scroll

Kaydırma çubuğunun üzerindeki scrollun değiştiği zaman bir olayın olmasını istiyorsak bu yordama yazarız.

Örnek 1

1 textbox, 3 hscroll , 3 label çiz.

```
Private Sub Form_Load()
```

```
    HScroll1.Max = 255
```

```
    HScroll2.Max = 255
```

```
    HScroll3.Max = 255
```

```
End Sub
```

```
Private Sub HScroll1_Change()
```

```
    Text1.BackColor = RGB(HScroll1.Value, HScroll2.Value, HScroll3.Value)
```

```
    Label1.Caption = HScroll1.Value
```

```
End Sub
```

```
Private Sub HScroll2_Change()
```

```
    Text1.BackColor = RGB(HScroll1.Value, HScroll2.Value, HScroll3.Value)
```

```
    Label2.Caption = HScroll2.Value
```

```
End Sub
```

```
Private Sub HScroll3_Change()
```

```
    Text1.BackColor = RGB(HScroll1.Value, HScroll2.Value, HScroll3.Value)
```

```
    Label3.Caption = HScroll3.Value
```

```
End Sub
```

Shape (Şekil KontrolElemanı)

Formlara dikdörtgen, kare, daire gibi geometrik şekiller çizmemize yarar.

BorderStyle

Çizilecek olan şeklin kenar çizgi özelliklerini belirtir.

- | | |
|---|----------------------------------|
| 0 | Çizgi yok |
| 1 | Düz çizgi |
| 2 | Uzun kesikli çizgi |
| 3 | Nokta çizgi |
| 4 | Kesikli + noktalı çizgi |
| 5 | Kesikli + noktalı çizgi+ noktalı |

6 Inside solid (şekil ile çerçeve kenarları çakışık şekilde)

BorderWidth

Çizilecek olan şeklin kenar çizgi kalınlığını değiştirir.

FillColor

Çizilecek olan şeklin iç rengini değiştirir.

FillStyle

Çizilecek olan şeklin iç desenini belirtir.

0	Tam dolu
1	Nesnenin zemin rengi
2	Yatay çizgi
3	Dikey çizgi
4	Sola eğik
5	Sağa eğik
6	Kareli
7	Çapraz

Shape

Çizilecek olan şeklin şeklini belirtir.

0	Dikdörtgen
1	Kare
2	Elips
3	Çember
4	Oval Dörtgen
5	Oval Kare

Örnek 1

Scrool 'ları kullanarak Shape özelliklerini gösteren program yap.

4 Vscroll, 1 Shape

Timer Control (Zamanlayıcı)

Enabled

True olursa zaman çalışmaya başlar. False olursa durur.

Interval

Saatın çalışma periyodunu belirtir. 0 – 65535 arasında değer alır. 1000 olursa 1 saniye olur.

```
Private Sub Form_Load()  
    Label1.Caption = Now
```

End Sub

```
Private Sub Timer1_Timer()  
    Label1.Caption = Now  
End Sub
```

Örnek 1

Saat

```
Private Sub Form_Load()  
    Label1.Caption = Time  
End Sub
```

```
Private Sub Timer1_Timer()  
    Label1.Caption = Time  
End Sub
```

Örnek 2

Kronometre

Option Explicit
Dim saat, dakika, saniye As Integer

```
Private Sub basla_Click()  
    saat = 0  
    dakika = 0  
    saniye = 0  
End Sub
```

```
Private Sub durdur_Click()  
    Label2.Caption = saat & " : " & dakika & " : " & saniye  
End Sub
```

```
Private Sub Form_Load()  
    Timer1.Interval = 1000  
    Label1.Visible = False  
End Sub
```

```
Private Sub Timer1_Timer()  
    Label1.Caption = Time  
    saniye = saniye + 1  
    If saniye = 59 Then  
        dakika = dakika + 1  
        saniye = 0  
    End If  
    If dakika = 59 and saniye = 59 Then  
        saat = saat + 1  
        dakika = 0  
    End If  
End Sub
```

Örnek 3

```
Private Sub Form_Load()  
    Label1.Caption = Time
```

```

Label2.Caption = Year(Now) & Month(Now) & Day(Now) & Hour(Now) & Minute(Now) &
Second(Now)
Timer2.Enabled = False
End Sub

```

```

Private Sub Timer1_Timer()
Label1.Caption = Time
If Hour(Now) = 20 And Minute(Now) = 59 Then Timer2.Enabled = True
End Sub

```

```

Private Sub Timer2_Timer()
Label2.Caption = Second(Now)
End Sub

```

Örnek 4

Ekrana Koruyucu

```

Private Sub Form_Load()
    Timer1.Interval = 1000
End Sub
Private Sub Timer1_Timer()
Dim x, y, a, b
    a = Form1.ScaleWidth
    b = Form1.ScaleHeight
    x = Int(Rnd * a)
    y = Int(Rnd * b)
    Line -(x, y)
    DrawWidth = Int(Rnd * 20) + 1
    ForeColor = Int(Rnd * 16000000)
End Sub

```

Örnek 5

Top oyunu

```

Form_KeyPreview = True

```

```

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyRight Then Command1.Left = Command1.Left + 100
    If Command1.Left + Command1.Width >= Form1.ScaleWidth Then Command1.Left =
Form1.ScaleWidth - Command1.Width
    If KeyCode = vbKeyLeft Then Command1.Left = Command1.Left - 100
    If Command1.Left <= 0 Then Command1.Left = 0
End Sub

```

```

Private Sub Form_KeyPress(KeyAscii As Integer)
    If Timer1.Enabled = True Then Timer1.Interval = 1
End Sub

```

```

Private Sub Form_Load()
    Command1.Left = (Form1.ScaleWidth / 2) - (Command1.Width / 2)
    Option1.Left = (Form1.ScaleWidth / 2) - (Option1.Width / 2)
    Option1.Top = Command1.Top - Option1.Height
End Sub

```

```

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Command1.Left = Form1.ScaleWidth - Command1.Left Then

```

```

Command1.Left = Form1.ScaleWidth - Command1.Left
Else
Command1.Left = X
End If
End Sub
Private Sub Timer1_Timer()
    Static sx, sy, X, Y
    Randomize Timer

```

If IsEmpty(sx) Then ' **değişkenin boş olup olmadığını kontrol eder ve boolean değer döndürür. Static olarak tanımladığımız (sx) programı ilk çalıştırdığımızda bir değeri olmadığı için isEmpty çalışacak ve sx=50 olacak, program geri döndüğünde sx bu sefer bir değere sahip olacağı için isEmpty çalışmayacak ve sx kendi değeri ile kalacak.**

```

        sx = 50
        sy = -50
        X = Option1.Left
        Y = Option1.Top
    End If
    If X <= 0 Then sx = -sx: Beep 'sol kenara gelince geri dön
    If X + Option1.Width > Form1.ScaleWidth Then sx = -sx: Beep 'sağ kenara gelince geri dön
    If Y < 0 Then sy = -sy: Beep 'üste gelince geri dön
    If (Y >= Command1.Top) And (Y < (Command1.Top + Command1.Height)) And X +
Option1.Width > Command1.Left And X < (Command1.Left + Command1.Width) Then sy = -sy
'command butona çarpınca geri dön
    If Y >= Form1.ScaleHeight Then 'top altta gelince oyunu bitir.
        MsgBox ("oyun bitti")
        Timer1.Interval = False
    End
End If
X = X + sx
Y = Y + sy
Option1.Left = X
Option1.Top = Y
End Sub

```

Örnek 6

Parlayan yazı : 2 label, 1 textbox, 1 timer, Timer1.Interval = 10

```

Private Sub Form_Load()
    Label2.Left = Label1.Left + Label2.Width
End Sub

Private Sub Text1_Change()
    If Len(Text1) = 0 Then Exit Sub
    Label1 = Left(Text1, Len(Text1) - 1)
    Label2 = Right(Text1, 1)
    Label1.FontSize = 15
    Label2.Left = Label1.Left + Label2.Width
End Sub

Private Sub Timer1_Timer()
    Static a, b
    If IsEmpty(a) Then
        a = Label1.FontSize
        b = 5
    End If
    If a > 50 Or a < 5 Then b = -b

```

```
a = a + b  
Label2.FontSize = a 'İkinci örnek olarak Labe1 yazılır.
```

End Sub

Frame Control (Çerçeve)

İçerdiği kontrolleri gruplamak için kullanılır. Genellikle Option Button'larda kullanılır. Enabled veya Visible özelliği kullanılır.

UpDown(Yukarı aşağı arttırma)

Textbox lara bilgi girişi yaparken kontrollü bir şekilde yapılmasını istiyorsak kullanırız. Bunun için Components menüsünden Microsoft Windows Common Controls-2 seçilerek ekrana getirilir.

Alignment

BuddyControl ile herhangi bir kontrole bağlantı kurulmuşsa UpDown o nesnenin sağında mı solunda mı olacağını ayarlar.

AutoBuddy

UpDown kontrolünün herhangi bir kontrole otomatik olarak atanmasını sağlar genellikle kullanılmaz.

BuddyControl

UpDown kontrolünün hangi kontrole atanmasını sağlar.

BuddyProperty

UpDown kontrolünün bağlandığı kontrolün hangi özelliğine bağlanacağını belirler.

Increment

UpDown kontrolünün her tıklamada ne kadar değişime uğrayacağını belirtiriz.

Min - Max

UpDown kontrolünün alt ve üst sınırlarını belirtiriz.

Orientation

UpDown kontrolünün yatay veya dikey olarak yerleştirilmesini sağlar.

SyncBuddy

UpDown kontrolünün herhangi bir kontrole atasanız bile bu özelliği False yaparak işlemi durdurabilirsiniz.

Value

UpDown kontrolünün aldığı değer.

Örnek 1

UpDown kullanarak bir text kutusunun fontsize ve forecolor özelliklerini, checkbox kullanarak textin Alignment, Bold, İtalic, Underline özelliklerini Vscroll kullanarak Backcolor özelliğini değiştirelim. Ekranda bir saat ve ToolTipte tarih olsun. Form büyütüldüğü zaman Text ona göre davransın.

Line (Çizgi)

ComboBox (Açılan Liste)

IntegralHeight

Bu özellik True ise açılan kutuda en alttaki yazının tam olarak görünmesi için kendisini otomatik olarak boyutlar.

List

Combo içine direk veri girişi için kullanılır.

ListIndex

Combo içindeki yazan textlerden seçim yapmayı sağlar. İlk veri **0** son veri ise **ListCount – 1** dir.

Sorted

Combo içindeki yazıların sıralı olmasını sağlar.

Style

Combo tipini ve çalışma biçimini düzenler.

- | | |
|---|---|
| 0 | Aşağı doğru açılan ve içeriği değiştirilebilen combo. |
| 1 | Aşağı doğru açılmayan ama içeriği değiştirilebilen combo. |
| 2 | Aşağı doğru açılan ama veri girişi yapılamayan combo. |

Text

Combonun aktif olan texti.

```
Private Sub Combo1_Click()  
    Text1.Text = Combo1.Text  
End Sub
```

Aynı zamanda seçilen texte görede program işletebiliriz.

```
Private Sub Combo1_Click()  
    If Combo1.Text = "kırmızı" Then Text1.BackColor = RGB(255, 0, 0)  
    If Combo1.Text = "yeşil" Then Text1.BackColor = RGB(0, 255, 0)  
    If Combo1.Text = "mavi" Then Text1.BackColor = RGB(0, 0, 255)  
End Sub
```

Örnek 1

Yazı sitilini ve büyüklüğünü değiştiren program.

```
Private Sub Combo1_Click()  
    Text2.FontName = Combo1.Text
```

```

End Sub

Private Sub Form_Load()
Dim i
    UpDown1.BuddyControl = Combo1
    UpDown1.BuddyProperty = "listindex"

    UpDown2.BuddyControl = Text1
    UpDown2.BuddyProperty = "text"

    For i = 0 To Screen.FontCount - 1
        Combo1.AddItem Screen.Fonts(i)
    Next

    Combo1.ListIndex = 0

    UpDown1.Min = 0
    UpDown1.Max = Combo1.ListCount - 1
    UpDown1.Increment = 1

    UpDown2.Min = 0
    UpDown2.Max = 100
    UpDown2.Increment = 5
End Sub

Private Sub Text1_Change()
    Text2.FontSize = Val(Text1)
End Sub

```

AddItem

Combonun listesine yeni bir text eklemek için kullanılır.

```

Private Sub Combo1_Click()
    If Combo1.Text = "kırmızı" Then Text1.BackColor = RGB(255, 0, 0)
    If Combo1.Text = "yeşil" Then Text1.BackColor = RGB(0, 255, 0)
    If Combo1.Text = "mavi" Then Text1.BackColor = RGB(0, 0, 255)
End Sub

Private Sub Form_Load()
    Combo1.AddItem "kırmızı"
    Combo1.AddItem "mavi"
    Combo1.AddItem "yeşil"
    Combo1.ListIndex = 0
End Sub

```

RemoveItem

Combonun listesinden bir text çıkartmak için kullanılır.

```

Private Sub Command1_Click()
    Combo1.RemoveItem 0
End Sub

```

ListBox (Listeleme Kutusu)

Elemanların gösterilmesini sağlayan komponentdir.

