

ÜNİTE 1 : TEMEL BİLGİLER

Microsoft Visual Basic Nedir?

Visual Basic (VB) çok amaçlı ve aynı zamanda kolay bir şekilde program geliştirmenizi ve oluşturmanızı sağlayan bir programlama dilidir. Visual anlamı görsellik anlamındadır. Program yapım aşamasında programın ekran tasarımı kod yazarak değil de Mouse ile resim çizermiş gibi yapılmaktadır. Basic ise bildiğimiz Basic programlama dilinden gelmiştir, ancak Basic programlama diliyle ilgisi yoktur.

Visual Basic ile sadece program geliştirmeye kalmazsınız, VBA(Visual Basic Application) olarak bilinen ve Microsoft içinde (Excel,Word,Access) programlarında da kullanılır. Ayrıca internet uygulamalarında kullanılan VBscript de Visual Basicin bir alt şeklidir. Visual Basic ile yapılan programlar arasında küçük uygulamaların yanı sıra büyük çaplı, internet uygulamaları, cari hesap , stok , satış, oyun programı gibi programları da yapabilirsiniz.

Kuruluş İşlemi

Visual Basic programını bilgisayarınıza kurmak için gerekli sistem gereksinimleri aşağıda verilmiştir.

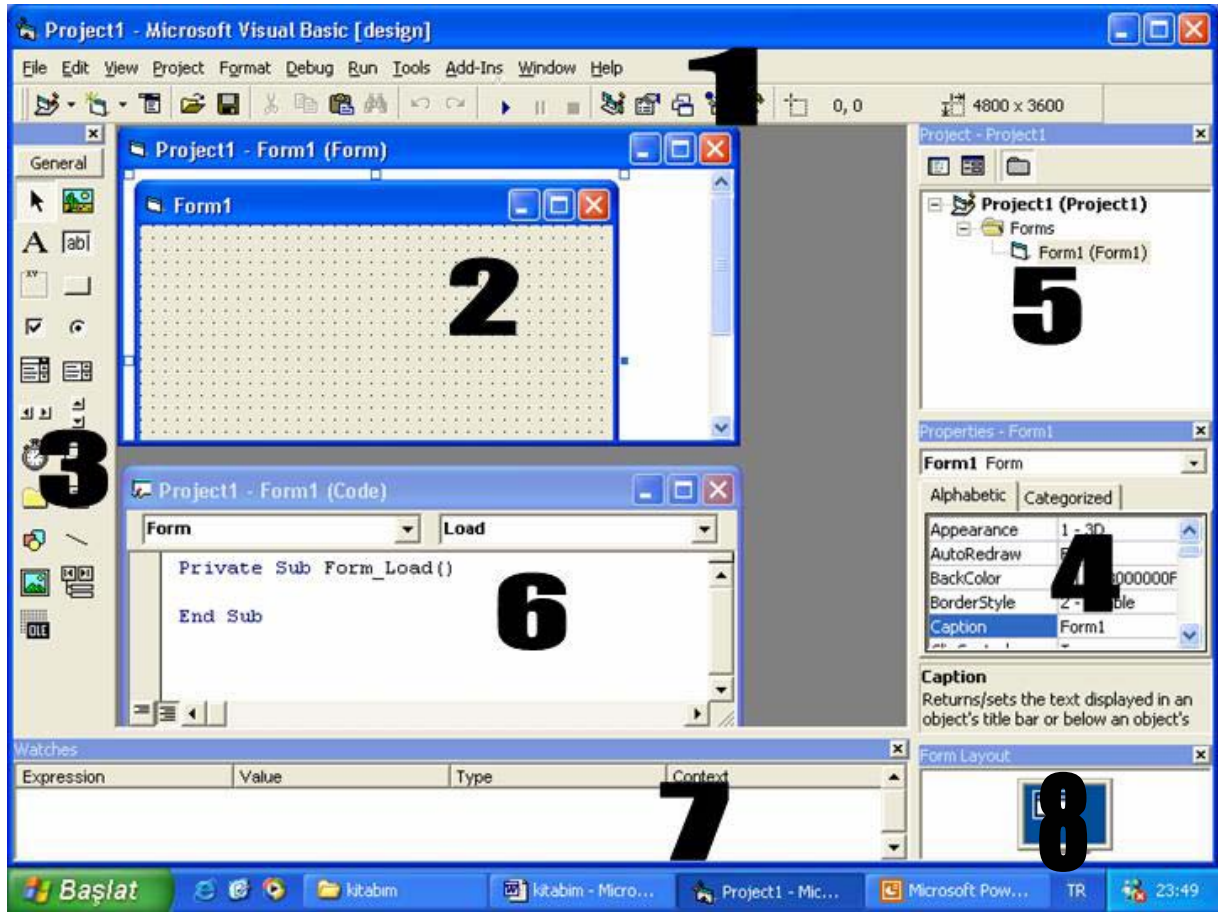
- Windows NT,Windows 95-98-XP işletim sistemi
- 486 mhz işlemciden yukarı işlemci
- en az 50 MB alan
- VGA yada daha yüksek çözünürlüklü ekran
- 16 MB Ram
- Fare
- MSDN(help dosyaları için) CDsi

ÜNİTE 2 :Visual Basic Programlamaya Giriş

Visual Basic programını ilk olarak çalıştırdığınızda ilk olarak aşağıdaki pencere karşınıza gelecektir. Bu pencerede **standart exe** normal programlar yapmak için kullanılır. Bu seçeneği işaretledikten sonra **Aç** düğmesine basılır. Daha sonra ekranınıza VB çalışacak ve boş bir form gelecektir.



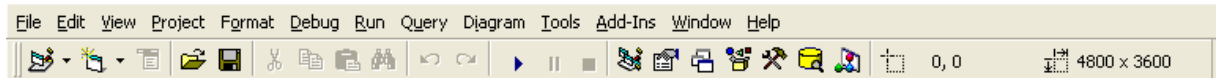
Visual Basic Ekranı



1. VB ye ait menüler ve araç çubukları
2. Kullanıcının kendi programını oluşturması için gerekli form
3. Hazır öğeleri programa yerleştirmek için gerekli Toolbox
4. Form üzerine yerleştirilen öğelerin özelliklerini değiştirmek için kullanılan Properties penceresi
5. Projede bulunan dosyaları gösteren Project Explorer penceresi
6. Program kodu yazmak için kullanılan kod penceresi
7. Programın adım adım çalışmasını izleyebilmek için değişken izleme penceresi
8. formun ekrandaki yerini gösteren Layout penceresi

1. VB Ana Formu

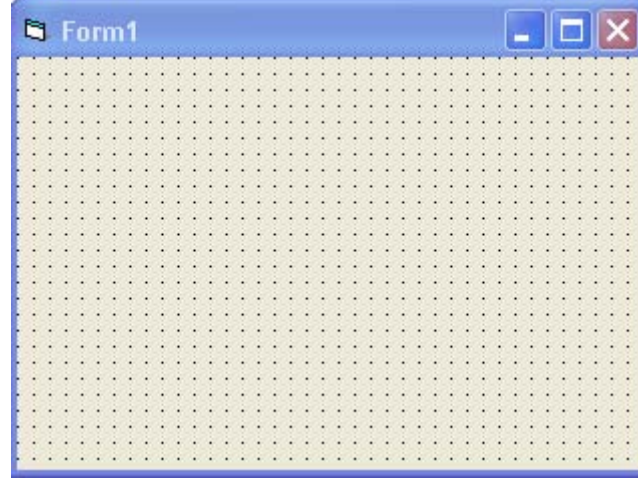
VB çalıştırıldığında VB kendine ait olan aşağıdaki form görülür.



Bu pencerede VB ait araç çubukları ve menüler bulunmaktadır.

2. Programcı Formu

VB en önemli özelliği daha öncede belirtildiği gibi programın form penceresinin resim çizerek yani kod ortamında değil, yapılması idi. Form ekranında oluşturduğunuz form görüntüsü program çalıştırıldığında aynen gözükecektir. VB Visual özelliği zaten bu özelliğidir. Programı tasarımında sizi ekran tasarımı için sıkıcı kodlardan kurtaran bu özellikte kolayca ve görerek form tasarlanmaktadır. Bir program içerisinde birden fazla form bulunabilir ve form üzerine yerleştirilen her kontrol elemanın özellikleri properties penceresinde gözükür.



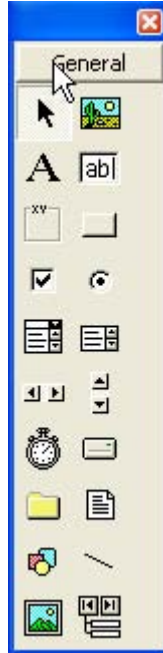
Form penceresi ekranınızda göremiyorsanız

1. açık olan proje olmayabilir.
2. programınız çalışıyor olabilir.
3. form kapatılmıştır.

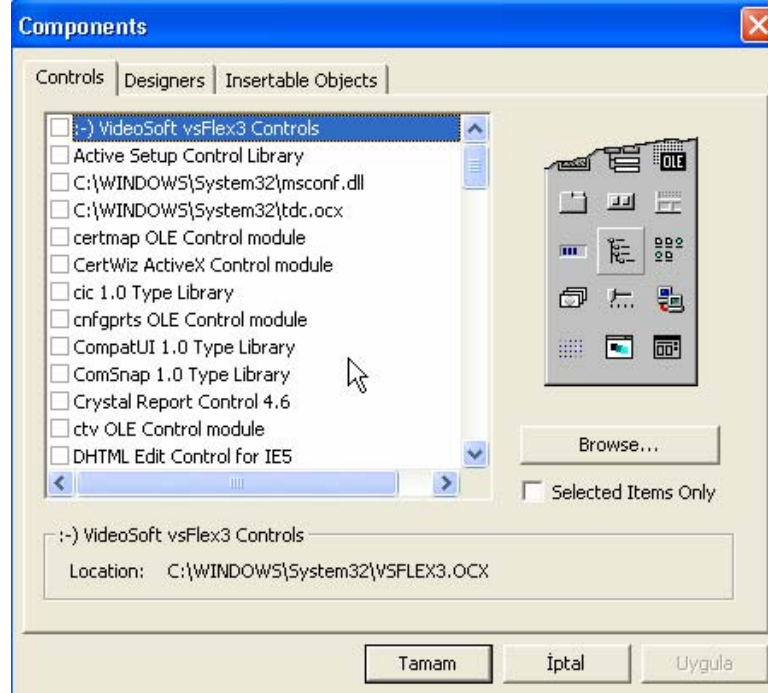
3. Toolbox

Form üzerine yerleştirilecek kontrollerin tamamı toolbox penceresinde yer alır.

Form üzerine herhangi bir kontrolü eklemek istediğinizde bu pencereden kontrolü seçip form üzerinde dikdörtgen şeklinde fare ile çizilir. Aynı işlem çift tıklama ile de yapılabilir.



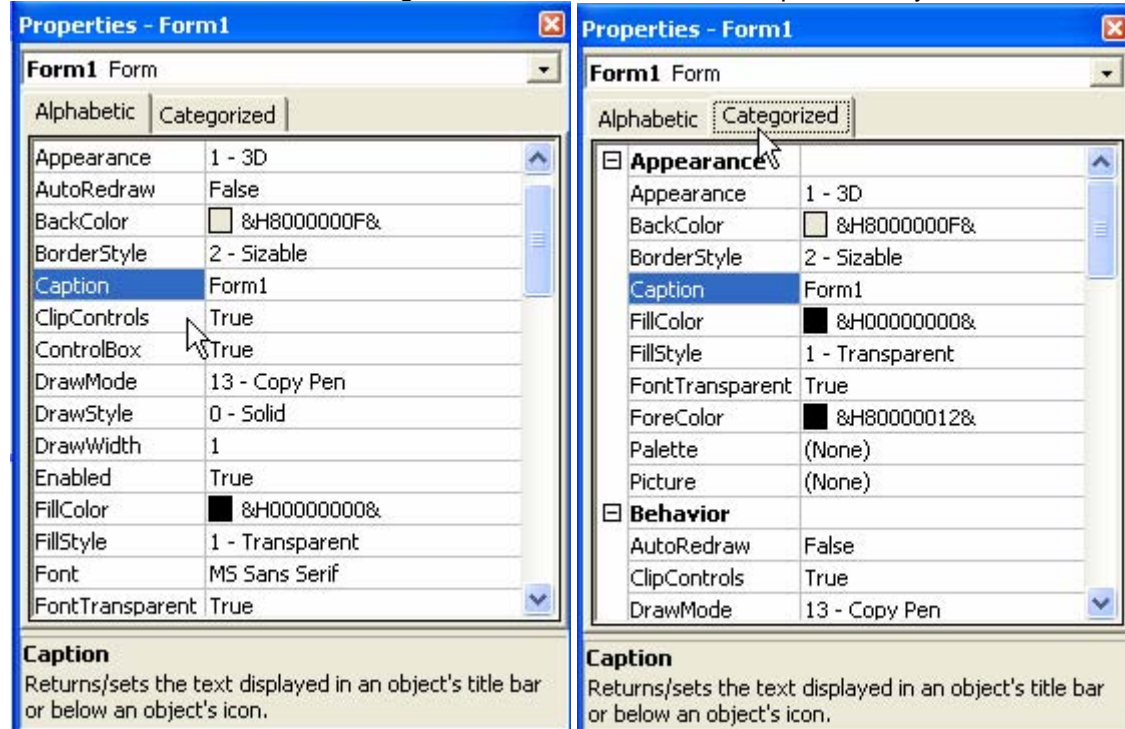
VB'nin kontrolleri sadece Toolbox penceresinde görülen kontrollerden ibaret değildir. VB içerisinde olan ancak bu pencerede gözükmeyen kontroller vardır. Bu kontroller **Project menüsünde Components** komutu tıklanarak eklenir.