Columns

Listbox ın içine kolon koymamıza yarar.

MultiSelect

Listbox içerisindeki verilerin seçilmesini sağlar.

- 0 Seçim yapılamaz
- 1 Fare ile üzerine gidilen text seçilebilir.
- 2 Klavyeden Shift veya Ctrl tuşuyla seçim yapmamızı sağlar.

Style

Listbox içeriği bir checkbox gibi olur.

AddItem

ListBoxın içeriğine bir veri girişi yapmak için kullanılır.

```
Private Sub Form_Load()  
    List1.AddItem "kırmızı"  
    List1.AddItem "mavi"  
    List1.AddItem "yeşil",1 'Buradaki 1 rakamı yeşil yazısını 2.sırada görünmesini sağlar.  
End Sub
```

RemoveItem

ListBoxın içeriğinden bir veriyi silmek için kullanılır.

```
Private Sub Command1_Click()  
    List1.RemoveItem 2  
End Sub
```

Text

ListBoxın içeriğinden bir veriyi göstermek için kullanılır.

```
Private Sub List1_Click()  
    Text1.Text = List1.Text  
End Sub
```

Clear

ListBoxın içeriğini tamamen silmek için kullanılır.

```
Private Sub Command1_Click()  
    List1.Clear  
End Sub
```

ListIndex

ListBoxın içerisinde seçilmiş olan veriyi veya seçilecek olan verinin index numarası. Aynı zamanda yan yana 2-3 ListBox olduğunda birbirlerine bağlamak içinde kullanılır.

```
Private Sub Command1_Click()  
    List1.RemoveItem List1.ListIndex  
End Sub
```

ListCount

ListBoxın içerisindeki veri sayısını verir. **List1.ListCount – 1** son elemanı verir.

```
Private Sub Command1_Click()  
    MsgBox ("listede " & List1.ListCount & " eleman var")  
End Sub
```

SelCount

MultiSelect özelliği 1 veya 2 olan Listbox larda seçili eleman sayısını verir.

```
Private Sub Command1_Click()  
    MsgBox ("listede " & List1.SelCount & " eleman var")  
End Sub
```

Selected

MultiSelect özelliği 1 veya 2 olan Listbox larda index numarasına göre seçim yapmayı sağlar.

Örnek 1

```
Private Sub Command1_Click()  
    Dim i  
    List2.Clear  
    For i = 0 To List1.ListCount - 1  
        If List1.Selected(i) Then  
            List2.AddItem List1.List(i)  
        End If  
    Next  
End Sub
```

```
Private Sub Form_Load()  
    List1.AddItem "1"  
    List1.AddItem "2"  
    List1.AddItem "3"  
    List1.AddItem "4"  
    List1.AddItem "5"  
    List1.AddItem "6"  
    List1.AddItem "7"  
    List1.AddItem "8"  
    List1.AddItem "5"  
    List1.AddItem "6"  
    List1.AddItem "7"  
    List1.AddItem "8"  
End Sub
```

Örnek 2

Tüm listeyi seçmek için bir program.
Private Sub Command2_Click()

```

Dim a
  For a = 1 To List1.ListCount
    List1.Selected(a - 1) = True
  Next
End Sub

```

Örnek 3

Listeye yazılan rakamları sıraya dizen program.

```

Private Sub Form_Load()
Randomize
  Dim i
  For i = 1 To 10
    List1.AddItem Int(Rnd * 100)
  Next
End Sub

Private Sub za_Click()
Dim i, j, c
  ReDim x(List1.ListCount - 1)
  For i = 0 To List1.ListCount - 1
    x(i) = List1.List(i)
  Next
  For i = 0 To List1.ListCount - 1
    For j = 0 To List1.ListCount - 1
      If Val(x(j)) < Val(x(i)) Then
        c = x(i)
        x(i) = x(j)
        x(j) = c
      End If
    Next
  Next
  For i = 0 To List1.ListCount - 1
    List1.List(i) = x(i)
  Next
End Sub

```

Örnek 4

Ctrl-A Kısayolunun çalıştığı program.

```

Private Sub List1_KeyDown(KeyCode As Integer, Shift As Integer)
  Dim a
  If Shift And vbCtrlMask Then
    Select Case KeyCode
      Case vbKeyA
        For a = 0 To List1.ListCount - 1
          List1.Selected(a) = True
        Next
      End Select
    End If
  End Sub

```

Örnek 5

Adres defteri

FORM1

```
Private Sub Command1_Click()  
    Me.Hide  
    Load Form2  
    Form2.Show  
End Sub
```

```
Private Sub Command2_Click()  
    Me.Hide  
    Load Form3  
    Form3.Show  
End Sub
```

```
Private Sub Command3_Click()  
    Me.Hide  
    Load Form4  
    Form4.Show  
End Sub
```

```
Private Sub Command4_Click()  
    Me.Hide  
    Load Form5  
    Form5.Show  
End Sub
```

```
Private Sub Command5_Click()  
End  
End Sub
```

FORM2

```
Private Sub cikis_Click()  
    Unload Form2  
    Form1.Show  
End Sub
```

```
Private Sub Form_Load()  
    Combo1.AddItem "1"  
    Combo1.AddItem "2"  
    Combo1.AddItem "3"  
    Combo1.AddItem "4"  
    Combo2.AddItem "Lisans"  
    Combo2.AddItem "Yüksek Lisans"  
    Combo2.AddItem "Doktora Lisans"  
    Combo3.AddItem "İstanbul"  
    Combo3.AddItem "Ankara"  
    Combo3.AddItem "İzmir"  
    Combo3.AddItem "Eskişehir"  
    Combo3.AddItem "Erzincan"  
    Combo1.ListIndex = 0  
    Combo2.ListIndex = 0  
    Combo3.ListIndex = 0  
    List1.Visible = False  
    List2.Visible = False  
    List3.Visible = False
```

```

List4.Visible = False
List5.Visible = False
List6.Visible = False
List7.Visible = False
List8.Visible = False
Dim ad, sad, sin, bol, adres, ilc, il, pkod
If Dir("ogren.dat") <> "" Then 'daha önce kayddilmiş dosya varsa onu aç yoksa
    Open "ogren.dat" For Input As #1
    While Not EOF(1) 'dosyanın sonuna kadar
        Input #1, ad, sad, sin, bol, adres, ilc, il, pkod
        List1.AddItem ad
        List2.AddItem sad
        List3.AddItem sin
        List4.AddItem bol
        List5.AddItem adres
        List6.AddItem ilc
        List7.AddItem il
        List8.AddItem pkod
    Wend
    Close #1
End If
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Dim ad, sad, sin, bol, adres, ilc, il, pkod, i
    Open "ogren.dat" For Output As #1
    For i = 0 To List1.ListCount - 1
        ad = List1.List(i)
        sad = List2.List(i)
        sin = List3.List(i)
        bol = List4.List(i)
        adres = List5.List(i)
        ilc = List6.List(i)
        il = List7.List(i)
        pkod = List8.List(i)
        Write #1, ad, sad, sin, bol, adres, ilc, il, pkod
    Next
    Close #1
End Sub

Private Sub kaydet_Click()
    List1.AddItem Text1
    List2.AddItem Text2
    List3.AddItem Combo1.Text
    List4.AddItem Combo2.Text
    List5.AddItem Text3
    List6.AddItem Text4
    List7.AddItem Combo3.Text
    List8.AddItem Text5
End Sub

```

FORM3

```

Private Sub Command1_Click()
    Unload Form3
    Form1.Show
End Sub

Private Sub Form_Load()

```

```

Dim ad, sad, sin, bol, adres, ilc, il, pkod
If Dir("ogren.dat") <> "" Then 'daha önce kaydedilmiş dosya varsa onu aç yoksa
    Open "ogren.dat" For Input As #1
    While Not EOF(1) 'dosyanın sonuna kadar
        Input #1, ad, sad, sin, bol, adres, ilc, il, pkod
        List1.AddItem ad
        List2.AddItem sad
        List3.AddItem sin
        List4.AddItem bol
        List5.AddItem adres
        List6.AddItem ilc
        List7.AddItem il
        List8.AddItem pkod
    Wend
    Close #1
End If
List2.Enabled = False
List3.Enabled = False
List4.Enabled = False
List5.Enabled = False
List6.Enabled = False
List7.Enabled = False
List8.Enabled = False
End Sub

```

```

Private Sub List1_Click()
List2.ListIndex = List1.ListIndex
List3.ListIndex = List1.ListIndex
List4.ListIndex = List1.ListIndex
List5.ListIndex = List1.ListIndex
List6.ListIndex = List1.ListIndex
List7.ListIndex = List1.ListIndex
List8.ListIndex = List1.ListIndex
List2.TopIndex = List1.TopIndex
List3.TopIndex = List1.TopIndex
List4.TopIndex = List1.TopIndex
List5.TopIndex = List1.TopIndex
List6.TopIndex = List1.TopIndex
List7.TopIndex = List1.TopIndex
List8.TopIndex = List1.TopIndex
End Sub

```

FORM4

```

Private Sub cikis_Click()
Unload Form5
Form1.Show
End Sub

```

```

Private Sub degis_Click()
Dim a
If List1.ListIndex < 0 Then
    MsgBox ("Lütfen bir eleman seçiniz")
    Exit Sub
End If
a = List1.ListIndex
List1.List(a) = Text1
List2.List(a) = Text2

```



```

List3.List(a) = Combo1.Text
List4.List(a) = Combo2.Text
List5.List(a) = Text3
List6.List(a) = Text4
List7.List(a) = Combo3.Text
List8.List(a) = Text5

Dim ad, sad, sin, bol, adres, ilc, il, pkod, i
Open "ogren.dat" For Output As #1
For i = 0 To List1.ListCount - 1
    ad = List1.List(i)
    sad = List2.List(i)
    sin = List3.List(i)
    bol = List4.List(i)
    adres = List5.List(i)
    ilc = List6.List(i)
    il = List7.List(i)
    pkod = List8.List(i)
    Write #1, ad, sad, sin, bol, adres, ilc, il, pkod
Next
Close #1

```

End Sub

```

Private Sub Form_Load()
    Combo1.AddItem "1"
    Combo1.AddItem "2"
    Combo1.AddItem "3"
    Combo1.AddItem "4"
    Combo2.AddItem "Lisans"
    Combo2.AddItem "Yüksek Lisans"
    Combo2.AddItem "Doktora Lisans"
    Combo3.AddItem "İstanbul"
    Combo3.AddItem "Ankara"
    Combo3.AddItem "İzmir"
    Combo3.AddItem "Eskişehir"
    Combo3.AddItem "Erzincan"
    Combo1.ListIndex = 0
    Combo2.ListIndex = 0
    Combo3.ListIndex = 0
    Dim ad, sad, sin, bol, adres, ilc, il, pkod
    If Dir("ogren.dat") <> "" Then 'daha önce kaydedilmiş dosya varsa onu aç yoksa
        Open "ogren.dat" For Input As #1
        While Not EOF(1) 'dosyanın sonuna kadar
            Input #1, ad, sad, sin, bol, adres, ilc, il, pkod
            List1.AddItem ad
            List2.AddItem sad
            List3.AddItem sin
            List4.AddItem bol
            List5.AddItem adres
            List6.AddItem ilc
            List7.AddItem il
            List8.AddItem pkod
        Wend
        Close #1
    End If
    List2.Enabled = False
    List3.Enabled = False