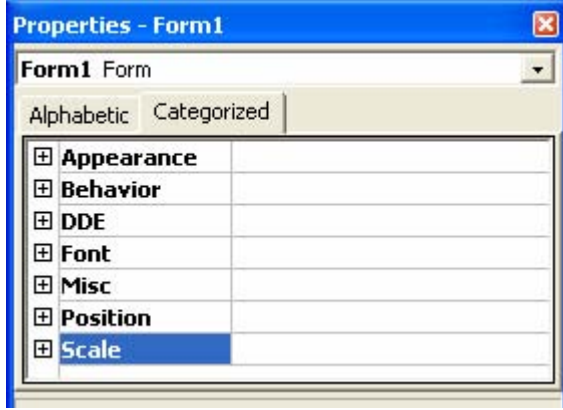
Bu pencerede VB eklenebilecek bütün kontroller bulunur. Eğer bu kontrollerden birini eklemek isterseniz kontrolün başındaki kutucuğu işaretlemeniz yeterlidir.

4- Properties (Özellikler) Penceresi

Form üzerine yerleştirilecek elemanların, form dahil tümünün özellikleri bu pencerede yer alır. Form üzerinde bir kontrole tıkladığınızda kontrole ait özellikler bu pencerede yer alır.



Seçili olan elemanın özellikleri (properties) penceresine yansıyan özellikler hem alfabetik hem de kategorik olarak listelenebilir.



Kategorik olarak gösterimde;

Apperance: Elemanın görünüşü ile ilgili özellikler

Behavior: Elemanın durumu ile ilgili özellikler

DDE,Font: font ile ilgili özellikler

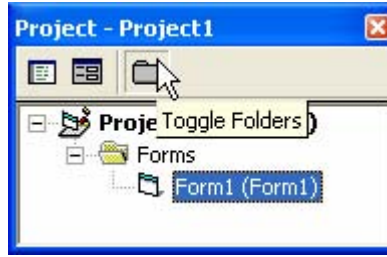
Misc: bir guruba dahil edilemeyen farklı farklı özellikler

Position: elemanın boyutları , from ev ekran üzerindeki konumları ile ilgili özellikler

Scale: Elemanın bulunduğu aralık ile özellikler

- Eğer properties penceresi ekranda gözüküyor ise View menüsünden Properties Windows komutunu seçilir. Veya kısaca F4 tuşuna basılır.

5- Project Penceresi



Bu pencerede programa dahil edilen form, module vb. gibi elemanların listesi yer alır. Bu pencere ile projeye dahil edilen formlar arasında geçiş sağlamak mümkündür.

Bu pencerenin sol üst köşesinde bulunan 3 adet düğmenin



1.si: seçili nesne ile ilgili kod penceresini açar

2.si. nesnenin form tasarımını gösterir.

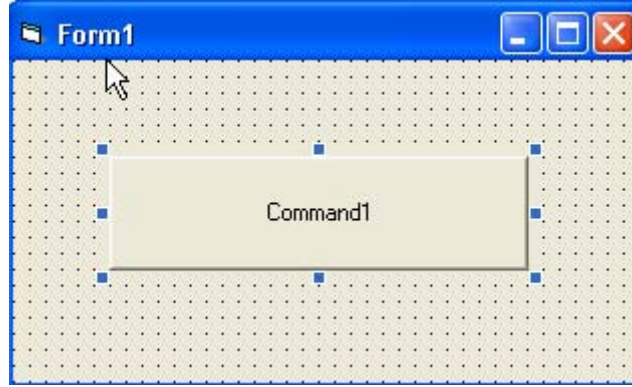
3.si. aynı cinsten olan form yada klasörleri bir klasör ortamında gösterir.

- Eğer pencere ekranda gözüküyor ise View menüsünden Project Explorer komutunu kullanılır.
- Projeye dahil edilen bir form yada modülü projeden çıkarmak için seçili form üzerinde sağ tıklayarak açılan pencereden remove seçeneği seçilir.
- Pencereye yeni bir dosya eklemek için ise yine aynı pencerede sağ fare tuşu ile açılan pencereden add menüsü seçilir.

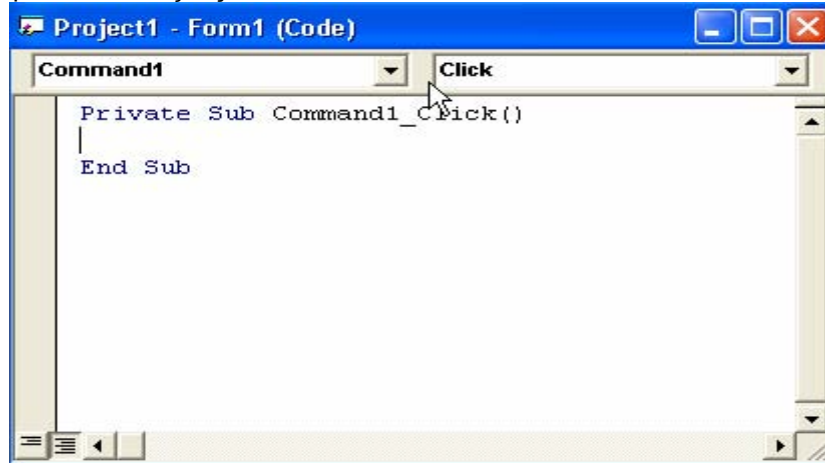
6- Kod Penceresi

VB programcının yazacağı kodlar için kendi kod penceresini kullanıcıya sunmuştur. Program yazarken genellikle kodlar form üzerine yerleştirilen nesnelerin olaylarına yazılır. Kod yazma için herhangi bir kontrolün üzerinde çift tıklamanız yeterlidir yada kontrol üzerinde iken F7 tuşuna basmanız da o kontrole ait kod penceresini açacaktır.

Örneğin formun üzerine bir Komut düğmesi (Command) ekleyelim.



bu düğmeye basıldığında bir mesaj vermesini istiyorsak, Komut düğmesi üzerinde çift tıklayarak kod penceresini açmış oluruz.



Açılan kod penceresi ;

```
Private Sub Command1_Click()
```

```
End sub
```

İle belirtilen ve komut düğmesinin tıklanmasındaki meydana gelebilecek olayları temsil eden bölümdür. Örneğimizde eğer komut düğmesi tıklandığında mesaj vermesini istiyorsak mesaj verecek komut satırlarını bu bölümde yazmamız gerekir.

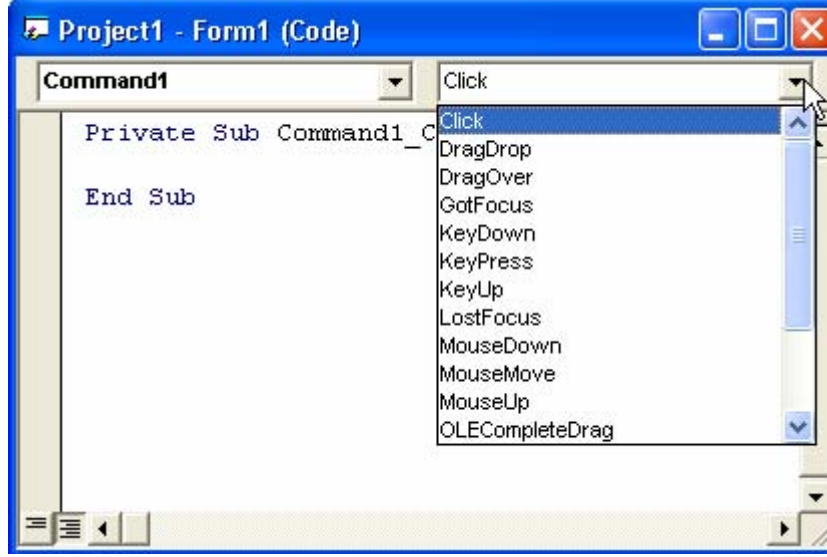
```
Private Sub Command1_Click()
```

```
Msgbox ("VB Öğreniyorum")
```

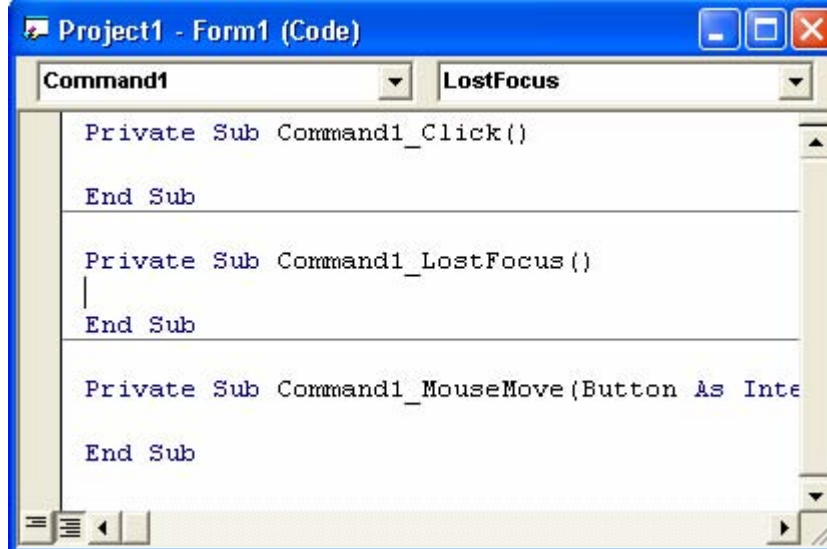
```
End sub
```

- Burada Command1_Click durumunda click terimi komut düğmesinin tek tıklandığındaki olay yordamıdır. End sub ise tek tıklama ile yani click ortamının sona erdiğini belirtir.
- **Şimdi Programımızı çalıştırmak istersek F6 tuşuna yada Run menüsünden Start düğmesine basılarak çalıştırabilirsiniz.**

Kodlar Kod ortamında yazılırken sadece o kontrole ait bir olay kullanılmaz. Örneğin komut düğmesinde click tek tıklama olayıdır. Başka bir olay olduğunda ise yapılması gereken ise kod penceresindeki sağ üst köşede bulunan listeyi açarak kontrole ait olayları görebiliriz.



Bu listede kontrole ait olay seçildiğinde o olay ait at program başlığında otomatik olarak VB tarafından kod ortamına yazılır.

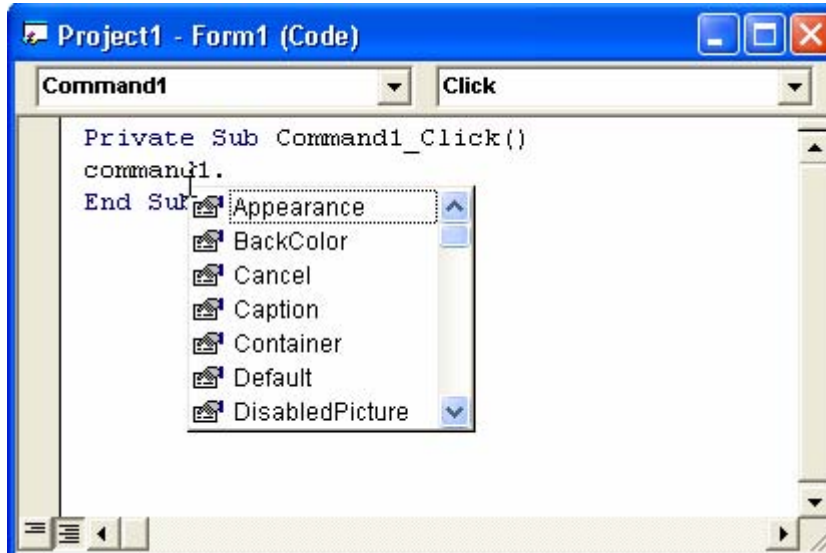


Yukarıdaki Pencerede sağ üst köşede bulunan ve comman1 ile belirtilen bölüm ise form üzerine yerleştirilmiş kontrolleri gösterir.