```

```

List4.Enabled = False
List5.Enabled = False
List6.Enabled = False
List7.Enabled = False
List8.Enabled = False

End Sub

Private Sub Form_Unload(Cancel As Integer)
    Dim ad, sad, sin, bol, adres, ilc, il, pkod, i
    Open "ogren.dat" For Output As #1
    For i = 0 To List1.ListCount - 1
        ad = List1.List(i)
        sad = List2.List(i)
        sin = List3.List(i)
        bol = List4.List(i)
        adres = List5.List(i)
        ilc = List6.List(i)
        il = List7.List(i)
        pkod = List8.List(i)
        Write #1, ad, sad, sin, bol, adres, ilc, il, pkod
    Next
    Close #1
End Sub

```

End Sub

```

Private Sub List1_Click()
List2.ListIndex = List1.ListIndex
List3.ListIndex = List1.ListIndex
List4.ListIndex = List1.ListIndex
List5.ListIndex = List1.ListIndex
List6.ListIndex = List1.ListIndex
List7.ListIndex = List1.ListIndex
List8.ListIndex = List1.ListIndex
List2.TopIndex = List1.TopIndex
List3.TopIndex = List1.TopIndex
List4.TopIndex = List1.TopIndex
List5.TopIndex = List1.TopIndex
List6.TopIndex = List1.TopIndex
List7.TopIndex = List1.TopIndex
List8.TopIndex = List1.TopIndex
Text1.Text = List1.Text
Text2.Text = List2.Text
Combo1.Text = List3.Text
Combo2.Text = List4.Text
Text3.Text = List5.Text
Text4.Text = List6.Text
Combo3.Text = List7.Text
Text5.Text = List8.Text
End Sub

```

Örnek 6

Öğrenci takip programı

FORM1

3 Command Buton çiz,

```

Private Sub cikis_Click()
    End
End Sub

Private Sub Kayit_Click()
    Load Form2
    Form2.Show
End Sub

Private Sub Listele_Click()
    Load Form3
    Form3.Show
End Sub

```

FORM2 Kayıt

17 Label, 5 textbox, 11 Listbox, 8 option button, 3 command

```

Private Sub cikis_Click()
    Dim a
    a = MsgBox("Kaydetmek istiyormusunuz", 4)
    If a = 6 Then
        Unload Form2
    Else
        End
    End If
End Sub

Private Sub Form_Load()
    Combo1.AddItem 1
    Combo1.AddItem 2
    Combo1.AddItem 3
    Combo1.AddItem 4
    Combo1.AddItem 5
    Combo1.AddItem 6
    Combo1.ListIndex = 0
    Dim a, b, c, d, e, f, g, h, m, n
    If Dir("kurs.dat") <> "" Then
        Open "kurs.dat" For Input As #1
        While Not EOF(1)
            Input #1, a, b, c, d, e, f, g, h, m, n
            List1.AddItem a
            List2.AddItem b
            List3.AddItem c
            List4.AddItem d
            List5.AddItem e
            List6.AddItem f
            List7.AddItem g
            List8.AddItem h
            List9.AddItem m
            List10.AddItem n
        Wend
        Close #1
    End If
    Option1 = False
    List11.Clear
    List11.FontName = "tahoma"
    List11.FontSize = 8

```

```

List11.FontBold = True
List11.ForeColor = RGB(255, 0, 0)
List11.BackColor = RGB(255, 255, 0)
Label12 = "1.200.000.000 TL."
Label13 = "700.000.000 TL."
Label14 = "130.000.000 TL."
Label15 = "90.000.000 TL."
Label16 = "150.000.000 TL."
Label17 = "120.000.000 TL."
End Sub
Private Sub Form_Unload(Cancel As Integer)
Dim a, b, c, d, e, f, g, h, m, n, i
    Open "kurs.dat" For Output As #1
        For i = 0 To List1.ListCount - 1
            a = List1.List(i)
            b = List2.List(i)
            c = List3.List(i)
            d = List4.List(i)
            e = List5.List(i)
            f = List6.List(i)
            g = List7.List(i)
            h = List8.List(i)
            m = List9.List(i)
            n = List10.List(i)
            Write #1, a, b, c, d, e, f, g, h, m, n
        Next
    Close #1
End Sub
Private Sub kaydet_Click()
List1.AddItem Text1
List2.AddItem Text2
List3.AddItem Text3
List4.AddItem Text4
List5.AddItem Text5
If Option1 Then List6.AddItem "Windows 2000"
If Option2 Then List6.AddItem "Programcılık"
If Option3 Then List6.AddItem "Office 2000"
If Option4 Then List6.AddItem "Grafik"
If Option5 Then List6.AddItem "Muhasebe"
If Option6 Then List6.AddItem "İngilizce"
If Option7 Then List7.AddItem "Peşin"
If Option8 Then List7.AddItem "Taksitle"
List8.AddItem Combo1.Text
List9.AddItem Label8
List10.AddItem Label11
End Sub
Private Sub Option1_Click()
    List11.Clear
    List11.AddItem "Windows 2000 Server"
    List11.AddItem "Windows 2000 Professional"
    Label8 = "168 Saat"
End Sub
Private Sub Option2_Click()
    List11.Clear
    List11.AddItem "Visual Basic 6.0"
    List11.AddItem "Delphi"
    List11.AddItem "Visual C"
    List11.AddItem "Java Script"

```

```

List11.AddItem "ASP"
Label8 = "300 Saat"
End Sub
Private Sub Option3_Click()
List11.Clear
List11.AddItem "Windows 2000"
List11.AddItem "Word 2000"
List11.AddItem "Excel 2000"
List11.AddItem "Outlook 2000"
List11.AddItem "Access 2000"
List11.AddItem "Powerpoint 2000"
List11.AddItem "İnternet"
Label8 = "96 Saat"
End Sub
Private Sub Option4_Click()
List11.Clear
List11.AddItem "Corel Draw"
List11.AddItem "Photoshop"
List11.AddItem "Freehand"
Label8 = "60 Saat"
End Sub
Private Sub Option5_Click()
List11.Clear
List11.AddItem "Genel Muhasebe"
List11.AddItem "Eta"
List11.AddItem "Logo"
List11.AddItem "Mikro"
Label8 = "160 Saat"
End Sub
Private Sub Option6_Click()
List11.Clear
List11.AddItem "Yerli ve Yabancı Öğitmenler Eşliğinde"
Label8 = "125 Saat"
End Sub
Private Sub Option7_Click()
Combo1.Enabled = False
Combo1.Text = "-----"
End Sub
Private Sub Option8_Click()
Combo1.Enabled = True
Combo1.ListIndex = 0
End Sub

```

FORM3 Listele ve Sil

10 Listbox, 10 Label, 3 Command Button

```

Private Sub Command1_Click()
Dim kay
kay = MsgBox("Kaydetmek istiyormusunuz", 4)
If kay = 6 Then
Dim a, b, c, d, e, f, g, h, m, n, i
Open "kurs.dat" For Output As #1
For i = 0 To List1.ListCount - 1
a = List1.List(i)
b = List2.List(i)
c = List3.List(i)
d = List4.List(i)

```

```

        e = List5.List(i)
        f = List6.List(i)
        g = List7.List(i)
        h = List8.List(i)
        m = List9.List(i)
        n = List10.List(i)
        Write #1, a, b, c, d, e, f, g, h, m, n
    Next
Close #1
Unload Form3
Else
    Unload Form3
End If
End Sub

```

```

Private Sub Form_Load()
Dim a, b, c, d, e, f, g, h, m, n
If Dir("kurs.dat") <> "" Then
    Open "kurs.dat" For Input As #1
    While Not EOF(1)
        Input #1, a, b, c, d, e, f, g, h, m, n
        List1.AddItem a
        List2.AddItem b
        List3.AddItem c
        List4.AddItem d
        List5.AddItem e
        List6.AddItem f
        List7.AddItem g
        List8.AddItem h
        List9.AddItem m
        List10.AddItem n
    Wend
    Close #1
End If

```

```

End Sub
Private Sub List1_Click()
    List2.ListIndex = List1.ListIndex
    List3.ListIndex = List1.ListIndex
    List4.ListIndex = List1.ListIndex
    List5.ListIndex = List1.ListIndex
    List6.ListIndex = List1.ListIndex
    List7.ListIndex = List1.ListIndex
    List8.ListIndex = List1.ListIndex
    List9.ListIndex = List1.ListIndex
    List10.ListIndex = List1.ListIndex
    List2.TopIndex = List1.TopIndex
    List3.TopIndex = List1.TopIndex
    List4.TopIndex = List1.TopIndex
    List5.TopIndex = List1.TopIndex
    List6.TopIndex = List1.TopIndex
    List7.TopIndex = List1.TopIndex
    List8.TopIndex = List1.TopIndex
    List9.TopIndex = List1.TopIndex
    List10.TopIndex = List1.TopIndex
End Sub

```

```

Private Sub sil_Click()

```

```

Dim sl
If List1.ListIndex < 0 Then
    MsgBox ("Bir eleman seçiniz")
    Exit Sub
Else
    sl = List1.ListIndex
List1.RemoveItem sl
List2.RemoveItem sl
List3.RemoveItem sl
List4.RemoveItem sl
List5.RemoveItem sl
List6.RemoveItem sl
List7.RemoveItem sl
List8.RemoveItem sl
List9.RemoveItem sl
List10.RemoveItem sl
End If
End Sub

```

DriveListBox (Sürücü Listeleme Kutusu)

Sistemdeki sürücü listesini göstermeye yarar.

```

Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
    ChDrive Drive1.Drive
End Sub

```

Directory List Box (Dizin Listeleme Kutusu)

```

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
    ChDir Dir1.Path
End Sub

```

FileList Box (Dosya Listeleme Kutusu)

```

Private Sub Combo1_Click()
    File1.Pattern = Combo1.Text
End Sub

```

```

Private Sub Combo1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        File1.Pattern = Combo1.Text
    End If
End Sub

```

```

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
    ChDir Dir1.Path
    Text1.Text = Dir1.Path + "\" + File1.FileName
End Sub

```

```

Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
    ChDrive Drive1.Drive
    Text1.Text = Dir1.Path + "\" + File1.FileName
End Sub

```

```

Private Sub File1_Click()
    Text1.Text = Dir1.Path + "\" + File1.FileName
End Sub

Private Sub File1_KeyDown(KeyCode As Integer, Shift As Integer)
    Dim ac
    If KeyCode = 13 Then
        ac = Shell(File1.FileName, vbNormalFocus)
    End If
End Sub

Private Sub Form_Load()
    Combo1.AddItem "*.*)"
    Combo1.ListIndex = 0
    Text1.Text = Dir1.Path + "\" + File1.FileName
End Sub

```

PictureBox (Resim Kutusu)

Programınıza resim eklemek için kullanılır.

Align

- | | |
|---|----------------|
| 0 | yok |
| 1 | Üste hizalı |
| 2 | Aşağıya hizalı |
| 3 | Sola hizalı |
| 4 | Sağa hizalı |

Image (Resim Gösterme Kontrolü)

Stretch

Otomatik boyutlandırma.

Common Dialog (Diyalog Pencereleeri)

Microsoft Common Dialog 6.0 Componenti seçilir. Aç, Kaydet, Renk, Help, Yazdır ve Font diyalog kutularının çıkartılmasını sağlar.

```

CommonDialog1.Action = 1  Aç
CommonDialog1.Action = 2  Yeni Adla Kaydet
CommonDialog1.Action = 3  Renk
CommonDialog1.Action = 4  Font
CommonDialog1.Action = 5  Yazdır
CommonDialog1.Action = 6  Help

```

CommonDialog1.Action = 1 Aç

CancelError

Dosya aç kutusunda iptal düğmesine basılınca uyarı vermez.

CancelError = False

Flags

Dosya açma özelliklerini etkiler.

Flags = 1 Aç kutusu üzerinde tek eleman seçilebilir ve salt okunur kutusu seçili.
Flags = 2 Aç kutusu üzerinde tek eleman seçilebilir ve salt okunur kutusu seçili değil.
Flags = 4 Aç kutusu üzerinde tek eleman seçilebilir ve salt okunur kutusu olmaz
Flags = 256 Aç kutusu üzerinde tek eleman seçilebilir.
Flags = 512 Aç kutusu üzerinde birden fazla eleman seçilebilir. Win 3.1 gibi olur.
Flags = 4096 Klasörde olmayan bir dosya açılırsa hata kodu verir.

DialogTitle

Dosya aç kutusunun ismini değiştirir.

DialogTitle = "Dosya Aç"

DefaultExt

Dosya uzantısı belirtilmezse bile otomatik olarak uzantıyı ekler.

DefaultExt = ".dat"

FileName

Dosya aç kutusunda açılmasını istediğiniz dosyalar.

FileName = "*.txt"

Filter

Dosya aç kutusunda açılmasını istediğiniz dosyalar.

Filter= "Text dosyaları (*.txt)|*.txt|Bitmap(*.bmp)|*.bmp|Tüm resimler(*.jpg;*.wmf;*.ico)|*.jpg;*.wmf;*.ico|Tüm dosyalar(*.*)|*.*|"

FilterIndex

Dosya aç kutusunda açılmasını istediğiniz dosyalardan default olarak gelmesini istediğinizi yazarsınız. İlk sıradaki 1 numaradır.

CommonDialog1.FilterIndex = 2

InitDir

Dosya aç kutusunda default olarak açılmasını istediğiniz directory.

CommonDialog1.InitDir = "c:\program files"

CommonDialog1.Action = 2 Kaydet

Flags = 2 Kaydet kutusunda aynı isimli dosyaları kaydederken uyarı verir.

CommonDialog1.Action = 3 Color

Flags = 2 Color kutusunda ayrıntılarda açılır.
Flags = 4 Ayrıntılar gizlenir.

CommonDialog1.Action = 4 Font

Flags = 1 Screen Fontlar gelir.
Flags = 2 Printer Fontlar gelir.
Flags = 3 Tüm Fontlar gelir.
Flags = 257 Font kuusunda özellikli gelir.
Flags = 258 Font kuusunda özellikli gelir.
Flags = 8192 Min-Max arasında bir değer girilmezse uyarı verir.
Flags =131074 Font kutusunda yalnız TTF fontlar gözüktür.
Flags =131330 Font kutusunda yalnız TTF fontlar gözüktür ve özelliklidir.

CommonDialog1.Min = Minimum değer
CommonDialog1.Max = Maksimum değer

CommonDialog1.Action = 5 Print

PrinterDefault = True ise varsayılan printerin ayarları aktif olur.
Copies = Kopya sayısı verilir.

Flags = 1 Yazdır kutusunda Seçili seçili olur.
Flags = 2 Yazdır kutusunda sayfalar seçili olur.
Flags = 4 Yazdır kutusunda Seçili kısım pasif olur.
Flags = 8 Yazdır kutusunda Sayfalar pasif olur ve harmanla açılır.
Flags = 16 Yazdır kutusunda Tümü seçili olur ve harmanlama şekli açılır.
Flags = 32 Yazdır kutusunda Dosyaya yazdır seçili olur.
Flags = 64 Yazdır kutusunda Yazıcı ayarları görünür.
Flags = 2048 Yazdır kutusunda Help menüsünde görünür.
Flags = 524288 Yazdır kutusunda Dosyaya yazdır kutusu pasif olur.
Flags = 1048576 Yazdır kutusunda Dosyaya yazdır ekranda olmaz.

Orientation = 1 Dikey 2 Yatay olur.

Örnek 1

Aç,Kaydet, Renk , Print, Font

```
Private Sub ac_Click()  
On Local Error Goto hata  
Dim a  
CommonDialog1.Filter = "metin|*.txt|"  
CommonDialog1.ShowOpen  
Open CommonDialog1.FileName For Input As #1  
Text1.Text = ""  
  
While Not EOF(1)  
Line Input #1, a  
Text1 = Text1.Text + a + Chr(13) + Chr(10)  
Wend  
Close #1  
Exit Sub  
Hata:  
Exit Sub  
  
End Sub  
  
Private Sub font_Click()  
CommonDialog1.FontName = Text1.FontName
```

```

CommonDialog1.FontSize = Text1.FontSize
CommonDialog1.FontBold = Text1.FontBold
CommonDialog1.FontItalic = Text1.FontItalic
CommonDialog1.FontUnderline = Text1.FontUnderline
CommonDialog1.FontStrikethru = Text1.FontStrikethru
CommonDialog1.Color = Text1.ForeColor
'Bu kodları font CommonDialog1 açıldığında
'Text kutusunun özelliklerinin otomatik olarak gelmesini sağlar.

```

```

CommonDialog1.Flags = 257
CommonDialog1.ShowFont

```

```

Text1.FontName = CommonDialog1.FontName
Text1.FontSize = CommonDialog1.FontSize
Text1.FontBold = CommonDialog1.FontBold
Text1.FontItalic = CommonDialog1.FontItalic
Text1.FontUnderline = CommonDialog1.FontUnderline
Text1.FontStrikethru = CommonDialog1.FontStrikethru
Text1.ForeColor = CommonDialog1.Color

```

End Sub

```

Private Sub color_Click()
CommonDialog1.Flags = cdICCPreventFullOpen
CommonDialog1.ShowColor
Text1.BackColor = CommonDialog1.Color
End Sub

```

```

Private Sub print_Click()
CommonDialog1.Flags = &H40
CommonDialog1.ShowPrinter
Printer.Print Text1
Printer.EndDoc
End Sub

```

```

Private Sub kay_Click()
Dim a
CommonDialog1.DialogTitle = "Kaydet"
CommonDialog1.DefaultExt = ".txt"
CommonDialog1.Flags = 2
CommonDialog1.ShowSave
Open CommonDialog1.FileName For Output As #1
a = Text1
Print #1, a
Close #1
End Sub

```

MSMasked (Formatlı Giriş)

Microsoft Masked Edit Control 6.0

Bu kontrol sayesinde formlarınızda özel formatlı telefon, tarih, saat gibi verileri hızlı bir şekilde girmesi sağlanır.

Mask

Kullanıcının ne şekilde veri girileceği belirlenir.

Örnek

Telefon (###) ### ## ##	(216) 455 55 55
Telefon #.###.### ## ##	0.216.455 55 55
Tarih ##-??-####	12-Ara-2000
Tarih ##-##-##	12-12-00
Saat ##:## ??	09:11 PM
Saat ##:##	09:11
Rakam ###	
Harf ???	

Appearance

MaskEdit textin 3D ve normal gözükmelerini sağlar.

AutoTab

Kullanıcının veri girişini tamamlamasının ardından setfocus un diğer komponente geçmesini sağlar.

Text

Kullanıcının ne şekilde veri girdiyse o şekilde alınmasını sağlar.

ClipText

Kullanıcının ne şekilde veri girilmesi önemli değildir. Veriler mask özelliği olmadan görünür.

ClipMode

0 verilirse kopyalama işlemlerinde text özelliğini
1 verilirse kopyalama işlemlerinde cliptext özelliğini alır.

Format

Texte yazılan bilgilerin ekranda görünme şekillerini değiştirir.

Para	#,##0.00TL
Tam sayı	0
Basamaklı sayı	#,##0
Yüzde sayı	%0
Üstlü sayı	0.00E+00
Tarih,saat	c
Tarih gg/aa/yyyy	ddddd
Tarih gün gg ay yyyy	ddddd
Tarih gg-ay-yy	dd-mmm-yy
Saat 24 saat	hh:mm
Saat 12 saat AM/PM	hh:mm AM/PM
Saat sa:dk:sn	tttt

MaxLength

Maskedit texte en fazla 64 karakter bilgi girilebilir.

PromptChar

MaskEdit texte yazılan mask aralarına konan _ işareti değiştirilebilir.

PromptInclude

PromptChar karakterinin gizlenmesi sağlanabilir.

Örnek 1

```
Private Sub tarih_LostFocus()  
    If IsDate(tarih) = False Then  
        MsgBox ("Lütfen uygun tarih giriniz")  
        tarih.Mask = ""  
        tarih.Text = ""  
        tarih.Mask = "##/##/####"  
        tarih.SetFocus  
    End If  
End Sub
```

SSTab

Microsoft Tabbed Dialog Control 5.0

Bu kontrol sayesinde seperatör kullanmamızı sağlar. Tab üzerine bir komponent yerleştirmek için Frame kullanılır.

Tabs

Tabs üzerinde bulunacak tabs sayısı belirlenir. En fazla 99 olur.

Tab

Program çalışınca hangi Tabın aktif olarak geleceği belirlenir.

TabHeight

Tabın başlık kısmının yüksekliğini ayarlar.

TabMaxWidth

Tabın başlık kısmının genişliğini ayarlar.

TabOrientation

4 özellik alır.

- | | |
|---|--------------------|
| 0 | Tablar üstte |
| 1 | Tablar altta |
| 2 | Tablar solda |
| 3 | Tablar sağda olur. |

TabsPerRow

Tab üzerinde bir satırda yerleştirilecek eleman sayısını belirlenir.

TabEnabled(index)

Program çalışınca herhangi bir Taba ulaşılmaması için kullanılır.

```
Private Sub Form_Load()  
    SSTab1.TabEnabled(3) = False  
End Sub
```

Picture

Tabın baş tarafına resim koymak için kullanılır.

PreviousTab

Bir Tabdan başka bir taba geçildiği zaman hangi tabdan geçildiğini bulmak için kullanılır.

Style

0 olunca tabın otomatik veya manuel olur.

1 olunca tabın genişliği sabit olur.

ShowFocus Rect

True olunca program çalışınca Tabın üzerine gelince caption etrafı focuslu olur.

WordWrap

Tabın üzerindeki captiondaki yazının kaymasını sağlar.

Richtextbox

Microsoft Rich Textbox Control 6.0

Bu kontrol sayesinde zenginleştirilmiş text özelliklere sahiptir.

SelChange

RichTextBox üzerinde kursorun yer değişmesi sonucu oluşan olaylar.

Örnek 1

```
Private Sub Chcbold_Click()  
    RichTextBox1.SelBold = Chcbold.Value  
End Sub
```

```
Private Sub chccitalic_Click()  
    RichTextBox1.SelItalic = chccitalic.Value  
End Sub
```

```
Private Sub chcunder_Click()  
    RichTextBox1.SelUnderline = chcunder.Value  
End Sub
```

```
Private Sub Command1_Click()  
    CommonDialog1.Flags = 258  
    CommonDialog1.ShowFont  
    RichTextBox1.SelFontName = CommonDialog1.FontName  
    RichTextBox1.SelFontSize = CommonDialog1.FontSize  
    RichTextBox1.SelUnderline = CommonDialog1.FontUnderline  
    RichTextBox1.SelStrikeThru = CommonDialog1.FontStrikethru  
    RichTextBox1.SelColor = CommonDialog1.Color  
    RichTextBox1.SelBold = CommonDialog1.FontBold  
    RichTextBox1.SelItalic = CommonDialog1.FontItalic  
End Sub
```

```
Private Sub Form_Load()
```

```
RichTextBox1.TextRTF = "En Büyük Galatasaray Başka Büyük Yok"
End Sub
```

```
Private Sub oportu_Click()
    If oportu Then RichTextBox1.SelAlignment = 2
End Sub
```

```
Private Sub opsag_Click()
    If opsag Then RichTextBox1.SelAlignment = 1
End Sub
```

```
Private Sub opsol_Click()
    If opsol Then RichTextBox1.SelAlignment = 0
End Sub
```

Örnek 2

Daktilo gibi yazan program.

```
Option Explicit
Dim i As Double
```

```
Private Sub Command1_Click()
    Dim a
    CommonDialog1.Filter = "metin|*.txt|"
    CommonDialog1.Flags = 259
    CommonDialog1.ShowOpen
    Open CommonDialog1.FileName For Input As #1
    Text1.Text = ""
```

```
While Not EOF(1)
    Line Input #1, a
    rt1.Text = rt1.Text + a + Chr(13) + Chr(10)
Wend
Close #1
Timer1.Enabled = True
```

```
End Sub
```

```
Private Sub Form_Load()
    Timer1.Interval = 10
    Timer1.Enabled = False
End Sub
```

```
Private Sub Text2_Change()
    Dim a
    a = Text2.SelStart
    Text2 = Format(Text2, "###,###")
    Text2.SelStart = a + 1
End Sub
```

```
Private Sub Timer1_Timer()
    Text1.Text = Left(rt1.Text, i): Beep
    i = i + 1
    If i = Len(rt1.Text) Then Timer1.Enabled = False
End Sub
```

Popup Menü

PopupMenu menü adı, flags, x, y, Defaultmenu

Farenin sağ klick yapılıncı oluşan menüye denir.

X,Y

Menünün ekranda çıkacağı koordinatlar.