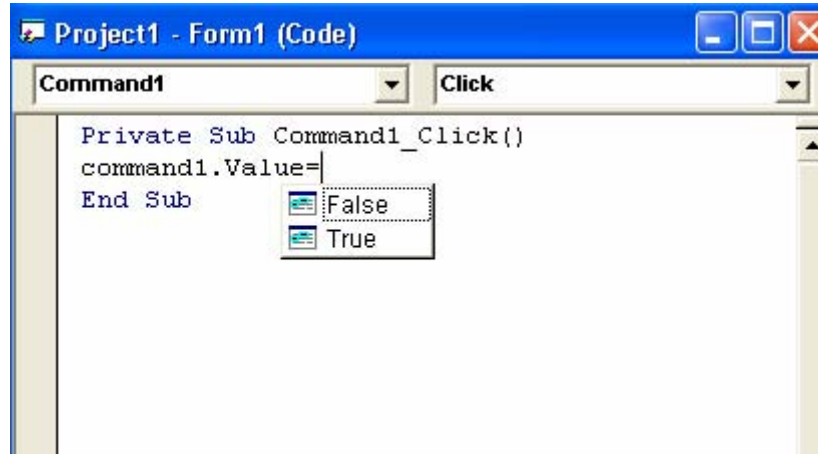
Kod Penceresine Ait Özellikler:

VB kullanıcıyı tamamen ezberden uzak tutmak amacı ile kod ortamında bir yöntem geliştirmiştir.

Kullanıcı herhangi bir kontrol elamanın adını yazıp "." Nokta karakterine basıldığında o anda kursorun bulunduğu noktadan itibaren aşağıya doğru bir liste açılır. Buarada tüm elemanların ortak kullandığı özellikler ve metodlar yer alır. Kullanılacak özellik bu pencereden seçilerek program satırına dahil edilir.



Aşağı doğru açılan listeden istenilen özellik seçilip “=” karakterine basıldığında seçilen özelliğe ilgili kullanılabilecek özellik aynı metotla gösterilir.



bu özelliğin kullanılabilmesi için mutlaka form üzerinde o kontrolün eklenmesi gerekir.

Diğer bir özellik ise kullanılacak fonksiyon, metod ve özelliklerin parametrelerini göstermesidir. Buda hatasız kodlama için büyük kolaylık sağlamaktadır.

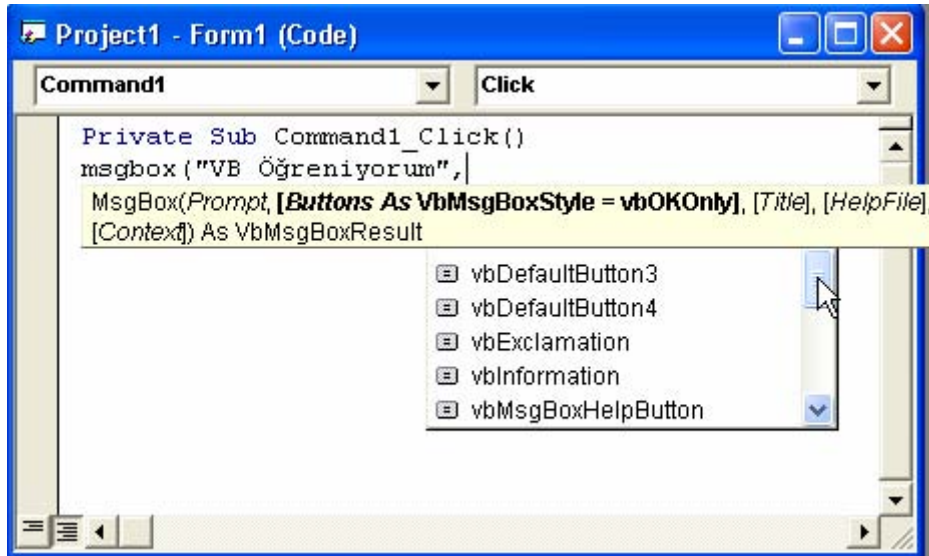
Örneğin bir mesaj kutusu oluşturalım.

Kod penceresinde aşağıdaki kodu yazmaya başladığımızda:

Msgbox (

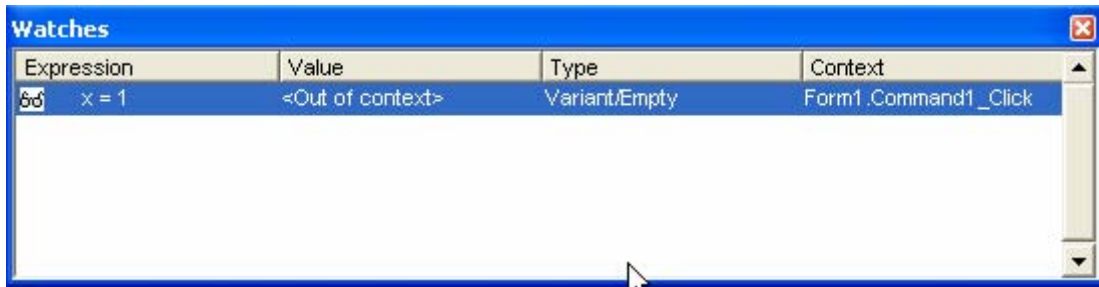
```
MsgBox(Prompt, [Buttons As VbMsgBoxStyle = vbOKOnly], [Title], [HelpFile], [Context])  
As VbMsgBoxResult
```

Msgbox ifadesi yazılıp boşluk yada parantez tuşuna basıldığında yukarıda görülen fonksiyona ait parametreler gösterilmektedir. Burada örneğin ilk parametre olan mesaj yazılıp “,” karakteri basıldığında sıradaki parametre koyulaşır ve o parametreye ait varsa özellikler penceresi açılır.



7- Watches – Değişken İzleme Penceresi

Program yazılırken bazen beklenmedik farklı sonuçlar çıkar. Bu durumda değişkenlerin değerlerini takip ederek sorunun nedenini bulmak gerekir. Bu işlemler için Debug-Add Watch menüleri ile tanımlanır. Böylece bu değişkenler Watches penceresine eklenecektir.



8- Form Layout Penceresi

VB ekranın sağ alt köşesinde bulunan ve içinde monitör resmi bulunan penceredir. Bu pencere aracılığı ile formumuzu ekranın istediğiniz konumuna almak için kullanılır.



9- Object Browser

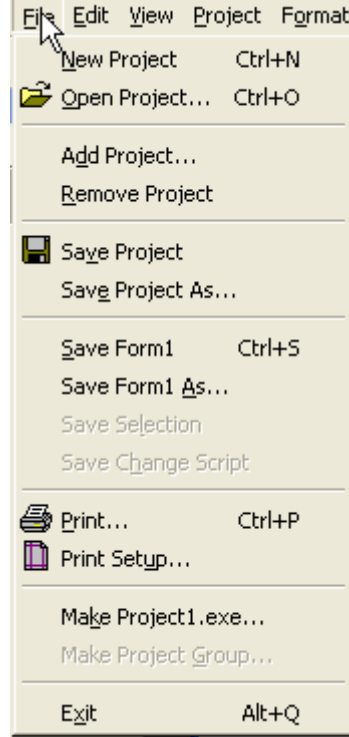
View menüsünde Object – Browser menü seçeneği ile pencere ekrana gelir. Bu pencere aracılığı ile VB kütüphanesindeki kullanılan tüm veriler, proje ve projenin modüllerinin kullandığı tüm tip ve değişkenler hakkında bilgi sahibi olmak mümkündür.



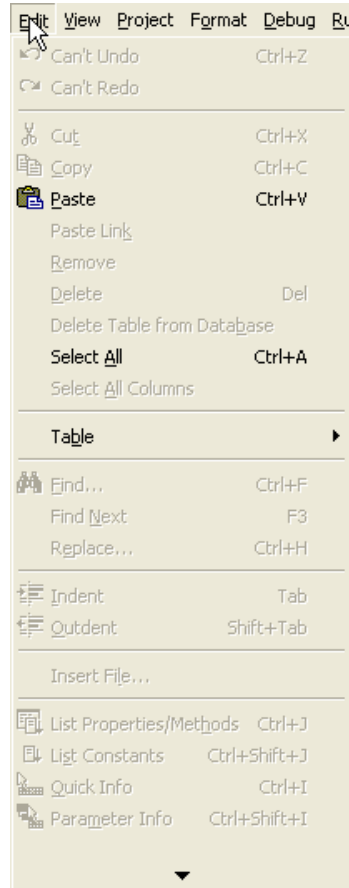
Bu pencerenin sol üst köşesinde combo kutusu aracılığı ile VB kütüphanelerine ulaşılır. Sağdaki liste kutusunda bu kütüphanenin veri sınıfları, soldaki liste ise kutusunda ise ilgili veri sınıflarının içerdiği fonksiyonlar, sabitler ve özellikler yer alır. Pencere altında ise seçili verinin tanımı ve parametreleri verilmektedir.

VİSUAL BASİC MENÜLERİ

File Menüsü



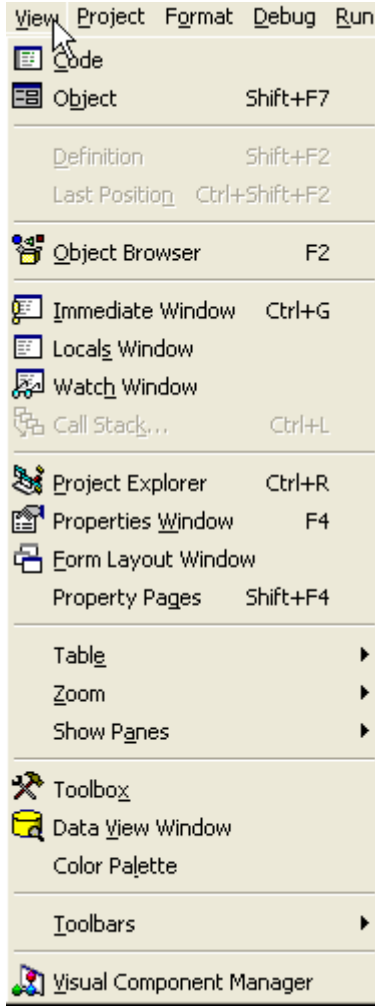
- New Project** : Yeni bir projeye başlamayı sağlar
- Open Project** : Daha önce kaydedilmiş projeyi açar.
- Add Project** : Mevcut projeye yeni proje ekler.
- Remove Project** : projeyi ve elemanlarını siler
- Save Project Group** : Projeleri kayıt eder
- Save Project Group As** : Projelere yeni ad verir.
- Save <form1>** : Formu kayıt eder.
- Save <form1> As** : Forma yeni ad verir.
- Save Selection** : Seçimi kayıt eder
- Save Change Script** : Transact SQL deyimlerin kayıt edilmesini sağlar.
- Print** : Projenin çeşitli elemanlarını yazdırır.
- Print Setup** : yazıcı ayarlamalarını yapar.
- Make Project1.exe** : Projenin .exe dosyası haline getirilmesini sağlar.
- Make Project Group** : projeleri bir grup olarak birleştirir.
- Exit** : VB çıkış



Edit Menüsü

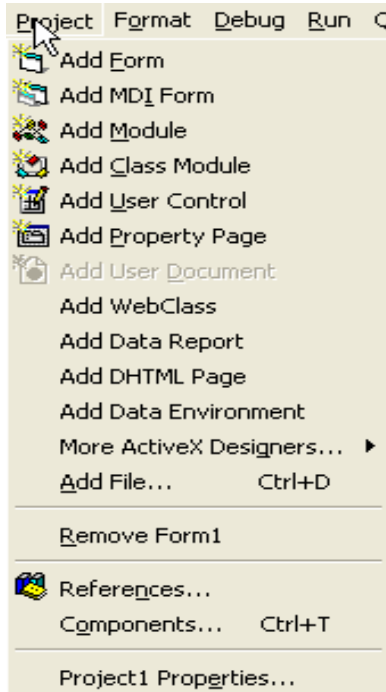
- Undo** : son yapılan işlemi geri alır
- Redo** : son yapılan işlemi tekrarlar.
- Cut** : seçili olanları keserek panoya kopyalar
- Copy** : seçili olanları panoya kopyalar.
- Paste** : Daha önce kesilmiş yada kopyalanmış olanları yapıştırır.
- Paste Link** : Daha Önce kopyalanmış bağlantıları oluşturur.
- Remove** : Seçilen elemanı siler
- Delete** :Seçilen elemanı siler.
- Delete Table From Database**: Tabloyu veri tabanından siler.
- Select All**: Form üzerindeki ya da kod penceresindeki bütün elemanları seçilir.
- Select All Columns**: Bütün Kolonları seçer.
- Table** : Tablo Komutlarını Gösteren Bir Alt menü
- Find** : Kod penceresinde İstenilen bilgiyi Bulur.
- Find Next**: Bulunan parametreye göre bir sonrakini bulur.
- Replace** : Bulunan Bir Sözcüğü değiştirmeyi sağlar.
- Indent** : Satırı İçerden başlatır.
- Outdent**: Satırı dışardan başlatır.
- 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 deyim kullanım biçimini gösterir.
- Parameter Info**: Bir deyim parametrelerini gösterir.

View Menüsü:



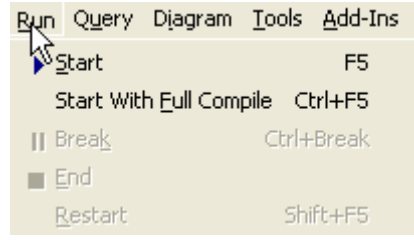
Code : Kod Penceresini görüntüler.
Object : Formu Görüntüler
Definition : Kod İçinde seçilen elemanın tanımlandığı yere gidilmesini sağlar
Last Position: Bir Önceki konuma gidilmesini sağlar.
Object Browser: Mevcut proje ya da Visual Basic nesnelerinin seçilmesini yardımcı
Immediate Window: Programın 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'ları gösterir
Project Explorer: Proje içinde yer alan elemanlar görüntülenir.
Properties Window: Özellikler penceresini görüntüler.
Form Layout Window: Formun ekrandaki görüntüsünü gösterir.
Property Pages: Özellikler tablosunu gösterir.
Table: Veritabanı, query gibi ortamlarda seçilen pencerenin açılıp kapanmasını sağlar.
Zoom: Büyütme
Show Panes: Bölümleri göster
Toolbox: araç kutusunun görüntülenmesini / görüntülenmemesini sağlar.
Data View Window:
Color Palette: Form üzerinde renklendirme yapmak için kullanılan renk paleti ekrana gelir.
Visual Component Manager:

Project Menüsü:



Add Form: projeye yeni form ekme yaparsınız.
Add mdi form: projenizi MDI form ekleyebilirsiniz.
Add Module: Projenize yeni module eklenir.
Add Class Module: projeye yeni class module eklenir.
Add Property Page: Property page eklenir. kullanıcı tanımlı kontrollerle birlikte o kontrolün özelliklerini belirleyecek property sayfaları da tanımlanabilir.
Remove: seçili olan dosya projeden çıkarılır.
Compenets: Tollbox penceresine yeni kontroller eklemek için kullanılır.
Project Properties: proje ile ilgili özellikler ayarlanır. bu pencerede örneğin birden fazla formunuz varsa program ilk çalıştığında hangi form ile çalışması gerektiğini ayarlayabilirsiniz.

Run Menüsü:



Run: projenizi çalıştırabilirsiniz.

Start With Full Compile: Start ile programlar çalıştırıldığında program içerisindeki hatalar yeri geldiği zaman uyarı mesajları ile yada programa kesilerek iletilir. ancak start with full compile seçeneğinde ise programın hataları ile birlikte denetlenir eğer hata yoksa program çalıştırılır.

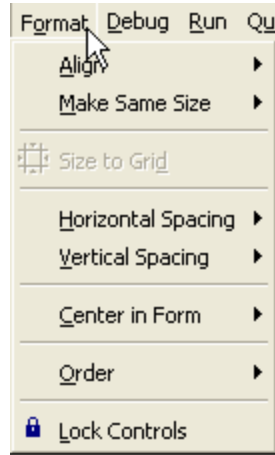
Break: çalışmakta olan program geçici olarak durdurulur. Programın kaldığı yerden devamı için F5 tuşuna basmanız

gerekir.

End: programın çalışması sona erer.

Restart: geçici olarak durdurulmuş program yeniden başlatılır.

Format Menüsü:



Aynı Hizaya Getirme İşlemleri:

Align: Form üzerindeki birden fazla kontrolü aynı hizaya getirmek amacıyla kullanılır. Alt menüleri ise:

Lefts: seçeneği ile seçili elemanlar sol taraf hizalanır.

Centers: seçili elemanların merkez noktalarını hizalar.

Rights: seçili elemanların sağ taraflarını hizalar.

Tops: Seçili elemanların üst taraflarını hizalar.

Middles: seçili elemanları üst üste getirerek orta kısımlarını hizalar.

Bottoms: seçili elemanların alt kısımlarını hizalar.

Boyutlarını Eşitleme İşlemleri:

Make Same Size:

Width: Seçili elemanları aynı genişlikte ayarlar.

Height: Seçili elemanları aynı yükseklikte ayarlar.

Both: Seçili elemanları aynı yükseklikte aynı genişlikte ayarlar.

Aralıklarını eşitleme İşlemleri:

Horizontal Spacing:

Make Equal: elemanların aralarındaki yatay boşlukları eşitler.

Increase: elemanların arasındaki yatay boşlukları artırır.

Decrease: Elemanların aralarındaki yatay boşlukları azaltır.

Remove: Elemanların arasındaki yatay boşlukları tamamen kaldırır.

Vertical Spacing:

Make Equal: elemanların arasındaki dikey boşlukları eşitler.

Increase: Elemanların arasındaki dikey boşlukları artırır.

Decrease: Elemanların arasındaki dikey boşlukları azaltır.

Remove: Elemanların arasındaki dikey boşlukları tamamen kaldırır.

Center In Form: form üzerindeki seçili elemanları formun ortasına ayarlar.

Horizontally: Seçili elemanları formun ortasına yatay olarak ortalar

Vertically: Seçili elemanları formun ortasına dikey olarak ortalar

Order: Form üzerindeki kontroller üst üste ise bunları alta yada üste alır. bir kontrol başka bir kontrolün altında kalarak gözüküyor ise bu menü kullanılır.

Bring to Front: alttaki elemanı üste alır.

Send to back: Üstteki elemanı alta alır.

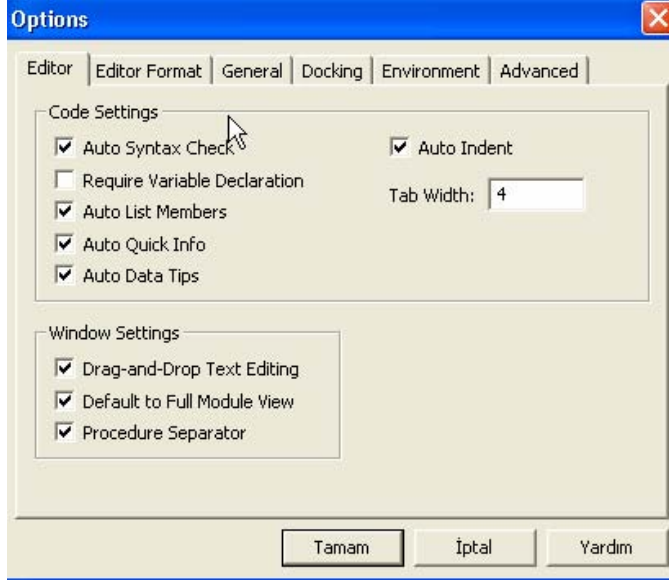
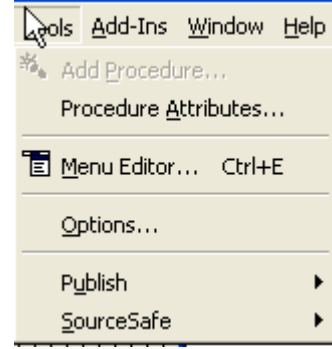
Lock Controls: Seçili elemanları kilitleyerek bunların yerlerinin değiştirilmemesini sağlar.

Tools Menüsü

Add Procedure: Kod penceresine alt program eklemek için kullanılır.

Menu Editör: Menü tasarım penceresine ulaşılır.

Options: üzerinde çalışılan pencere ile ilgili bir çok özellik değiştirilebilir. bu özellik tıklandığında aşağıdaki pencere açılır ve bu pencerede



☒ Auto Syntax Check

: kod penceresinde yazılan program kodları otomatik olarak yazım kontrolüne tutulur. yanlışlık halinde kullanıcı otomatik olarak uyarılır.

☐ Require Variable Declaration

: Seçeneği işaretli ise kullanılacak her değişkene tanımlama mecburiyeti getirilir.

☒ Auto List Members

: seçeneği işareti ise kontrol elemanların adı yazıldığında ilgili özellikler otomatik olarak bir liste halinde karşımıza çıkacaktır.

☒ Auto Quick Info

: İşaretli ise kod ortamında herhangi bir fonksiyon yazıldığında bu fonksiyonun içermesi gerektiği parametreler otomatik olarak gösterilerek kullanıcıya hatırlatmada bulunulur.

☒ Auto Data Tips

: Seçeneği işaretli ise debug modunda bir değişkenin üzerinde fare ile durulduğunda VB p değişkenin değerini gösterir.

☒ Auto Indent

: kod penceresinde bir satıra tab verildiğinde ondan sonra gelen satırlarda aynı hizada devam eder.

☒ Default to Full Module View

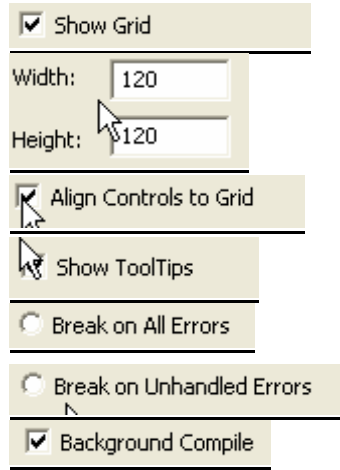
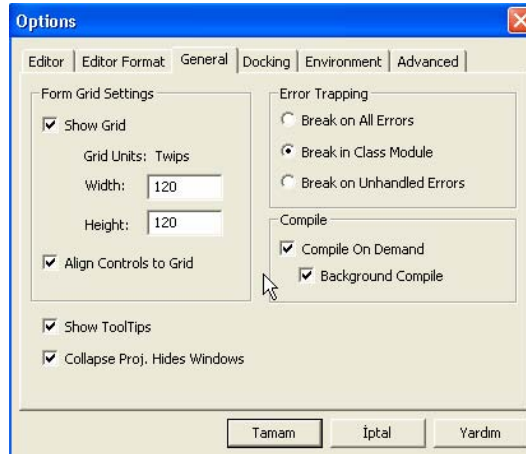
: kod penceresinde bütün kodlar aynı anda görülür. işaretli değilse sadece seçili olayın kodu gözükür.

☒ Procedure Separator

: işaretli ise kod penceresindeki prosedürler arasında çizgi gösterilir.

Editör Format Kısmı: kod penceresinde yazılan ifadelerin renkleri ve fon büyüklükleri ayarlanır. Foreground rengi ile yazı rengi background ile de zemin rengi ayarlanır.

General Kısmı:



Form üzerindeki noktalar gösterilir

Form üzerinde gösterilen noktalar arası boşluklar ayarlanır. eğer kontrolelerinizi hassas olarak yerleştirmek istiyorsanız bu değerleri küçültün.

Seçeneği işaretli ise form üzerindeki kontrollerin koordinatları noktalara denk getirilir.

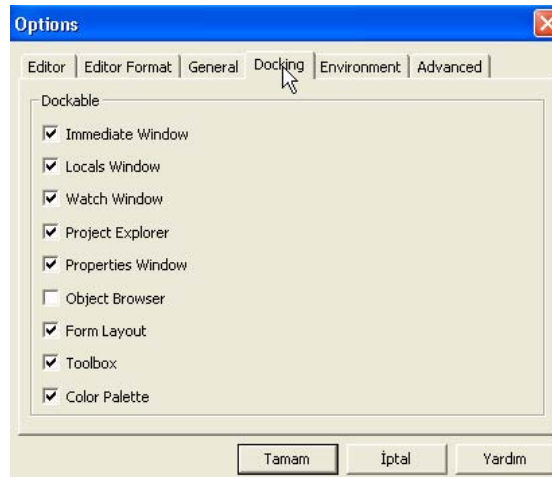
Bu eleman seçili ise fare kontrol üzerinde durduğunda kısa bir süre içerisinde o kontrol hakkında açıklama bilgisi gelir.

Derleyicinin herhangi bir hata ile karşılaşması durumunda programın derlenmesi kapatılır

Yakalanmamış hataların olması halinde programın derlenmesi durur.

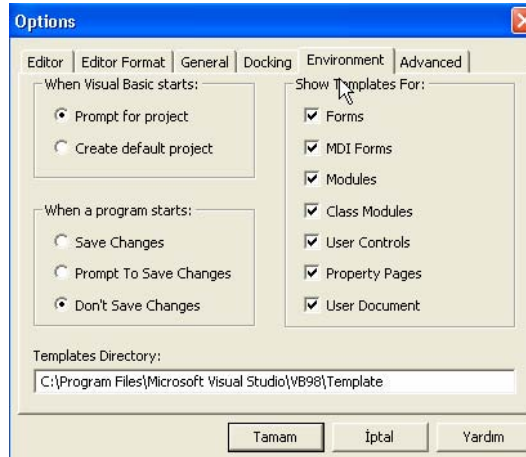
Derleme olaylı arka planda gerçekleştirilir.

Docking Kısmı:



Bu pencerede ise Vb ortamında kullanılan pencerelerin grup olarak gösterilip gösterilmeyeceğini belirler. işaretli olan seçenekler kendi grup pencerelerinde, işaretsiz olanlar ise ayrı pencerelerde gösterilir.