Flags

- 0 Farenin sol tuşuyla eleman seçilecek ve menü x koordinatının sağında açılacak.
- 2 Farenin sağ tuşuyla eleman seçilecek.
- 4 Menü x koordinatını ortalayacak
- 8 Menü x koordinatının solunda çıkacak

DefaultMenü

Menü açıldığı zaman bir parametre bold çıkartılabilir.

Menü editöre girilir ve orada bir menü oluşturulur. Menülerin Nameleri yazılırken sonra şaşırmamak için önlerine aynı harfleri koyun (oomenü gibi) ve Visible= False olsun

Örnek 1

Menü isimleri

Kopyala

Yapıştır

Kes

Sil

---- - ` Düz çizgi koymak için kullanılır.

Font

Zemin Rengi olsun.

Private Sub Form_Load()

Label1.Caption = X

Label2.Caption = Y

End Sub

Private Sub mnfont_Click()

On Error GoTo hata

With CommonDialog1

.Flags = 259

.ShowFont

End With

If CommonDialog1.FontName <> "" Then ActiveControl.FontName = CommonDialog1.FontName

ActiveControl.FontSize = CommonDialog1.FontSize

ActiveControl.ForeColor = CommonDialog1.Color

ActiveControl.FontUnderline = CommonDialog1.FontUnderline

ActiveControl.FontBold = CommonDialog1.FontBold

ActiveControl.FontItalic = CommonDialog1.FontItalic

hata: Exit Sub

End Sub

Private Sub mnkes_Click()

Clipboard.SetText ActiveControl.SelText


```

        ActiveControl.SelText = ""
End Sub

Private Sub mnkopyala_Click()
    Clipboard.SetText ActiveControl.SelText
End Sub

Private Sub mnsil_Click()
    ActiveControl.SelText = ""
End Sub

Private Sub mnyapistir_Click()
    ActiveControl.SelText = Clipboard.GetText
End Sub

Private Sub mnzemin_Click()
On Error GoTo hata
    With CommonDialog1
        .CancelError = True
        .ShowColor
    End With
    ActiveControl.BackColor = CommonDialog1.Color
Exit Sub
hata:
    Exit Sub
End Sub

Private Sub text1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button = 4 Then PopupMenu mnmenu, 4, ActiveControl.Left, ActiveControl.top, mnkopyala
End Sub

Private Sub Text1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Label1.Caption = X
    Label2.Caption = Y
End Sub

Private Sub Text2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button = 4 Then PopupMenu mnmenu, 4, ActiveControl.Left, ActiveControl.top, mnkopyala
End Sub

Private Sub oozrengi_Click()
    CommonDialog1.ShowColor
    Text1.BackColor = CommonDialog1.Color
End Sub

```

MsFlexGrid

Microsoft FlexGrid Control (MsflxGrd.OCX)

Cols

Grid üzerinde oluşturulacak Kolon sayısı.

ColAlignment(index)

Grid üzerindeki aktif kolonun alignment özelliğini belirler.

- 0 Sola, üste yanaşık
- 1 Sola, orta yanaşık
- 2 Sola, alta yanaşık
- 3 Orta, üste yanaşık
- 4 Orta, orta yanaşık
- 5 Orta, alta yanaşık
- 6 Sağ, üst yanaşık
- 7 Sağ, orta yanaşık
- 8 Sağ, alta yanaşık
- 9 Rakamlar sağ-orta ve metinler sol-orta yanaşık

Dim i

```
MSFlexGrid1.Rows = 15  
MSFlexGrid1.Cols = 15
```

```
MSFlexGrid1.Col = 0  
For i = 0 To MSFlexGrid1.Rows - 1  
    MSFlexGrid1.Row = i  
    MSFlexGrid1.Text = i  
    MSFlexGrid1.RowHeight(i) = 500  
Next  
MSFlexGrid1.ColAlignment(0) = 4
```

```
MSFlexGrid1.Row = 0  
For i = 1 To MSFlexGrid1.Cols - 1  
    MSFlexGrid1.FixedAlignment(i) = 4  
    MSFlexGrid1.Col = i  
    MSFlexGrid1.Text = Chr(64 + i)  
    MSFlexGrid1.ColWeight(i) = 1500  
Next
```

ColWidth(index)

Grid üzerindeki kolonların genişliğini belirler.

FillStyle

- 0 Değişiklikler tek hücreyi etkiler.
- 1 Değişiklikler tüm gridi etkiler.

FixedCols, Rows

Grid üzerinde normalde ilk satır ve sütuna veri girişi yapılmaz. İstenirse bu sayı arttırılabilir.

FixedAlignment(index)

Grid üzerindeki fixed kolonun alignment özelliğini belirler.

GridLines

Grid üzerinde çizgilerin özelliklerini değiştirir.

GridLineWidth

Grid üzerinde çizgilerin kalınlığını ayarlar.

GridLinesFixed

Grid üzerinde çizgilerin özelliklerini değiştirir.

HighLight

Grid üzerindeki aktif hücre üzerindeki taramayı kapatır.

Rows

Grid üzerindeki satır sayısını ayarlar.

RowHeight(index)

Grid üzerindeki satırların yüksekliğini belirler.

Col, Row

Grid üzerinde o andaki aktif hücreyi belirtir.

Text

Grid üzerindeki hücreye yazılacak yazıyı belirler.

Clip

Grid üzerindeki hücelere birden veri atamak için kullanılır. Chr(9) ve Chr(13) kullanılabilir.

ColSel

Seçilen alanların ilk kolon değerini verir. MouseMove yazılır.

RowSel

Seçilen alanların son satır değerini verir. MouseMove yazılır.

TextMatrix

Grid üzerinde koordinat vererek yazı yazmak için kullanılır.

MSFlexGrid1.TextMatrix(1, 1) = "Oguz"

Sort

Grid üzerinde sıralama yapmak için kullanılır. Seçili kısım varsa o kısımda uygulanır. Seçilmiş alan yoksa aktif sütunda yapılır.

- | | |
|---|--|
| 0 | Sıralama yok |
| 1 | Artan sıra |
| 2 | Azalan sıra |
| 3 | Artan sayısal sıralama |
| 4 | Azalan sayısal sıralama |
| 5 | Büyük-küçük harf ayrımı yapmayan artan sıralama |
| 6 | Büyük-küçük harf ayrımı yapmayan azalan sıralama |
| 7 | Büyük-küçük harf ayrımı yapan artan sıralama |
| 8 | Büyük-küçük harf ayrımı yapan azalan sıralama |
| 9 | özel |

MergeCells

Grid üzerinde hücreleri birleştirmek için kullanılır.

```
Private Sub MSFlexGrid1_KeyPress(KeyAscii As Integer)
    Text1 = MSFlexGrid1.Text
    Text1.SetFocus
    Text1.SelStart = Len(Text1)
    SendKeys Chr(KeyAscii)
End Sub
Private Sub Text1_Change()
    MSFlexGrid1.Text = Text1
End Sub
Dim a
MSFlexGrid1.MergeCells = 1
For a = 0 To MSFlexGrid1.Cols - 1
    MSFlexGrid1.MergeCol(a) = True
Next
```

ColPosition, RowPosition

Grid üzerinde index numarası verilen satır veya sütunların yerini değiştirir .

```
MSFlexGrid1.ColPosition(2) = 3
```

ColFont

Grid üzerindeki aktif hücrenin font özelliklerini değiştirir.

CellFont, CellFontSize

Grid üzerindeki seçilmiş hücrenin font özelliklerini değiştirir.

Additem

Grid üzerine veri eklemek için kullanılır.

```
'Satır Ekle
MSFlexGrid1.AddItem ""

MSFlexGrid1.AddItem "",1 'Birinci satırın sonuna ekler.
```

Removeitem

Grid üzerine veri silmek için kullanılır.

Clear

Grid üzerine veri silmek için kullanılır.

```
MSFlexGrid1.Clear
```

```
Private Sub ac_Click()
Dim a
If Dir("oguz.xls") <> "" Then
    Open "oguz.xls" For Input As #1
```

```

        MSFlexGrid1.Col = 0
        MSFlexGrid1.Row = 1
        MSFlexGrid1.ColSel = MSFlexGrid1.Cols - 1
        MSFlexGrid1.RowSel = MSFlexGrid1.Rows - 1
    Input #1, a
        MSFlexGrid1.Clip = a
    Close #1
End If
End Sub

Private Sub Command1_Click()
    MSFlexGrid1.ColPosition(2) = 3
End Sub

Private Sub Command2_Click()
'Satır Ekle
    MSFlexGrid1.AddItem "" : End Sub
Private Sub Command3_Click()
    MSFlexGrid1.Clear
End Sub

Private Sub font_Click()
MSFlexGrid1.FillStyle = 1
CommonDialog1.Flags = 259
CommonDialog1.ShowFont
    MSFlexGrid1.CellFontName = CommonDialog1.FontName
    MSFlexGrid1.CellFontSize = CommonDialog1.FontSize
    MSFlexGrid1.CellFontBold = CommonDialog1.FontBold
    MSFlexGrid1.CellForeColor = CommonDialog1.Color
    MSFlexGrid1.CellFontStrikeThrough = CommonDialog1.FontStrikethru
    MSFlexGrid1.CellFontUnderline = CommonDialog1.FontUnderline

End Sub

Private Sub Form_Load()
Dim i
    MSFlexGrid1.Cols = 4
    MSFlexGrid1.Rows = 10
    MSFlexGrid1.FixedCols = 0
    MSFlexGrid1.TextMatrix(0, 0) = "Adı"
    MSFlexGrid1.TextMatrix(0, 1) = "Soyadı"
    MSFlexGrid1.TextMatrix(0, 2) = "Telefon"
    MSFlexGrid1.TextMatrix(0, 3) = "Adresi"

End Sub

Private Sub kaydet_Click()
Dim a
    Open "oguz.xls" For Output As #1
        MSFlexGrid1.Col = 0
        MSFlexGrid1.Row = 1
        MSFlexGrid1.ColSel = MSFlexGrid1.Cols - 1
        MSFlexGrid1.RowSel = MSFlexGrid1.Rows - 1
        a = MSFlexGrid1.Clip
    Write #1, a
    Close #1
End Sub

```

```

Private Sub MSFlexGrid1_KeyPress(KeyAscii As Integer)
    Text1 = MSFlexGrid1.Text
    Text1.SetFocus
    Text1.SelStart = Len(Text1)
    SendKeys Chr(KeyAscii)
End Sub

```

```

Private Sub MSFlexGrid1_Click(KeyAscii As Integer)
    Text1 = MSFlexGrid1.Text
    Text1.SelStart = Len(Text1)
End Sub

```

```

Private Sub MSFlexGrid1_EnterCell(KeyAscii As Integer)
    Text1 = MSFlexGrid1.Text
    Text1.SelStart = Len(Text1)
End Sub

```

```

Private Sub sirala_Click()
    Dim a
    MSFlexGrid1.MergeCells = 1
    For a = 0 To MSFlexGrid1.Cols - 1
        MSFlexGrid1.MergeCol(a) = True
    Next
    For a = 0 To MSFlexGrid1.Rows - 1
        MSFlexGrid1.MergeRow(a) = True
    Next
End Sub

```

```

Private Sub Text1_Change()
    MSFlexGrid1.Text = Text1
End Sub

```

```

Private Sub zrenji_Click()
    MSFlexGrid1.FillStyle = 1
    CommonDialog1.ShowColor
    MSFlexGrid1.CellBackColor = CommonDialog1.Color
End Sub

```

```

Private Sub sil_Click()
    Dim a,b,c,i
    a=MSFlexGrid1.Row
    b=MSFlexGrid1.Row
    if a > b Then
        c=a
        a=b
        b=c
    End if
    For i=a To b
        MSFlexGrid1.RemoveItem a
    Next
End Sub

```

ProgressBar (İlerleme Çubuğu)

Yapılan işlemlerin ilerleme durumunu gösterir.

Min, Max

ProgressBar'ın alacağı değerler yazılır.

Orientation

ProgressBar'ın dikey veya yatay olması sağlanır.

Scrolling

ProgressBar'ın üzerindeki artışlar kutu kutu veya düz gözüktür.

Value

ProgressBar'ın artma oranı.