Environment Kısmı:



- ☒ Prompt for project
- ☐ Create default project
- ☐ Save Changes
- ☐ Prompt To Save Changes
- ☒ Don't Save Changes

Seçeneği işaretli ise VB açılırken hangi tip projeye başlayacağını soran pencere çıkar

Seçeneği işaretli ise Auto Load dosyasına kayıt edilen proje başlatılır

Program çalıştırılmadan önce yapılan değişiklikler kayıt eldir

Program çalıştırılmadan önce saklama sorusu sorulur

Program çalıştırılmadan önce kayıt edilmez.

Debug Menüsü

Bu menü yardımı ile program içerisindeki hatalar ayıklanabilir. bu hatalar mantık hatalarıdır yazım hataları değildir.

Step Info:

Program adım adım çalıştırılır.

Step Over:

Bu bölümde satır satır çalıştırılır. ancak procedür çağrıları da tek satırmış gibi düşünülür.

Add Watch:

Program çalıştırılırken izlenecek değişkenler bu bölümde eklenir. bu bölümde eklenen değişkenler debug penceresine gözüktür.



Break When Value True: seçeneği işaretli ise program çalışır ve eklenen değişkene sıfırdan farklı bir değer atandığı zaman programın çalışması kırılır.

Break When Value Changes: Seçeneği işaretli ise program çalışır ancak değişkenin değeri değiştiği anda programın çalışması kırılır.

Bu durumlar bazı değişkenlerin değeri istemediğiniz şekilde değişiyorsa kullanılabilir.

Run To Cursor

Program kursorün bulunduğu yere kadar çalıştırılır.

Quick Watch:

Bu seçenekle sadece seçilen değişkenin veya işlemin o andaki değeri öğrenilebilir.

Edit Watch:

Add Watch ile eklenen değişkenleri eklemek yada silmek amacıyla kullanılabilir.

Toggle Breakpoint

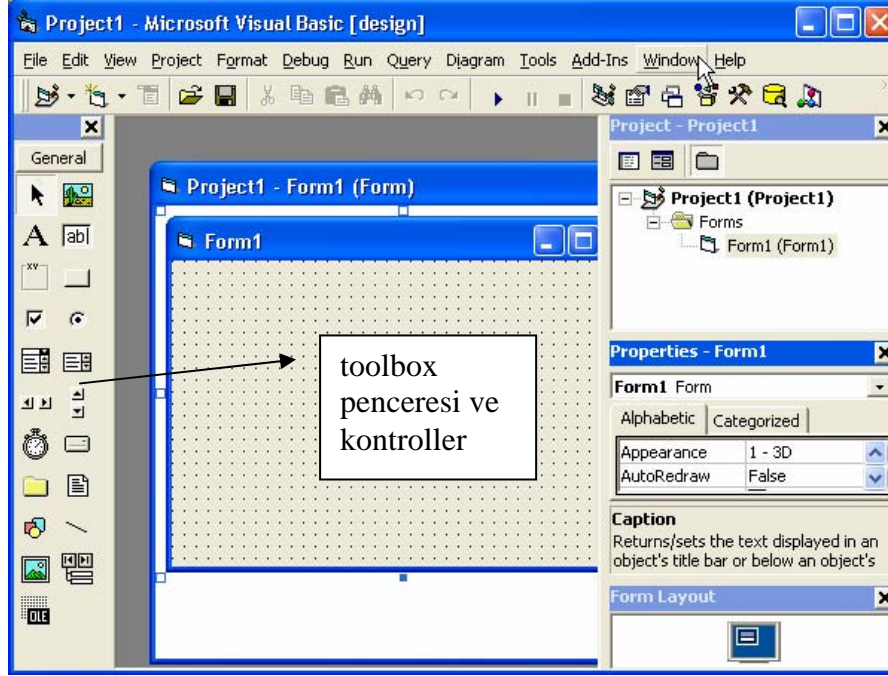
Programın bazı satırlarına Break Point kesmeleri konularak programın sadece o satırlara kadar çalışması sağlanır.

Clear All Break Points:

Tüm kesme noktaları silinir

KONTROLLER -1-

Visual dillerin en önemli özelliği kontrol elemanlarıdır. Artık bir kontrol için uzun uzadıya kontrol yazmak yerine kontrol elemanları kullanılmaktadır.



Toolbox penceresinde kontroller sadece gördüklerinizle sınırlı değildir. Bunlara ek olarak kontroller eklenebilir. Kontroller Windows altındaki System dizininde genellikle saklanırlar. (ocx uzantılı dosyalardır) . Toolboxda gözükmeyen kontrolleri eklemek için ise Project menüsündeki Components seçeneği yada toolbox bölümünde sağ tık ile Components seçeneği seçilir.

Kontroller eklenirken iki tip özelliği vardır.

- Properties özelliği
- Olay Yordamı özelliği

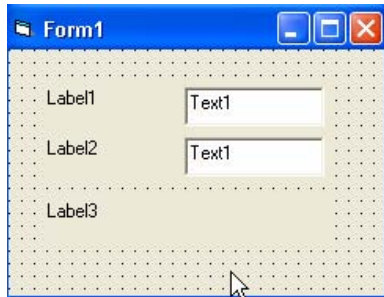
Bu özelliklerden properties özelliği tasarım ortamında yapılırken olay yordamı özellikleri kod penceresinde yazılırlar.



LABEL (ETİKET)

Form üzerinde kullanıcılara bilgi vermek amacı ile kullanılır. Bu kontrol text kutusu gibi bilgi girişi için kullanılmaz. ancak diğer özellikler bakımından text kutusuna çok benzer. bu kontrolün properties özelliğindeki özellikleri text kutusu ile aynı olduğundan text kutusu bölümünde bu özellikler anlatılacaktır. ancak şimdi farklı olan özellikleri inceleyelim.

örnek olarak text kutusu için etiket olarak kullanılabilir.



Properties

Caption

Label kontrolün üzerindeki yazı bu özellik ile belirlenir. tasarım ve kod penceresinde bu özellik kullanılabilir. özellikler penceresinin yanı sıra kod penceresinde ise aşağıdaki şekilde caption özelliği değiştirilir.

```
label1.caption = "Öğrenci No"  
label2.caption = "Öğrenci Adı"
```

eğer birden fazla satırda label kutusunu yazmak istiyorsanız satır geçişlerinde chr(10)+chr(13) satırları eklenir. chr(13) alt satır, chr(10) ise satır başı anlamındadır. örnek aşağıda verilmiştir.

```
label1.caption="Lütfen kayıtları saklamadan önce"+chr(13)+chr(10) +" Saklamayı unutmayınız"
```

Autosize

true ise nesnenin boyutları içeriğinin boyutlarına göre yeniden ayarlanır. içerik değiştiğinde boyutlar yeniden ayarlanır.

Backstyle

0 ise üzerinde bulunduğu nesnenin görülmesini sağlar
1 ise kendi zemin rengi üzerinde bulunur altındaki nesne görülmez..

TEXTBOX (METİN KUTUSU)

Bilgi girişi için kullanılan bir kontroldür. Bazı durumlarda Properties bölümünde bulunan özellikleri aynı zamanda kod penceresinde kod yazarak da değiştirebilirsiniz.

Properties (özellikler penceresindeki özellikleri)

Text

Text kutusunda girilen metin text özelliğine atanır.

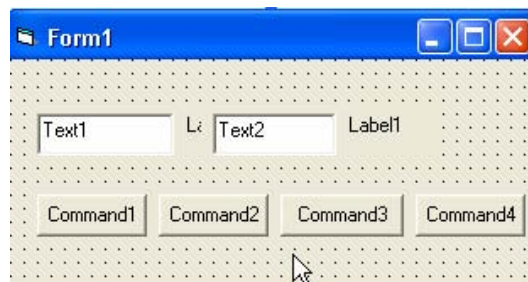
```
text1.text = "VB öğreniyorum"  
yada  
text1 = "VB öğreniyorum"
```

eğer text kutusundaki değerleri bir değişkene aktarmak istiyorsak bu durumda ise aşağıdaki şekilde kullanılır.

```
ad=text1.text  
soyad=text2.text  
ogr_no=val(text3.text)
```

yukarıdaki örnekte eğer aktarılacak değişken sayısal değer ise text özelliği alınırken val değişken türünü kullanmanız önerilir. Soru Neden?

ÖDEV: BASİT BİR HESAP MAKİNESİ PROGRAMI YAPINIZ. (FROM GÖRÜNÜMÜ AŞAĞIDA VERİLMİŞTİR.)



ToolTipText

Programlar çalışırken Mouse ile herhangi bir text kutusu üzerine gelindiğinde açıklama satırı vermek istiyorsanız kullanılır.

ÖDEV: Not Ortalaması hesabı yapan ve text kutuları üzerine gelindiğinde açıklama satırı veren programı yapınız.

MultiLine

text kutusu içerisine text özelliği ile yazılan metin eğer birden fazla satırda yazılsın isteniyorsa bu özellik true yapılır.

Alignment

nesne içerisindeki yazının sağa-sola –ortaya yazılmasını sağlar.

ScrollBars

MultiLine özelliği True (birden fazla satır) olması durumunda text kutusu etrafına kaydırma çubuğu eklenmesini sağlar. bu özelliğin 0 olması durumunda çerçeve yok,1 olması durumunda yatay çerçeve,2 olması durumunda dikey çerçeve, 3 olması durumunda yatay ve dikey çerçeve eklenecektir.

MaxLenght

Text kutusuna girilebilecek max karakter sayısıdır. bu kısımdaki değer 0 ise herhangi bir sınır yoktur.

PasswordChart

Text kutusu içerisinde yazılan metinlerin şifre karakter olarak yazılması isteniyorsa kullanılır. örneğin şifre için açılmış bir text kutusunda bu özellik * yapılırsa şifre girilirken metin gözükmeyecek yerine * karakterleri eklenecektir. (MultiLine özeliği false olması durumunda çalışır)

Locked

Text kutusunun Locked özelliği True ise text kutusunu kullanıcıya kilitlenir. kullanıcı ku alanda değişiklik yada ekleme yapamaz.

BorderStyle

Nesnenin ekran üzerindeki çerçeve şeklini belirler

0 : çerçeve yok

1 : çerçeve var

Appearancet

bu özellik kontrolün üç boyutlu gösterimini sağlar.

1: üç boyutlu gösterim

0: düz gösterim

SelStart,SelLenght

Kullanıcının text kutusu içerisinde seçme işlemini yapması için kullanılır. eğer kullanıcı text kutusunda seçme işlemi yapmış ise seçilen bölümü öğrenmek için bu kısım kullanılır.

SelText

Seçilmiş olan metin değiştirilmek isteniyorsa kullanılır.

ÖDEV: text kutusu içerisinde yazılan metni büyük harfe ve küçük harfe çeviren programı yazınız (büyük harf için kullanılan fonksiyon Ucase , küçük harf için çeviren fonksiyon Lcase dir.

HideSelectiont

True ise text kutusunda seçilen metnin, kontrol başka bir nesneye geçtiğinde seçilen kısmın gizlenmesini sağlar.

False ise kontrol başka bir nesneye geçse de seçilen kısmın görülmesini sağlar.

Enabled

Nesnenin aktif, veya pasif olmasını sağlar. False yani pasif olması durumunda kullanıcı nesneyi görür ancak herhangi bir işlem yapamaz.

Visible

True ise nesne görülür, False yapılırsa görülmez..

CausesValidation

text kutusundan başka bir kontrole geçildiğinde validate olayı meydana gelir. bu özellik ise validate olayının çalışıp çalışmamasını sağlar. bu özellik örneğin kullanıcının text kutusuna girdiği bilgiyi anında kontrol etmek için kullanılabilir.

ÖDEV: not giriş yapılırken 100 den büyük değer girilmemesini sağlayın

FontBold

true ise text kutusu içerisindeki yazı koyu olur.

FontItalic

true ise text kutusu içerisindeki italik olur.

FontStrikethru

true ise text kutusu içerisindeki yazının ortası çizilir.

FontUnderline

true ise text kutusu içerisindeki yazının altı çizilir.

FontName

nesne için kullanılan yazı tipini belirler.

FontSize

nesne için kullanılan yazının puntosunu belirler. maksimum 2048

Font

Yukarıdaki yazılan özelliklerin hepsini tek bir defada yapmak için font penceresini açar.

BackColor

True ise Text kutusu nesnenin zemin rengini değiştirir.

ForeColor

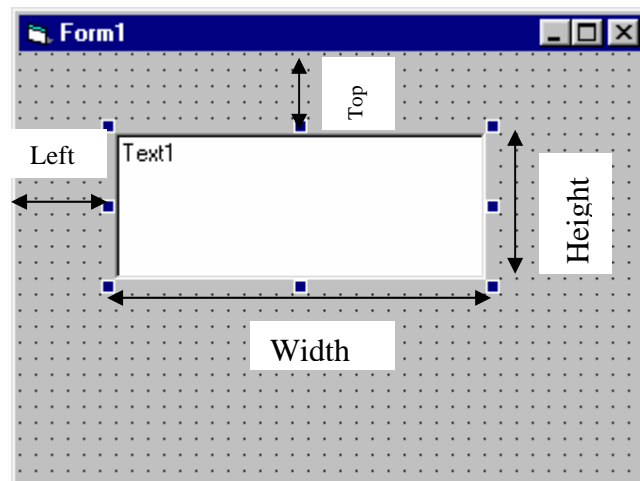
Yazının rengini belirler. Yukarıda anlatılan renk bileşimleri burada da geçerlidir.

Left,Top

Nesnenin Sol ve üst noktalarının koordinatlarını belirler.

Height,Width

Nesnenin boyu ve enini belirler



HelpContextId

Kontrolle ilgili yardım dosyasındaki konu numarası bu özellik ile belirlenir.

Index

Aynı isimli birden çok nesne oluşturulmuşsa VB bu nesneleri bir dizi olarak görür. Index parametresi bu nesnenin dizideki kaçınca eleman olduğunu belirler, bu nesneye bu dizi indexi ile ulaşır.

MousePointer, MouseIcon

Mouse göstergesinin nesne üzerine geldiğinde ulaşacağı şekli belirler. 0- 15 arası bir sayıdır. 0 normal hali, 11 bekleme hali vb.

TabStop

True ise kullanıcı bu nesneye Tab tuşuyla ulaşabilir. False ise tab tuşuyla bu kontrol üzerine gelinmez, mouse ile yazılım yoluyla veya varsa kısayol tuşuyla gelebilir.

TabIndex

Form üzerinde kullanıcının ulaşabileceği her nesnenin tabindexi vardır. TabIndex kullanıcının tab tuşuyla kontroller arasında dolaşırken bu kontrollerin sıralamasını belirler.

Tag

Bu özelliğin Visual Basic için hiçbir anlamı yoktur. Kullanıcı bu parametreyi bir değişken gibi kullanabilir. Bu özellik bir string değişken olarak kullanılmalıdır.

hWnd

Windows altında çalışan kontrollerin handle diye adlandırılan tanıtıcı bir numarası vardır. Windows bu numarayı kullanarak kontrolleri tanıtır.

Parent

Nesnenin üzerinde bulunduğa Form'a ulaşmayı sağlar. Nesnenin isminden sonra verilen parent özelliği o nesneyi değil nesnenin üzerinde bulunduğu formu temsil eder.



COMMAND BUTTON (KOMUT DÜĞMESİ)

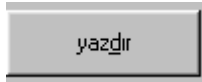
Bir olayın kullanıcı tarafından başlatılması için programlarda çok kullanılan kontrollerden biridir.

PROPERTIES

Caption

Komut düğmesi üzerinde yazılacak olan mesajı içerir.

Command1.Caption="Yazdır "



Style

Bu özellik komut düğmesinin yazılı mı, resimli mi olacağını belirler.

0 ise komut düğmesi üzerinde caption özelliği ile belirlenen yazı bulunur.

1 ise komut düğmesinin üzerinde Picture özelliği ile belirlenen resim bulunur.



Picture, DisabledPicture, DownPicture

Eğer komut düğmesinin style özelliğine 1 verilerek resimli bir komut düğmesi olacağı belirlenmişse bu üç özellik ile komut düğmesi üzerinde bulunacak resim belirlenebilir.

Default

Bir komut düğmesinin default özelliği True ise o düğmenin bulunduğu form üzerinde entere basılması durumunda o düğme tıklanmış gibi olur.

Cancel

Bu özellik de default özelliği gibidir ancak ESC tuşu ile aktif hale gelir.

Events**Click()**

Komut düğmesinin en önemli olayı budur. Düğme tıklandığında yapılması istenen işlem bu olaya yazılır.

BÖLÜM 5

5. VB'DE DEĞİŞKEN TANIMLAMA

GW ve Quick Basic'te programcı ne zaman bir değişkene gereksinim duyarsa her hangi bir bildiri deyimini kullanmadan değişkeni tanımlayıp kullanabilir. Bu değişkene aktarılabilecek bilgiler, değişkenin adının son karakterine bağlı olarak değişiyordu.

VB'de ise değişken yapısı daha da gelişmiştir. Değişkenler için önemli durum, değişkenlerin karışıklığa meydan vermemesi için tanımlanması mecburiyeti olmasıdır. Daha önceki iki kuşak Basic'te bu mecburiyet yoktur ve değişkenleri benzerlikten dolayı her zaman karıştırmak mümkündür. VB'de değişkenleri kullanmadan önce tanımlama mecburiyeti programcının seçimine bırakılmıştır. Programcı isterse Tools-Options menüsü Require Variable Declaration seçeneğini işaretleyerek derleyicinin tanımlanmamış değişken kullanımda hata mesajı vermesini sağlayabiliriz.

Bir değişken tanımlanırken aşağıdaki kuralları göz önünde bulundurulması gerekir.

- Değişken ismi bir harf ile başlamalıdır.
Ad1, Ad2
- Değişken isminde boşluk bulunmaz. Bunun yerine alt çizgi karakteri kullanılabilir.
- Değişken isminde sadece harfler, rakamlar ve alt çizgi karakterleri bulunabilir.
Adı_Soyadı, No1
- Değişkene verilecek isim VB komutlarından oluşmamalıdır.
Not, Dim, Valvs
- Değişken ismi 255 karakterden fazla olmamalıdır.

5.1. VB'de Kullanılan Standart Veri Tipleri

VB'de değişken tipleri Dim deęimi ile tanımlanır. Bu deęimin yazılımı;

Dim deg_adı [(Dizi boyutu)] As tipi

deg_adı : Tanımlanacak deęişkenin ismidir.

Dizi boyutu : Deęişken dizi olarak tanımlanacaksa dizinin boyutu deęişken isminden sonra parantez içine yazılır.

Dim x (5) tanımıyla 6 elemanlı dizi tanımlamış oluruz. Dizinin ilk elemanı 0 son elemanı 5 dir.

Dim x (3 to 6) tanımıyla ilk elamanı x (3) ve son elamanı x (6) olan bir dizi tanımlanmıştır.

Tipi : Tanımlanacak deęişkenin tipi belirlenir. Bunlar ;

Byte : 1 Baytlık tamsayı tipidir. 0 ile 255 arasında değer alabilir.

Integer : 2 Baytlık işaretli tamsayı tipidir. -32768 ile 32767 arasında değer alabilir.

Long : 4 Baytlık işaretli tamsayı tipidir. -2.147.483.648 ile 2. 147.483.647 arasında değer alabilir.

Not: Yukarıdaki değişken tiplerine ondalık sayı atanması durumunda sayı en yakın tamsayıya yuvarlatılacaktır. (x=4,3 ise x= 4 atanır, y=3.6 ise y=4 atanır)

Single : 4 Baytlık ondalık sayı tipidir. $+ 3.402823 \times 10^{38}$ ile $+ 1.401298 \times 10^{-45}$ arasında değer alabilir.

Double : 8 Baytlık ondalık sayı tipidir. $+ 1.79769313486232 \times 10^{308}$ ile $+ 4.94065645841247 \times 10^{-324}$ arasında değer alabilir.

Currency : 8 Baytlık ondalık sayı tipidir. Ancak sayının ondalık kısmı 4 basamaktan fazla olamaz. -92337203685477.5808 ile 922337203685477.5807 arasında değer alabilir.

Decimal : 14 Baytlık bir sayı tipidir. Bu tipin en önemli özelliği sayıdaki bütün basamakların tutulmasıdır.

Boolean : 2 Baytlık bir veri tipidir. Sadece True ve False değerleri alabilir.

Dim ogrenci As Boolean	}	Bu atamaların hepsi aynı işi görür.
Ogrenci=True		
Ogrenci=1		
Ogrenci=100		

String : Karakter sınırı verilmezse 2 milyar karaktere kadar atama yapılabilen sayısal veri tipidir. +10 bayt yer kaplar. Karakter sayısını sınırlamak için ;

Dim a As String * [sınır] yazılır.

Dim a String İstenildiği kadar karakter ataması yapılabilir.
Dim b String * 3 3 tane karakter ataması yapılabilir.

Variant : VB'de değişken tanımlanırken tip ismi verilmemişse bu tip Variant olarak ele alınır ve değişkenin tipi atanacak değere göre değişir. Variant tipler yukarıdaki tiplerden herhangi biri gibi işleme girebilir ve programın çalışması esnasında değişebilir.

Date : 8 Baytlık yer kaplayan bu değişken tipi 1/1/100 ile 31/12/9999 arasındaki tarih ve 0:00:00 ile 23:59:59 arasında saat atamaları yapılabilir.

Object : 4 Baytlık bir veri tipidir. OLE işlemlerinde kullanılmak amacıyla tasarlanmıştır. Bu tipte değişkenlere atama yapılırken Set operatörü kullanılır.

5.2. Kullanıcı Tarafından Yeni Tip Tanımlanması

```
Type tip_adı
    Deg_adı As tipi
    Deg_adı As tipi
    .....
    .....
    .....
End Type
```

Bu şekilde tanımlanan tipte bir değişken tanımlamak için de;

Dim deg_adı As tip_ismi olarak tanımlanabilir.

Tip tanımı bir form içerisinde ancak Private olarak tanımlanabilir. Yani tanımlanan tip sadece o formda kullanılır. Genel yani Public bir tip ise ancak bir modül içerisinde kullanılabilir.

```
Bir formda tip;
    Private Type Ad
        .....
    End Type
```

 şeklinde tanımlanır.

Örnek 5.1:

```
Type Rehber
    Ad As String
    Soyadı As String
    Tel As Integer
    Adres As String
End Type
```

Burada Rehber değişkenini değil Rehber tipini tanımladık. Artık Rehber yapısındaki değişkenleri tanımlayabiliriz.

Dim adı As Rehber şeklinde tanımlayabiliriz.

Bu değişkene değer atamak için “.” operatörünü kullanalım ;
Adı.Ad = “Fırat Üniversitesi”

5.2.1. Enum

```
Enum Tip İsmi
    Eleman1=Sayı1
    Eleman2=Sayı2
    .....
    Eleman3=SayıN
End Enum
```

Bu tip her bölüme bir numara verir. VB bu bölümü bir sayı olarak kullanmasına rağmen bunu bir string gibi kullanma imkanı verir.

Örnek 5.2:

```
Enum bolum
    Bilgisayar=1
    Elektronik=2
    Makine=3
    Metal=4
    Yapı=5
    YapıRes=6
End Enum
```

Tanımını yaptıktan sonra bolum tipinde değişkenler tanımlayarak bu değişkenlere değer atamak mümkündür.

```
Dim eğitim As bolum
Bolum=Bilgisayar
```

Satırı ile bolum değişkenine Bilgisayar atamaktayız. Bu işlem eğitim=1 satırı ile aynı işi yapmasına karşın anlaşılması daha kolaydır.

5.3. Dinamik Dizi Tanımlama

Programda tanımlanacak dizinin boyutu her zaman belli olmayabilir. Bu durumda diziye dinamik olarak tanımlamamız gerekir. Diziye dinamik olarak tanımlayabilmek için ReDim değişimi kullanılır. Diziye dinamik olarak tanımlamak demek dizinin boyunun ihtiyaç olduğunda artırılması veya azaltılması demektir. Üstelik dizinin eski halini koruyarak.

ReDim [Preserve] deg_adı (dizi boyutu) As tipi

ReDim ile dizinin ilk defa tanımlanacağı gibi yine ReDim'le tanımlanmış dizinin boyutunu değiştirmek için de kullanılabilir. Burada Dim'den farklı olarak Preserve parametresi vardır.

Preserve, dizinin yeniden boyutlandırılması durumunda dizide bulunan eleman içeriklerini yeni dizide de bulunmasını sağlar.

Tanımlanacak dizi local bir dizi olmayacaksa dizi öncelikle Dim veya Global değimiyle aşağıdaki gibi tanımlanmalıdır.

Dim deg_adı () As tipi

Görüldüğü dizi tanımlanırken parantez içerisinde dizi boyutu verilmemiştir. Bu Visual Basic' e dizinin dinamik olarak tanımlanacağını belirtecektir. Daha sonra dizinin elaman sayısı ReDim değimi kullanılarak belirlenir.

```
Private Type takip
    Ad As String * 25
    Soyad As String * 25
    Sınıfı As String * 2
    No As String *6
End Type
```

```
Dim Ogr () As takip "Diziyi dinamik olarak tanımla
Dim dizi_boyu As Integer 'En son tanımlanan dizinin boyutu
Dim eleman_say As Integer 'Dizideki son eleman indeksi
```

5.4. Global ve Local Değişkenler

Global değişkenler programın bütün fonksiyonları kullanılabilen değişkenlerdir. Local değişkenler ise sadece tanımlandıkları fonksiyonda kullanılabılırler.