Örnek 1

```
Private Sub Command1_Click()  
    Timer1.Enabled = True  
    Timer1.Interval = 1000  
End Sub  
  
Private Sub Timer1_Timer()  
    ProgressBar1.Value = ProgressBar1.Value + 5  
    If ProgressBar1.Value > 99 Then  
        MsgBox ("Yükleme tamamlandı")  
        Timer1.Enabled = False  
    End If: End Sub
```

Örnek 2

```
On Local Error GoTo hata  
Static a, b  
For a = 1 To 10000  
    ProgressBar1.Value = ProgressBar1.Value + b  
    If ProgressBar1.Value >= 9999 Then  
        MsgBox ("Yükleme tamamlandı")  
        b = a + b  
    End If  
hata:  
Exit Sub  
Next
```

Örnek 3

```
Private Sub Command1_Click()  
    ProgressBar1.Value = 0  
    ProgressBar2.Value = 0  
    Timer1.Enabled = True  
    Timer1.Interval = 100  
End Sub  
  
Private Sub Timer1_Timer()  
    If ProgressBar1.Value >= ProgressBar1.Max Then  
        ProgressBar1.Value = 0
```

```

Else
    ProgressBar1.Value = ProgressBar1.Value + 10
    If ProgressBar1.Value = ProgressBar1.Max Then
        ProgressBar2.Value = ProgressBar2.Value + 10
        If ProgressBar2.Value = ProgressBar1.Max Then
            MsgBox ("Yükleme Tamamlandı")
            Timer1.Enabled = False
            Exit Sub
        End If
    End If
End If
End If
End Sub

```

Slider

Microsoft Windows Common Controls 6.0 (MSCOMCTL.OCX)

Ekrana yerleştirilen herhangi bir komponentin özelliklerini değiştirmede kullanılır.

LargeChange

Mouse ile Slider üzerine tıkladığımız zaman ne kadarlık bir değişim olacağı belirlenir .

Max

Slider in alacağı en fazla değer.

Min

Slider in alacağı en düşük değer.

Orientation

0	Yatay
1	Dikey

SmallChange

Slider in üzerinde ok tuşlarıyla ne kadarlık bir değişim olacağını belirler.

TextPosition

0	Yazılar Slider in üstünde.
1	Yazılar Slider in altında.

TickFrequency

Slider in min-max arasında kaç adet çizgi olacağını belirler.

TickStyle

Slider in üzerindeki çizgilerin stillerini belirler.

0	Çizgiler altta
1	Çizgiler üstte
2	Çizgiler iki tarafta
3	Çizgi yok.

Value

Slider üzerinde oluşan deęişim miktarı.

[DateTimePicker](#)

Ekrana bir tarih gösterici koymak için kullanılır.

[CheckBox](#)

DateTimePicker üzerine bir checkbox koyarak kullanıcının tarihi onaylamasını sağlayabilirsiniz. Eğer check işaretlenmezse tarih boş olarak döner.

[Format](#)

- 0 Uzun tarih
- 1 Kısa tarih
- I. Saat
- II.

[MinDate, MaxDate](#)

Minimum ve Max tarih girilir.

[UpDown](#)

Eğer bu seçilirse Combobox gibi gözükmeyen sadece updown larla deęişim olabilir.

[Örnek 1](#)

```
Private Sub DTPicker1_CallbackKeyDown(ByVal KeyCode As Integer, ByVal Shift As Integer, ByVal  
CallbackField As String, CallbackDate As Date)  
Text1.Text = DTPicker1.Value  
End Sub
```

[MonthView](#)

Ekrana bir takvim koymak için kullanılır. Mouse ile ay üzerine tıklanırsa ay seçmek için otomatik bir liste çıkar. Yıl üzerine tıklanırsa bir updown çıkar.

[MaxSelCount](#)

Multiselect özellięi true olan bir monthiew de en fazla kaç adet tarih seçilebileceęini belirtmek için kullanılır.

[MonthRows, MonthColumns](#)

Ekranda kaç aylık Monthiew olacaęını belirtmek için kullanılır.

[ShowWeekNumber](#)

Haftanın numaraları gösterilir.

[StartOfWeek](#)

Haftanın başlangıç günü ayarlanır.

[StartOfWeek](#)

Haftanın başlangıç günü ayarlanır.

```
Private Sub MonthView1_DateClick(ByVal DateClicked As Date)
    Text1.Text = DateClicked
End Sub
```

```
Private Sub MonthView1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Text1.Text = MonthView1.SelStart & "-" & MonthView1.SelEnd
End Sub
```

Coolbar

Microsoft Windows Common Controls-3 6.0 (COMCT332.OCX)

Ekrana toolbar yerleştirmek için kullanılır.

- I. Ekrana bir tane ImageList1 eklenir ve içerisine koymak istediğiniz resimler yerleştirilir.
- II. Coolbar üzerine bir tane Toolbar yerleştirilir.
- III. Toolbar üzerinde sağ klik yapılır ve ButtonHeight = 330 yapılır.
- IV. ImageList eklenir.
- V. Button sekmesinden butonlar eklenir.
- VI. Coolbar properties. Bands sekmesinden.
- VII. Child bölümüne Toolbar eklenir.

Bands sekmesinden.

InsertBand

Coolbar üzerine bant eklemek için kullanılır.

FixedOrder

Coolbarın boyutlarıyla oynanmasını istemiyorsanız burası işaretlenir.

RemoveBand

Coolbar üzerinden bant silmek için kullanılır.

MinWidth

Coolbarın en fazla küçülebileceği mesafe.

MinHeight

Coolbarın yüksekliği.

Key

Kod yazarken kullanılacak ismi.

Image

Coolbarın başında bulunan resim.

NewRow

Coolbarın üzerine yeni bir row ekler.

NewRow

Coolbarın üzerine yeni bir row ekler.

Toolbar

Ekrana toolbar yerleřtirmek için kullanılır. Öncelikle ekrana bir tane toolbar yerleřtirilir ve sađ klik yapılarak properties ine girilir. Toolbara sonradan resim yerleřtirmek içinde bir tane imagelist konulur.

ImageList

Toolbar üzerine yerleřtirilecek resimlerin listesi eklenir.

Button sekmesine geçilir.

InsertButton

Toolbar üzerine buton eklemek için kullanılır.

RemoveButton

Toolbar üzerinden buton silmek için kullanılır.

Index

Toolbar üzerindeki butonları temsil eder.

Key

Toolbar üzerinde hangi butona basıldıđını anlamak için buraya name yazılır.

Style

Toolbar üzerindeki butonun özelliđini belirtir.

- | | |
|---|---|
| 0 | Normal |
| 1 | Check Butonu |
| 2 | Option Buton |
| 3 | Seperatör Araya boşluk koyar. (Frame gibi davranır) |
| 4 | Seperatör gibidir. Boşluk oranını biz verebiliriz. |
| 5 | Aşađı açılan menü |
| | İnsert buton yaparak alt menüler oluşturulur. |

Wrappable

Toolbar üzerindeki butonlar forma sığmadıđı zaman düğmeler bir alt satırdan devam eder.

Statusbar

Ekranın altına açıklama barı yerleřtirmek için kullanılır.

General sekmesinden.

Style

Status barın görünümünü belirler.

Style

Status barın görünümünü belirler.

- | | |
|---|--------------------|
| 0 | Çoklu panel |
| 1 | Bir panel bulunur. |

SimpleText

Style özelliği 1' ken barın üzerindeki yazıyı görmemizi sağlar.

Panels Sekmesinden

Index

Status barın üzerine panel eklememize yarar max. 16 adet.

Text

Status barın index numarasına göre panellere text yazmamızı sağlar.

Key

Status barın panellerini index numarasına göre değilde ismine göre kullanmak istediğimizde kullanılır.

Style

Status barın panellerine otomatik olarak komut koymak için kullanılır.

- | | |
|---|------------------|
| 0 | Text veya resim |
| 1 | Caps Lock tuşu |
| 2 | Num Lock tuşu |
| 3 | İnsert tuşu |
| 4 | Scroll Lock tuşu |
| 5 | Saati gösterir. |
| 6 | Tarihi gösterir. |

Bevel

Status barın panellerinin görünümünü belirler.

- | | |
|---|----------------|
| 0 | Düz |
| 1 | İçeri basık |
| 2 | Dışarıya doğru |

AutoSize

Status barın panellerinin boyut görünümünü belirler.

- | | |
|---|---|
| 0 | Boyutlandırma yok |
| 1 | Form resize olduğunda paneller oluşan yeni mesafeyi paylaşır. |
| 2 | Paneller metinlerini gösterecek şekilde genişler. |

Picture

Status barın panellerine resim koymak için kullanılır.

Örnek 1

```

Private Sub Combo1_Click()
    RichTextBox1.SelFontName = Combo1.Text
End Sub

Private Sub Combo2_Click()
    RichTextBox1.SelFontSize = Combo2.Text
End Sub

Private Sub Form_Load()
    Dim i
    For i = 0 To Screen.FontCount - 1
        Combo1.AddItem Screen.Fonts(i)
    Next
    'Combo1.Sorted = True
    Combo1.ListIndex = 5

    For i = 6 To 72 Step 2
        Combo2.AddItem (i)
    Next
    Combo2.ListIndex = 2
    RichTextBox1.SelBullet = True
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
    Select Case Button.Key
        Case "bld"
            RichTextBox1.SelBold = Button.Value
        Case "italik"
            RichTextBox1.SelItalic = Button.Value
        Case "altcizgi"
            RichTextBox1.SelUnderline = Button.Value
        Case "sol"
            RichTextBox1.SelAlignment = 0
        Case "orta"
            RichTextBox1.SelAlignment = 2
        Case "sag"
            RichTextBox1.SelAlignment = 1
    End Select
End Sub

Private Sub Toolbar2_ButtonClick(ByVal Button As MSComctlLib.Button)
    Select Case Button.Key
        Case "ac"
            CommonDialog1.FileName = "*.txt"
            CommonDialog1.ShowOpen
            Open CommonDialog1.FileName For Input As #1
            RichTextBox1.TextRTF = ""
            While Not EOF(1)
                Line Input #1, a
                RichTextBox1.Text = RichTextBox1.Text + a + Chr(13) + Chr(10)
            Wend
            Close #1

        Case "kaydet"
            CommonDialog1.Flags = 2
            CommonDialog1.DefaultExt = ".txt"
            CommonDialog1.ShowSave
            Open CommonDialog1.FileName For Output As #1

```

```

a = RichTextBox1.Text
Print #1, a
Close #1

Case "kes"
Clipboard.SetText RichTextBox1.SelRTF
RichTextBox1.SelText = ""

Case "koya"
Clipboard.SetText RichTextBox1.SelRTF

Case "yap"
RichTextBox1.SelRTF = Clipboard.GetText

Case "yaz"
CommonDialog1.ShowPrinter
Printer.Print RichTextBox1.Text
Printer.EndDoc

End Select
End Sub

Private Sub rtfText_SelChange()
fMainForm.tbToolBar.Buttons("Bold").Value = IIf(rtfText.SelBold, tbrPressed, tbrUnpressed)
fMainForm.tbToolBar.Buttons("Italic").Value = IIf(rtfText.SelItalic, tbrPressed, tbrUnpressed)
fMainForm.tbToolBar.Buttons("Underline").Value = IIf(rtfText.SelUnderline, tbrPressed,
tbrUnpressed)
fMainForm.tbToolBar.Buttons("Align Left").Value = IIf(rtfText.SelAlignment = rtfLeft, tbrPressed,
tbrUnpressed)
fMainForm.tbToolBar.Buttons("Center").Value = IIf(rtfText.SelAlignment = rtfCenter, tbrPressed,
tbrUnpressed)
fMainForm.tbToolBar.Buttons("Align Right").Value = IIf(rtfText.SelAlignment = rtfRight, tbrPressed,
tbrUnpressed)
End Sub

```

Veri Tabanı Uygulamaları

Visual Data Manager

Add-Ins den Visual Data Manager seçilir.

New – Microsoft Access – Version 7.0 MDB

Bir isim verilir. **(üye takibi)**

Properties üzerindeyken sağ klik yapılır ve New Table seçilir.

Table Name : (Bilgi_Girisi)

Add Field seçilerek tabloya girilecek veriler girilir, **Remove Field** seçilerek tablodan veri silinir. ve **Built The Table** basılır.

Name	:	'verilere girilecek isim
Type	:	'verilerin typeleri
Size	:	'verinin uzunluğu kaç verilirse o sayıdan fazla bilgi girilemez
FixedField	:	'seçili olursa random dosyada olduğu gibi her veriye aynı uzunluk ayırır.
VariableField	:	'seçili olursa veriler kendi uzunluklarını alır.
AutoIncField	:	'seçili ise sayılar otomatik olarak artar.