Dim değimi ile yapılan tanımlamalar yerine Global ve Local'dır. Dim ile tanımlana değişkenler;

- General-Declarations kısmında tanımlanmışsa modül veya form içerisinde Global'dır. Buna VB'de Modüle Level denir.
- Bir fonksiyon veya alt programda tanımlanmışsa Local'dır. Sadece o prosedür tarafından kullanılabılır.
- Bir formun General Declarations kısmında Public değimi ile tanımlanmışsa diğer modüller ve formlarda bu değişkeni kullanabilirler. Ancak değişkenin isminden önce modülün/formun ismini vermesi gerekir.
Form1 . x = 4
- Bir değişkeni programın her tarafında kullanılacak şekilde tam olarak global tanımlamak için o değişkeni bir modülün General-Declaration kısmında Public değimi ile tanımlamak gerekir.
Public x

Local değişkenlerin önemli bir avantajı bellekte fazladan yer işgal etmemeleridir (Static olarak kullanılmamışlarsa). Global değişkenler program çalıştığı sürece bellekte tutulmalarına rağmen local değişkenler, değişkenin bulunduğu fonksiyona girince değişken belleğe alınır ve fonksiyonun çalışması bitince bellekten atılır.

5.5. Static Değişkenler

Local olarak tanımlanan bir değişken tanımlandığı alt program veya fonksiyonun çalışması bittikten sonra bellekten atılır. İlgili prosedür ikinci kez çalıştırıldığında Local değişkenler eski değerlerinden değil baştan başlarlar.(string ise “”, sayı ise 0)

Local olarak tanımlanan bir değişkenin değerinin tanımlandığı alt program veya fonksiyonun çalışması bittikten sonra da değerinin korunması isteniyorsa Dim yerine Static değimiyle tanımlama yapılır. Static değimi bir alt program veya fonksiyon altında tanımlanabilir:

Static deg_adı [(dizi boyutu)] As tipi

5.6. Sabit Tanımlama

Programda değerlerinin değişmesi istenmeyen değerler sabit olarak tanımlanır. VB sabit tanımlama;

Cost Sabit ismi=Değeri

Örnek 5.3:

Cost PI=3.1415

5.7. Değişkenlere İlk Değer Ataması

Local değişkenlere ilk değer ataması şu şekildedir.

- Dim değimiyle ile tip ismi kullanılmadan yapılan tanımlamalarda ilk değer ataması direk olarak tanımın yapıldığı satırda yapılabilir.
Dim x=5, y=10
- Tip ismi kullanılarak yapılan tanımlamalarda ise ikinci bir satırda ilk değer ataması yapılır.

Dim x As Integer
x=5

Global olarak tanımlanan değişkene ilk değer ataması Declarations kısmında yapılır ve burası bir alt program değildir. Dolayısıyla Global olarak tanımladığımız değişkenlerimize ilk değer atamasını Form_Load veya projenin çalışmaya başladığı alt program altında yapmalıyız.

Static değişkenlere de ilk değer ataması IsEmpty () fonksiyonu ile yapılır. Bu fonksiyon bir değişkene değer atanıp atanmadığını kontrol eder. Verilen değişkene değer ataması yapılmamışsa fonksiyondan geri dönen değer TRUE olur.

Fonksiyonun bu özelliğinden yararlanarak static olarak tanımlanan değişkene daha önce değer atanıp atanmadığını yani ilk defamı kullanıldığını öğrenebiliriz.

**Static i;
If IsEmpty(i) then i=5**

Böylece alt program ilk defa çalıştığında i'ye her hangi bir değer ataması olmadığı için şartımız çalışacak ve ilk değer atanacaktır. Alt programın sonraki seferlerde çalışması durumunda ise i'nin static olmasından dolayı atanmış bir değer olacak ve şartımız çalışmayacaktır. Böylece sadece alt program ilk defa çalıştığında ilk değerimizi atamış olacağız.

VB'de bir diziye değer ataması da değişkeni Variant olarak tanımladıktan sonra Array fonksiyonu ile o değişkenin bir dizi olduğunu belirtmek ve ilk değer atamak mümkündür.

**Dim Dizi [As Variant]
Dizi=Array(eleman)**

Örnek5.4:

Dim gün
Gün= Array("Pazar", "Pazartesi", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi")

5.8.Kullanılan Değişkenlerin Tipini Belirlemek

Tipi belirtilmeyen Variant tipi değişkenlere programın işletimi sırasında istenen tipteki bilgi aktarılabilir. VarType fonksiyonu ile Variant bir değişkenin hangi tipte bilgi içerdiği öğrenilebilir.

VarType (Variant Değişken Adı)

Yazılımında da görüldüğü gibi bu fonksiyon dışarıdan içerdiği bilginin tipi öğrenilmek istenen değişkenin adını almaktadır. Geriye ise değişkenin içerdiği bilginin tipini temsil eden sayısal bir değeri döndürmektedir. Eğer Variant değişken karakter sel bilgi içeriyorsa VarType() fonksiyonu geriye 8 sayısal değerini döndürür. VarType fonksiyonu ile döndürülecek sayısal değer ve bu değerlerin temsil ettikleri veri tipleri aşağıdaki tabloda verilmiştir.

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

5.9. Tip Değiştirme İşlemleri

Ccur(ifade): Parantez içinde verilen ifadeyi Currency tipine dönüştürür.

Cdbl(ifade): Parantez içinde verilen ifadeyi Double tipine dönüştürür.

CInt(ifade): Parantez içinde verilen ifadeyi Integer tipine dönüştürür.

CLng(ifade): Parantez içinde verilen ifadeyi Long tipine dönüştürür.

CSng(ifade): Parantez içinde verilen ifadeyi Single tipine dönüştürür.

CVar(ifade): Parantez içinde verilen ifadeyi Variant tipine dönüştürür.

CBool(ifade): Parantez içinde verilen ifadeyi Boolean tipine dönüştürür.

CByte(ifade): Parantez içinde verilen ifadeyi Byte tipine dönüştürür.

CDec(ifade): Parantez içinde verilen ifadeyi Decimal tipine dönüştürür.

CDate(ifade): Parantez içinde verilen ifadeyi Date tipine dönüştürür. İki tarih arasında işlem yapabilmek için bu dönüşüm kullanılır.

5.10. Değişkenlerle İlgili Komutlar

5.10.1. IsDate

Değişkenin içeriği geçerli bir tarih ise geri dönen tarih True, değilse False'dir kabul edilen tarih formatı Windows'da tanımlanan formattır.

IsDate (değişken)

Örnek 5.4:

Dim a

a=IsDate("30/01/1999") ' Verilen tarih geçerlidir. A=True (Ocak ayı 30 veya 31 gündür.)

a=IsDate("13bfg") 'Verilen tarih geçerli değildir. A=False

5.10.2. IsNumeric

Verilen değişkenin içeriğinin bir sayı olup olmadığını kontrol eder. Değişkenin içeriği sayı ise geri dönen değer True değilse False'dir.

IsNumeric(değişken)

5.10.3. IsArray

Girilen değişken bir dizi ise True değilse false üretir. Yani bir değişkenin dizi olup olmadığını öğrenmek için kullanılır.

IsArray(değişken)

5.10.4. IsEmpty

Tanımlanmış bir değişkene henüz bir değer atanmamışsa o değişken Empty'dir. Bir değişkene değer atanıp atanmadığını öğrenmek için kullanılır. Dönen değer False ise değişkene bir atama yapılmıştır.

IsEmpty (Değişken)

5.10.5. IsMissing

VB'de fonksiyon veya alt programlara giriş parametresi olarak Optional değimi kullanılırsa o parametre kullanılmadan da o fonksiyon çağrılabilir. IsMissing fonksiyonu bu tip parametrelerin verilip verilmediğini kontrol etmek için kullanılır.

IsMissing (Değişken)

5.10.6. IsObject

Verilen değişkenin tipi Object tipinde ise fonksiyondan geriye True değeri döner.

IsObject (Değişken)

5.10.7. Erase

İsmi verilen diziyi siler.

Erase Diziismi

BİLGİ GİRİŞ VE MESAJ PENCERELERİ (Msgbox ve Inputbox pencereleri)

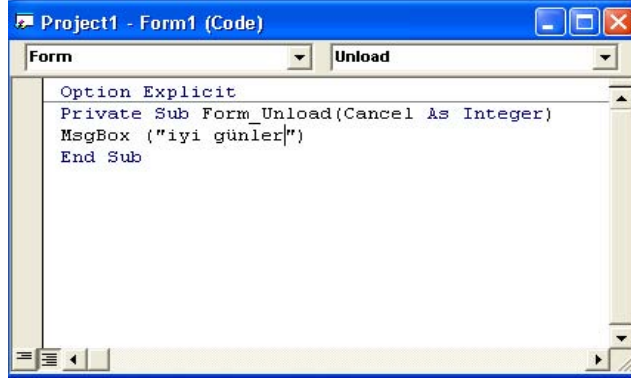
MSGBOX (MESAJ KUTULARI)

Bir programda ve Windowsta en aşına olduğumuz şeylerden birisine programın akışı sırasında bilgi vermek veya onaylatmak amacıyla karşımıza çıkan mesaj pencereleridir. Örneğin programdan çıkarken **kayıt edilsin mi?** gibi bir soru ve bu soruya verilen **evet hayır iptal** seçenekleri olan onay kutuları gibi.

Eğer msgbox ile sadece kullanıcıya mesaj verilmek isteniyorsa aşağıdaki gibi kullanılır.

Msgbox("iyi günler")

Örneğin programınız kapatıldığında bu mesajı vermek istiyorsanız programınız form olayında unload olayına aşağıdaki kodu yazmanız gerekir.



Mesaj pencereleri sadece kullanıcıya mesaj vermek için değil de çoğu zaman onay almak içinde kullanılır. Kullanıcının mesaj kutularında verebileceği cevaplar için uygun düğmeleri seçmek ve seçilen düğmelere kullanıcının verdiği cevaba göre işlem yapmak için msgbox fonksiyonu aşağıdaki formatta kullanılır.

Cevap=msgbox(mesaj,[tip],[pencerebaşlığı],[helpfile,context]

Komut formatına göre parametreler şunlardır.

Mesaj: Kutu içerisine yazılması istenilen mesaj

Pencere Başlığı: Pencerenin başlığına yazılacak metin

Helpfile,Context .Bu iki özellik yardım dosyası ismi ve bir konu numarasını belirler. Bu sayede kullanıcı mesaj kutusunda yardım istediğinde ilgili help dosyası açılır ve ilgili konu gösterilir.

Tip: Pencerenin içine konacak seçeneklerdir,bunlar iconlar, butonlar, gibi seçeneklerdir.bir tip belirtilirken aşağıdaki format kullanılır.

Tip=secenek+icon+varsayılan+oncelik

Tip formatında **secenek** en önemli parametredir. Seçenek bölümünde msg kutusu üzerinde olması istenen butonlar seçilir. (örneğin evet – hayır , yada evet –hayır-iptal vb) bu butonlar ve diğer seçenekler aşağıdaki tabloda verilmektedir. . ikinci parametre olan **icon** ise msgbox penceresinde kullanıcının dikkatini çekmek için kullanılan resimdir. **Varsayılan düğme** ise seçenek bölümünde belirlenen butonlardan hangisinin öncelikli aktif olacağıdır. Örneğin silme butonunda evet hayır eklendi ise yanlış işlem yapılmaması için varsayılan düğme hayır seçilir. **Öncelik** parametresi ise mesaj kutularında herhangi bir işlem yapılmadan diğer uygulamalar geçiş yapılmamasını sağlar. Öncelik olarak 4096 verilmesi durumunda mesaj kutusunda herhangi bir işlem yapılmadan diğer uygulamalara geçiş yapılamaz.

Bu dört ayrı parametre toplanarak tip parametresini oluşturur.

SEÇENEK		
0	Vnokonly	TAMAM
1	Vboncancel	TAMAM / İPTAL
2	Vbabortretryignore	DURDUR / YENİDEN DENE / YOKSAY
3	Vbyesnocancel	EVET / HAYIR / İPTAL
4	Vbyesno	EVET / HAYIR
5	Vbretrycancel	YENİDEN DENE / İPTAL
İCON		
16	Vbcritical	X iconu
32	Vbquestion	? iconu
48	Vbexclamation	! iconu
64	Vbinformation	İ bilgi iconu
VARSAYILAN		
0	Vbdefault1	Birinci düğme
256	Vbdefault2	İkinci düğme
512	Vbdefault3	Üçüncü düğme
ÖNCELİK		
4096	Vbapplicationmodal	System modal
0	Vbsystemmodal	normal

Yukarıdaki tablodaki parametrelere göre msgbox kutusu oluşturulur. MsgBox kutusu oluşturulurken dikkat ederseniz **cevap** değişkeni kullanılmıştır. Bu değişkenin görevi ise msgbox kutusunda butonlardan dönen değerdir. Bu dönen değer bir tamsayı değerdir ve bu değerler ise;

SEMBOLİK	SAYISAL	ANLAMI
Vbok	1	Tamam düğmesi seçildi
Vbcancel	2	İptal düğmesi seçildi
Vbabort	3	İşlemi durdur düğmesi seçildi
Vbretry	4	Tekrar dene düğmesi seçildi
Vbignore	5	Gözardı et düğmesi seçildi
Vbyes	6	Evet düğmesi seçildi
Vbno	7	hayır düğmesi seçildi

Örnekler:

1. kullanıcıya sadece hoş geldiniz mesajı vermek için mesaj kutusunu ayarlayınız.