AllowZeroLength : 'seçili ise herhangi bir veri girilmezse bile işlem devam eder.
Required : 'seçili ise herhangi bir veri girilmezse işlem devam etmez. Kullanıcıyı zorlar.
OrdinalPosition : 'veri alanı sırası girilir.
ValidationText : 'veri alanına girilecek bilgi hakkında ValidationRule 'ye göre açıklama yazılır.
ValidationRule : 'veri alanına girilebilecek yöntem girilir. Text kutusundaki Validate olayı.
DefaultValue : 'Inputbox'daki varsayım değeri gibi çalışır.

Field

Name	Tupe	Size
uyeno	Integer	2
Ad	Text	25
soyad	Text	30
Uye_turu	Text	15
unvan	Text	10
telefon	Text	15
Faks	Text	10
Email	Text	30
adres	Memo(32Kb bilgi)	0
sehir	Text	20
pkodu	long	4

Add Index tablolar arasında ilişki kurmak, tablomuzda arama yaptırmak için kullanılır ve sadece bir tanesi birincil anahtar (**Primary**) olabilir. Ayrıca bir veri üzerinde aynı veriden olmasını istemiyorsanız (örneğin aynı üye numarasından olmasını istemiyorsak) **Unique** işaretlenir

Otomatik Form Hazırlama

Utility sekmesinden Data Form Designer seçilir. Record Source sekmesinden table seçilir ve From Name bölümüne de bir isim verilir. Built the Form seçilerek tablo oluşturulur.

Data Environment ile Form Hazırlama

Add menüsünden More ActiveX Designers, Data Environment seçeneği ile ekrana bir tane Data Environment eklenir. (Eğer gözükmüyorsa Project menüsünden references den seçilir.) Connection1 üzerindeyken sağ klik yapılarak Properties ine girilir. Microsoft 3.51 veya 4.0 seçilir. Next'e basılır. 1 numaralı bölümden mdb dosyamız bulunur ve güvenli olması için Test Connection yapılır. Tamam diyerek oradan çıkılır. View menüsünden Data View Window açılır.

Data View Window üzerinde istediğimiz table ları Data Environment mouse yardımıyla taşıyız ve Data View Window kapatılır. Data Environment üzerinden istediğimiz elemanlar bu form üzerine yerleştirilir.

```
Private Sub Onceki_Click()
    If DataEnvironment1.rsBilgiGirisi.BOF Then
        MsgBox "İlk kayıt"
    Else
        DataEnvironment1.rsBilgiGirisi.MovePrevious
    End If
End Sub
```

```
Private Sub Sonraki_Click()
    If DataEnvironment1.rsBilgiGirisi.EOF Then
        MsgBox "İlk kayıt"
    Else
        DataEnvironment1.rsBilgiGirisi.MoveNext
    End If
End Sub
```

End If
End Sub

Veri Tabanları İşlemleri

Data1.Recordset.MoveNext	Sonraki kayda götürür.
Data1.Recordset.MovePrevious	Önceki kayda götürür.
Data1.Recordset.AddNew	Yeni kayıt oluşturur.
Data1.Recordset.MoveFirst	İlk kayda götürür.
Data1.Recordset.MoveLast	Son kayda götürür.
Data1.Recordset.BOF	İlk kayıt üzerinde olduğunuzu belirtir.
Data1.Recordset.EOF	Son kayıt üzerinde olduğunuzu belirtir.
Data1.Recordset.AbsolutePosition	O anda hangi kayıt üzerinde olduğunuzu belirtir.
Data1.Recordset.Bookmark	Mevcut kaydın data içersindeki yerini belirler.
Data1.Recordset.LastModified	Üzerinde değişiklik yapılan son kayda gider.
Data1.Recordset.Close	Veri dosyasını kapatmak için kullanılır.
Data1.Recordset.Delete	Veri dosyasından bilgi silmek için kullanılır.
Data1.Recordset.EditMode (Bir kaydın düzenlenişini belirler)	DbEditAdd : Yeni bir kayıt üzerinde işlem yapılıyorsa DbEditInProgress : Kullanıcı bir kayıt üzerinde değişiklik yapıyorsa
Data1.UpdateControls	Yenilemek istediğiniz kayıt.
Data1.UpdateRecord	Bir önceki kayıta yapılan değişiklikleri kaydeder.
Data1.Recordset.CancelUpdate	Yenilemek istediğiniz kaydı iptal eder.
Data1.ReadOnly = True	Data dosyası ReadOnly olur.
Data1.Recordset.Seek	Arama yapılacak bölümlerdeki şart belirtilir.
Data1.Recordset.RecordCount	Toplam kayıt sayısını verir.
Data1.Recordset.Update	Kayıda bilgi ekler.
Data1.Recordset.NoMatch	Arama sonucunda aranılan kayıt bulunamazsa True olur.
Data1.Recordset.Index	Arama yapılırken başlıklarda kullanılır.
Data1.Recordset.Fields(index)	Herhangi bir yerden kayıtlı verilere ulaşmak için kullanılır.
Data1.Refresh	Data dosyasının yeniden açılmasını sağlar
Data1.Exclusive = True	Açılmış olan bir dataya başka kullanıcıların erişimi engellenir

Öncelikle form üzerine bir tane **Data** komponenti yerleştirilir ve **DataBaseName** bölümüne hazırladığımız **üye takibi** 'ni yazarız. **RecordSource** bölümüne ise **Bilgi_Girisi** seçeriz. Artık komponentleri ekrana yerleştirebiliriz.

11 Label ve 11 TextBox, 9 Command Buton yerleştirilir.

Textlere sırasıyla **DataSource** bölümüne **Data1** aktarılır.
Textlere sırasıyla **DataField** bölümleri doldurulur.

```
Private Sub cikis_Click()  
    Data1.Recordset.Close  
    Unload Me  
End Sub
```

```
Private Sub Data1_Reposition()  
    Data1.Caption = Str(Data1.Recordset.AbsolutePosition + 1) + "/" +  
    Str(Data1.Recordset.RecordCount)  
End Sub
```

```
Private Sub ikayit_Click()  
    Data1.Recordset.MoveFirst  
End Sub
```



```

Private Sub iptal_Click()
    Data1.UpdateControls
    If Data1.Recordset.EditMode = dbEditAdd Then
        Data1.Recordset.CancelUpdate
    End If
End Sub

Private Sub kaydet_Click()
    On Local Error Resume Next 'Hata olursa hatanın olduğu satırdan sonraki satıra geçer.
    Data1.UpdateRecord
    Data1.Recordset.Bookmark = Data1.Recordset.LastModified

    ' Data1.Recordset.Update
    ' Data1.Recordset.MoveLast
End Sub

Private Sub kayitsil_Click()
    On Local Error Resume Next
    If Data1.Recordset.RecordCount > 0 Then
        If MsgBox("Kaydı silmek istediğinizden emin misiniz?", vbYesNo + vbDefaultButton2) = vbYes Then
            If Data1.Recordset.AbsolutePosition = 0 Then
                Data1.Recordset.Delete
                Data1.Recordset.MoveNext
            Else
                Data1.Recordset.Delete
                Data1.Recordset.MovePrevious
            End If
        Else
            Exit Sub
        End If
    Else
        MsgBox ("Dosyada kayıt yok")
    End If
End Sub

Private Sub onkayit_Click()
    On Local Error Resume Next
    Data1.Recordset.MovePrevious
    If Data1.Recordset.BOF Then
        MsgBox "İlk kayıt üzerindesiniz"
        Data1.Recordset.MoveFirst
    End If
End Sub

Private Sub sonkayit_Click()
    On Local Error Resume Next
    Data1.Recordset.MoveLast
End Sub

Private Sub sonrakikayit_Click()
    On Local Error Resume Next
    Data1.Recordset.MoveNext
    If Data1.Recordset.EOF Then
        MsgBox "Son kayıttasınız"
        Data1.Recordset.MoveLast
    End If
End Sub

```

```
Private Sub yenikayit_Click()  
On Local Error Resume Next  
Data1.Recordset.AddNew  
Text1.SetFocus  
End Sub
```

DBGrid

Ekrana bir tane DBGrid yerleştirilir ve DataSource: Data1 yapılır.

AllowAddNew

Tablo üzerinde yeni kayıt düğmesini aktifleştirir.

AllowArrows

Tablo üzerinde yön tuşlarını aktifleştirir.

AllowDelete

Tablo üzerinde silme işlemi yapılabilir.

AllowUpdate

Tablo üzerinde yenileme işlemi yapılabilir.

İstenirse DBGrid de birkaç bölüm gösterilebilir. Bunun için sağ klik yapılarak Properties'e girilir ve oradan da Columns sekmesine geçilir ve istediğiniz kolonlar ayarlanır. Default olarak 2 kolon gelmektedir istenirse arttırılabilir. Bunun için sağ klik yapılır ve Edit yapılır, sonra yine sağ klik ve Append veya Insert yapılarak kolon eklenir.

ADO ile Veri Tabanı

Microsoft ADO Data Control 6.0 (MSADODC.OCX)

Data bileşenine benzeyen yeni bir komponentdir.

Adodc üzerindeki sağ klik yapılır ve properties girilir. Use Connection String seçiliyken Build seçeneğinden mdb dosyamız seçilir. RecordSource sekmesinden Command Type bölümünden 2-adCmdTable seçilir. Altta açılan menüden Table adı seçilir ve Tamam'a basılır.

Form üzerine gerekli komponentler yerleştirilir.

Komponentlerin DataSource bölümüne Adodc1 yazılır.

DataGrid

Microsoft DataGrid Control 6.0 (MSDATGRD.OCX)

Ekrana bir tane DataGrid yerleştirilir ve DataSource: Adodc1 yapılır. DataGrid üzerinde istediğimiz gibi ayarlamalar yapabiliriz. Bunun için datagrid üzerindeki sağ klik yapılır ve Retrieve Fields seçilir. Tekrar sağ klik yapılır ve Edit seçilir.

AllowAddNew

Tablo üzerinde yeni kayıt düğmesini aktifleştirir.

AllowArrows

Tablo üzerinde yön tuşlarını aktifleştirir.

AllowDelete

Tablo üzerinde silme işlemi yapılabilir.

AllowUpdate

Tablo üzerinde yenileme işlemi yapılabilir.

ColumnHeaders

Tablo üzerindeki isimleri gizler.

DefColWidth

Tablo üzerindeki sütunların genişlikleri ayarlanır.

HeadLines

Tablo üzerindeki başlıklara ayrılacak satır sayısı.

RowHeight

Tablo üzerindeki satırların yükseklikleri ayarlanır.

RowDividerStyle

Tablo üzerindeki satırlar çizgilerini biçimlendirir.

TabAcrossSplits

Tablo üzerinde bir split uygulanmışsa tabla bu split içine gidilebilir.

WrapCellPointer

Tablo üzerinde yön tuşlarıyla hareket ederken satır bitince otomatik olarak bir üst veya alt satıra imleci götürür.

TabAction

Tablo üzerinde tab tuşunun rolünü ayarlar.

- I. Tab tuşu grid üzerinde aktif olmaz.
- II. Tab tuşu grid üzerinde sadece kolonlar arasında geçer. Kolonlar bitince bir üst komponente geçer.
- III. Tab tuşu sadece grid üzerinde geçerlidir.

Locked

Tablo üzerinde işaretlenen kolona girilemez.

AllowSizing

Tablo üzerinde işaretli sütunun genişliği fare ile ayarlanabilir.

Button

Tablo üzerinde işaretli kolon üzerinde aşağıya doğru açılan bir ok çıkar.

DividerStyle

Tablo üzerinde işaretli kolon üzerindeki çizgileri değiştirir.

Split Locked

Tablo üzerinde split olan bölüme bilgi girişi yapılamaz.

Split AllowFocus

Tablo üzerinde split olan bölüme geçiş yapılabilir.

Split AllowSizing

Tablo üzerinde split olan bölümün genişliği ayarlanabilir.

Split AllowRowSizing

Tablo üzerinde split olan bölümün satır genişlikleri ayarlanabilir.

Split RecordSelectors

Tablo üzerinde split olan bölümün başındaki satırı seçme bölümü kaldırılır.

Format

Tablo üzerinde kolonlara uygulanacak formatlar ayarlanır.

Data Report

Project menüsünden Add Data Report eklenir

RptLabel

Rapor başlıklarında kullanılır.

RptTextBox

Bilgi girişlerini göstermek için kullanılır.

RptImage

Rapora resim eklemek için kullanılır.

RptLine

Rapor üzerine çizgi koymak için kullanılır.

RptShape

Rapor üzerine şekilleri çizmek için kullanılır.

RptFunction

Rapor içinde üst veya alt bilgi bölümlerinde kullanılır.

Form üzerine bir tane DataEnvironment yerleştirilir, mdb dosyası açılır. Data View açılır ve istediğiniz Table DataEnvironment üzerine mouse yardımıyla yerleştirilir. DataEnvironment üzerinden istediğiniz komponentler Rapor üzerine yerleştirilir. Komponentler üzerinde değişiklik yapmak için sağdaki properties seçeneğinden veya sağ klik yapılarak uygulanabilir.

```
Private Sub Command2_Click()  
    DataReport1.Show  
End Sub
```

Raporlara özel komutlar eklemek

Rapor içinde sağ klik yapılarak açılan menüden Insert Control seçeneğinden

%p	Aktif sayfanın numarası
%P	Toplam sayfa sayısı
%d	Kısa formatlı günün tarihi
%D	Uzun formatlı günün tarihi
%t	Sistem saati kısa formatlı
%T	Sistem saati uzun formatlı
%i	Raporun başlığı

Raporlara hesaplamalar eklemek

Rapor içinde rptFunction eklenir.

0	rptFuncSum	Toplamı bulur,
1	rptFuncAve	Ortalamayı bulur,
2	rptFuncMin	En küçük değeri bulur,
3	rptFuncMax	En büyük değeri bulur,
4	rptFuncRCnt	Satır sayısını bulur,
5	rptFuncVCnt	Boş olmayan alan sayısını bulur,
6	rptFuncSDEV	Standart sapmayı bulur,
7	rptFuncSERR	Standart hatayı bulur.

Like

Like komutu ile yazdığımız bir Text in neresinde olursa olsun bir yazıyı bulmak için kullanılır.

ÖRNEK 1

```
Private Sub Command1_Click()
```

```
If Text1 Like "*İSTANBUL*" Then 'like komutu text içinde arama yaptırır
```

```
    '* * yazinin neresinde olursa olsun bulur.  
    '*   yazinin sonunda olursa bulur.  
    ' * yazinin başında olursa bulur.  
    ' ÖNEMLİ = arama büyük küçük harfe duyarlıdır. Çünkü arama ASCII kodlarına göre yapar.
```

```
MsgBox ("Bu arkadasın İstanbul'da ikamet ettiği tespit edildi")  
Else  
MsgBox ("Bu arkadaş İstanbul' lu değil")  
End If  
End Sub
```

ÖRNEK 2

Private Sub Command2_Click()

' ??İstanbul dersek İstanbul'un önündeki 2 harfi anlatır.

' ## İstanbul dersek İstanbul'un önündeki 2 rakamı anlatır.

' "[a-g]İstanbul" dersek İstanbul'un önünde a-g harfi varmı diye aratır.

'örnek aistanbul, eistanbul gibi

If Text1 Like "[a-g]İSTANBUL" Then

MsgBox ("Bu arkadasın İstanbul'da ikamet ettiği tespit edildi")

Else

MsgBox ("Bu arkadaş İstanbullu değil")

End If : End Sub

SQL Komutları

Yeni bir mdb oluşturulur.

Name: uye

TableName : giris

ÜyeNo

Adı

Soyadı

dogtarihi

Sehir

SehirKodu

Telefon

Maas 'long

En çok kullanılan sql komutları.

Form üzerine bir tane Adodc1 ve DataGrid1 yerleştirilir. Adodc1'e mdb dosyası bağlanır. RecordSource : 8 yapılır ve Command Text bölümüne : **select * from mdb** yazılır. Oluşturduğumuz DataGrid1.DataSource : Adodc1 yapılır.

Select

Tablo içerisinden bilgi seçmek için kullanılır.

1.

Adodc1.RecordSource = "select * from giris"

Adodc1.Refresh

2.

Adodc1.RecordSource = "select uyeno, ad, soyad from giris"

Adodc1.Refresh

Where

Tablo içerisinden sorgulama kriterini, koşul belirtmeyi sağlar.

ComboBox oluşturulur ve içerisinde oluşturduğumuz datalar eklenir. İki tane OptionButton eklenir ve adları asc ve desc yapılır.

```

Private Sub Combo1_Change()
Dim a
If Option1.Value = True Then
    a = "asc"
Else
    a = "desc"
End If
Adodc1.Recordset.Sort = Combo1.Text & " " & a
End Sub

```

1.

Adodc1.RecordSource = "select * from giris where (ad like '%o%') " **Sadece adı O harfi ile başlayan kişiler gösterilir.**

Adodc1.Refresh
2.

Adodc1.RecordSource = "select * from giris where ad = 'oguz'" **Sadece adı oguz olanlar listelenir.**

Adodc1.Refresh
- 2.1.

Dim a

a = InputBox("Sorgulama yapmak istediğiniz harfi giriniz")

Adodc1.RecordSource = "select * from giris where (ad like '" & a & "%') " **Girdiğiniz harfle başlayanlar listelenir.**

Adodc1.Refresh
3.

Adodc1.RecordSource = "select * from giris where ad = 'oguz' and sehir='İstanbul'" **sadece istanbulda oturan oğuz adlı kişiler listelenir.**

Adodc1.Refresh
4.

Adodc1.RecordSource = "select * from giris where ad = 'oguz' or sehir='İstanbul'" **sadece istanbulda oturanlar veya ismi oğuz olanlar listelenir.**

Adodc1.Refresh
5.

Adodc1.RecordSource = "select * from giris where dogtarihi >= #01/01/73# " **sadece doğum tarihi 01/01/73 tarihinden büyük olanlar listelenir.**

Adodc1.Refresh
6.

Adodc1.RecordSource = "select * from giris where maas > 250000000" **sadece maaşı 25000000 TL üzerinde olanlar listelenir.**

Adodc1.Refresh
- 6.1.

Dim a

a = InputBox("Miktarı giriniz")

Adodc1.RecordSource = "select * from giris where maas <" & a **Maaşı girdiğiniz miktardan küçük olanlar listelenir.**

Adodc1.Refresh
7.

Adodc1.RecordSource = "select distinct ad, soyad from giris " **sadece ad, soyad gösterilir ve aynı kayıtlar gösterilmez**

Adodc1.Refresh
8.

Adodc1.RecordSource = "select ad, soyad, sehir, (sehkodu*2) as [yeni] from giris order by ad desc"

Adodc1.Refresh **Buradaki (sehkodu*2) sehir kodunu 2 ile çarpar ve as [yeni] adlı bir sütun açarak oraya yazar.**
- 9.

Adodc1.RecordSource = "select top 3 * from giris where sehkode='212'" **Burada sehirkodu 212 olan ve üst sırada bulunan 3 kayıt gösterilir.**

Adodc1.Refresh

10.

Adodc1.RecordSource = "select count(ad) from giris "

Adodc1.Refresh **ad bölümünde bulunan eleman sayısını verir.**

11.

Adodc1.RecordSource = "select max(sehkodu) from giris " **ilgili elemanın en büyük sayısını bulur.**

Adodc1.Refresh

12.

Adodc1.RecordSource = "select min(sehkodu) from giris " **ilgili elemanın en küçük sayısını bulur.**

Adodc1.Refresh

13.

Adodc1.RecordSource = "select avg(sehkodu) from giris " **ilgili elemanın ortalamasını bulur.**

Adodc1.Refresh

14.

Adodc1.RecordSource = "select sum(sehkodu) from giris " **ilgili elemanın toplamını bulur.**

Adodc1.Refresh

15.

Adodc1.RecordSource = "select ad, maas from giris group by [ad]" **ada göre grup yapar ve maasları toplar.**

Adodc1.Refresh

16.

Adodc1.RecordSource = "select ad, sum(maas) from giris group by [ad] having avg(maas) < 250000000"

Adodc1.Refresh **ada göre grup yapar ve oluşturulacak grubu maaşı < 250000000 olanları gösterir.**

17.

Adodc1.Recordset.Sort = "ad asc" **ad'a göre sort yapar.**

18.

Dim a

a = InputBox("Sıralama yapmak istediğiniz alan adını giriniz")

Adodc1.Recordset.Sort = a & " asc" **istediğiniz bir alana göre sıralama yaptırabilirsiniz.**

19.

Dim a, b

a = InputBox("Sıralama yapmak istediğiniz alan adını giriniz")

b = InputBox("Sıralama şeklini giriniz", , "asc")

Adodc1.Recordset.Sort = a & " " & b & " "

20.

Dim a

a = InputBox("Sıralama yapmak istediğiniz alan adını giriniz")

Adodc1.Recordset.Filter = a & "'oguz'" **istediğiniz bir alanı flitrelemek için kullanılır.**

21.

Dim a

If KeyAscii = 13 Then

a = InputBox("Sıralama yapmak istediğiniz alan adını giriniz")

Adodc1.Recordset.Filter = a & "=" & Text2.Text & ""

End If **istediğiniz bir alanı textbox kullanarak flitrelemek için kullanılır.**

22.

Private Sub Text3_Change()

If Len(Text3.Text) <> 0 Then

Adodc1.Recordset.Filter = "ad like '" & Text3.Text & "%'"

Else

Adodc1.Recordset.Filter = adFilterNone

Adodc1.Refresh **Bir text kutusuna harf yazdıkça flitreleme yapar.**

End If

End Sub

Order By

Tablo içerisinden bilgileri sıralı şekilde göstermek için kullanılır.

Asc A-Z sıralı

Desc Z-A sıralı

Distinct

Tablo içerisinden aynı kayıtları göstermez.

Distinctrow

Tablo içerisinden aynı kayıtları gösterir.

Group By

Tablo içerisinden seçili veriler arası gruplandırma yapar.

Between

Tablo içerisinde şartın iki aralıkta olup olmadığını kontrol eder.

Like

Tablo içerisinden belirlenen kriterlere uygun bilgileri seçmek için kullanılır.

Having

Tablo içerisinden gruplar arası şart belirler.

Formlar

AutoRedraw

Form üzerinde bulunan çizimlerin ve yazıların form üzerine başka bir form geldiğinde veya form kapatıldığı zaman bil bozulmamasını sağlarız.

BorderStyle

- | | |
|---|--|
| 0 | Formun boyutları değiştirilemez. |
| 1 | Boyutları değiştirilemeyen ama taşınabilir form. |
| 2 | Default değeri. |
| 3 | 1 e benzer |
| 4 | Başlığı küçük olan form. |
| 5 | 4 e benzer ama boyutlandırılabilir. |

ControlBox

Form üzerinde bulunan sol taraftaki ikon üzerine basınca çıkan control kutusunun açılması sağlanır.

Icon

Form üzerinde bulunan ikonun değişmesi için kullanılır.

KeyPreview

Form üzerinde klavyenin kullanılması için True yapılır.

MaxButton

Form üzerinde bulunan max butonu için kullanılır.

MinButton

Form üzerinde bulunan min butonu için kullanılır.

MDIChild

Form üzerinde eğer bir MDI form olursa true yaparak bu formu onun üyesi yapılır.

Moveable

Form un kullanıcı tarafında taşınmaması için kullanılır.

ShowInTaskbar

Formu minimize yapınca görev çubuğunda göstermez.

StartPosition

Formun başlangıçta ekranın neresinde olacağını belirler.

- | | |
|---|--|
| 0 | Manual |
| 1 | Form içinde bulunduğu formun ortasında (MDI formlarda) |
| 2 | Ekranın ortasında |
| 3 | Devamlı kaydırarak |

WindowState

- | | |
|---|----------|
| 0 | Normal |
| 1 | Minimize |
| 2 | Maximize |

MDIFormlar

Add Form dan MDI form seçilir.

AutoRedraw

Form üzerinde bulunan çizimlerin ve yazıların form üzerine başka bir form geldiğinde veya form kapatıldığı zaman bil

Arrange

MDIForm içerisinde bulunan formların görünüm özelliklerini değiştirir.

- | | |
|------|-----------|
| I. | Basamakla |
| II. | Yatay |
| III. | Dikey |

WindowList

MDIForm içinde açık bulunan formları otomatik olarak gösterilmesini sağlar.