Msgbox "hoş geldiniz"



2. Kullanıcıya sadece hoş geldiniz mesajı ve bu mesajı ikon olarak information iconu ile görüntüleyen mesaj kutusunu yazınız.

Msgbox "hoş geldiniz",vbinformation



3. Kullanıcıya sadece hoş geldiniz mesajı ve bu mesajı ikon olarak information iconu ile Ve pencere başlığı olarak merhaba yazan mesaj kutusunu yazınız.

MsgBox " hoş geldiniz",vbInformation,"merhaba"



4. Evet hayır iptal butonları olan , mesaj olarak programı kapatmak istiyormusunuz mesajı olan, pencere başlığı olarak Programı kapat yazan ve icon olarak x simgesi olan mesaj kutusunu yazınız.

cevap = MsgBox("programı kapat", vbCritical + vbYesNoCancel + vbDefaultButton2 + 4096,"Programı")



yukarıdaki örnekte eğer kullanıcı evet tuşuna bastı ise yapılması istenen programı kapatmaktır. Şimdi msgbox kutusunu şimdi bu şekilde yazalım.

cevap = MsgBox("programı kapatmak istiyormusunuz", vbCritical + vbYesNoCancel + vbDefaultButton2 + 4096, "Programı Kapat")

' yukarıdaki satır tek satır olarak yazılacaktır.

If cevap = 6 Then

End

Else

Cancel=true ' çıkışı iptal eder

End If

INPUTBOX (Bilgi Giriş Kutusu)

Kullanıcının bazı durumlarda değişkenlere değer girmesi gerekir örneğin öğrenci ismi , öğrenci numarası gibi, bu durumlarda inputbox kutusu geliştirilmiştir.

Cevap=Inputbox(mesaj,[başlık],[varsayılan değer],[x],[y],[helpfile,context])

Mesaj:

Girilmesi istenen değer için açıklama veya soru

Başlık:

Pencere başlığı

Varsayılan değer:

Değer giriş kutusunda bulunması istenen değer

x,y:

Pencerenin sol üst köşesinin x ve y koordinatlarını belirler.

Örnekler:

1. Klavyeden girilen sayı değeri 10 dan küçük ise 10dan küçük yazan , 10 dan büyük ise 10 dan büyük , 10 a eşit ise 10 eşittir yazan programı yazınız.

cevap = InputBox("sayı değerini giriniz", "Sayı", "10", 0, 0)

If cevap > 10 Then

MsgBox "sayı 10 dan büyüktür"

Elseif cevap < 10 Then

MsgBox "sayı 10da küçüktür"

Elseif cevap = 10 Then

```
MsgBox " sayı 10 ' a eşittir"  
End If
```

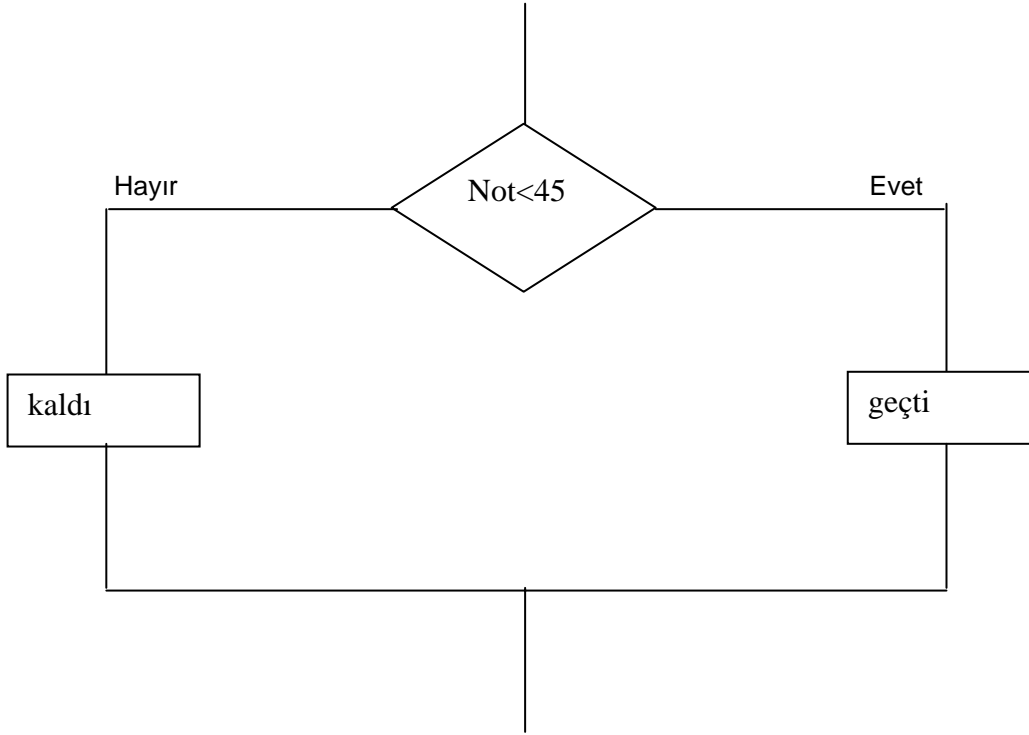
2. Klavyeden 10 adet öğrenci isimlerini ve numaralarının giren ve sonuçları ekrana yazan program satırlarını yazınız.

```
Dim ogrsay, i  
ogrsay = InputBox("öğrenci Sayısını giriniz", "GİRİŞ", 10)  
ReDim adi(ogrsay) As String * 30  
' öğrenci sayısı kadar adi isimli 30 karakter uzunluğunda dizi tanımlandı  
ReDim notu(ogrsay) As String * 5  
' öğrenci sayısı kadar notu isimli dizi tanımlandı  
For i = 1 To ogrsay  
    adi(i) = InputBox(i & " numaralı öğrencinin adı:")  
    notu(i) = InputBox(i & " numaralı öğrencinin notu:")  
Next  
Show  
For i = 1 To ogrsay  
    Print adi(i), notu(i)  
    If notu(i) > 45 Then  
        Print "geçti"  
    Else  
        Print "kaldı"  
    End If  
Next
```

PROGRAM KONTROL DEYİMLERİ

If Then Else Yapısı:

If şart yapısı bütün programlama dillerinde olan bazı şartların gerçekleşmesi durumunda veya gerçekleşmemesi durumunda programın akışını değiştiren yapılardır. Programlar çalıştırılırken satır satır incelenirler ve bunu da akış şeması olarak düşünersek, eğer herhangi bir satırda; örneğin bir not ortalaması alınmış ve bu not ortalamasına göre not ortalaması <45 ise kaldı yazmasını ≥ 45 ise geçti yazmasını istiyoruz. Bu durumu akış şeması üzerinde gösterirsek:



If yapısı ile karar verme

Şimdi if yapısını ve kullanım çeşitlerini inceleyelim.

1. Sadece şartın gerçekleşmesi durumu

```
if şart then  
    komutlar  
endif
```

Burada şart verilecek ve eğer şart doğru ise yani şart gerçekleşiyorsa komutlar işletilecektir. Eğer şart gerçekleşmiyorsa program kaldığı yerden akışına devam edecektir. if yapısı içerisindeki komutlar icra edilmeyecektir.

Örnek: eğer voltaj <220 ise ekrana voltaj düştü mesajı veren programı yazınız.

```
if voltaj <220 then  
    msgbox "voltaj 220 voltan küçük "  
endif
```

DİKKAT: if yapısı kullanıldığında if satırlarının bittiğini anlatan endif mutlaka kullanılmalıdır. Aksi takdirde programınız hata verir.

2. Şartın gerçekleşmesi veya gerçekleşmemesi durumu

```
if şart then
```

```
        Komutlar1
Else
        Komutlar2
endif
```

Burada ise şart veriliyor, şart doğru ise komutlar1 icra ediliyor ancak şart doğru değilse ise komutlar2 icra ediliyor ve program akışı devam ediyor. Burada **else** şartın gerçekleşmemesi durumunu göstermektedir.

Örnek: eğer voltaj <220 volt ise ekrana voltaj düşük değilse voltaj düzgün yazan programı yazınız.

```
if şart <220 then
    msgbox "voltaj 220 den küçük"
else
    msgbox "voltaj yerinde"
endif
```

3. Şartın gerçekleşmesi ve eğer şart doğru değil ise yeni şartların verilmesi durumu:

```
if şart then
    komutlar1
elseif şart then
    komutlar2
elseif şart then
    komutlar3
.
.
.
else
    komutlar n
endif
```

Şimdi ise şart veriliyor, şart doğru ise doğru şarta ait komutlar icra ediliyor , ancak şart doğru değil ise ve şartın yanlış olması durumunda başka şartlar devreye giriyorsa aşağıdaki gibi format uygulanır.

- Eğer if yapısı içerisinde istenildiğinde çıkılmak isteniyorsa **exit if** deyimini kullanırsınız.

Ödev Soruları:

1. Klavyeden alınan üç sayıdan hangisinin en büyük olduğunu gösteren programı yazınız?
2. İki bilinmeyenli bir denklemde denklemin katsayıları klavyeden verilecek ev buna göre denklemin köklerini bulan programı yazınız.

Select Case Deyimi

If yapısı gibi bir koşul yada değişken için olası değerlerin değerlendirilmesini sağlar. Bu işlemler if deyimi ile yapılacağı gibi select case deyimi ile, bir değişkenin aldığı bir çok değere göre ayrı ayrı komutların yazılması durumunda daha avantajlıdır.

Kullanım Yapısı:

```
Select case şart
    Case durum1:
        Komutlar
    Case durum2,durum3:
        Komutlar
    Case durum4 to durum 7:
        Komutlar
    Case is <durum8:
        komutlar
    Case is >durum9:
        Komutlar
    Case durum n:
        Komutlar
    Case else
        Komutlar

End select
```

Örnekler:

1. ücret hesabı yapılıyor. Buna göre vergi hesaplanacak. Vergi hesabında kişinin ücreti 100-200 arasında ise vergi katsayısı 0,20 , 201-300 arasında ise 0,25, 301-400 arasında ise 0,30 , bunların dışında fazla ise 0,40'dır. Buna göre vergi hesaplayan programı yazınız.

```
Private Sub Form_Load()
    Dim ucret As Integer
    Dim vergi As Single
    ucret = InputBox("ücreti giriniz")
    Select Case ucret
        Case 100 To 200:
            vergi = ucret * 0.2
            MsgBox vergi
        Case 201 To 300:
            vergi = ucret * 0.25
            MsgBox vergi
        Case 301 To 400:
            vergi = ucret * 0.3
            MsgBox vergi
        Case Else
            vergi = ucret * 0.4
            MsgBox vergi
    End Select
End Sub
```

2. Klavyeden girilen herhangi bir değer ne tür olduğunu belirleyen programı yazınız.

```
Dim cevap
cevap = InputBox("herhangi bir değer giriniz")
Select Case cevap
    Case "A" To "Z":
        MsgBox "A ile Z arasında Bir değer Girdiniz"
```



```
Case "a" To "z":  
    MsgBox "a ile z arasında Bir değer Girdiniz"  
Case 0 To 9:  
    MsgBox "0 ile 9 arasında Bir değer Girdiniz"  
Case Else:  
    MsgBox "istenmeyen bir karaktere basıldı"  
End Select
```

- Select yapısından çıkmak için Exit select deyimini kullanırsınız.

Switch Yapısı

Birden fazla şartı aynı satırda kontrol etmek istiyorsanız switch deyimini kullanabilirsiniz.

Sonuc=switch(şart1,deger1,şart2,deger2,şart3,deger3.....şartn,degern)

Ödev Sorusu: Switch deyimi ile klavyeden girilen not değerine göre

Notu 0 40 arasında ise kötü

>=40 ve<60 ise orta

>=60 <85 ise iyi

>=85 ve <100 ise pekiyi yazan program satırlarını yazınız.

DÖNGÜ DEYİMLERİ

Programlama temellerinden biriside döngülerdir. Döngüler işlemlerin yinelenmesi anlamına gelir. Örnek olarak bir veri tabanı dosyasındaki kayıtların birer birer okunarak ekrana listelenmesi gibi. VB iki ana grupta döngü tipi vardır. Bunlar;

- Koşullu Döngüler (Do while – loop, While – End))
- Sayaçlı Döngüler (for –next)

Koşullu döngüler belli bir koşul yerine gelene dek yineleme işlemi yaparlar. Sayaçlı döngüler ise işlemleri belirtilen sayı kadar yenilerler.

For – Next Döngüsü

Bir grup deyimi belli sayıda çalıştırarak (başlangıç değerinden bitiş değerine kadar) döngü oluşturur.

```
For sayac=başlangıç değeri to bitiş değeri [ step artım]
    Komutlar
Next
```

- **Başlangıç değeri bitiş değerinden büyük ise döngüye girilmez.**
- **Eğer döngüyü azalarak çalışmasını istiyorsanız step değerine – değer yazmanız gerekir. Bu durum da ise başlangıç değeri , bitiş değerinden küçükse döngüye girilmez.**
- **Sayaç değişkenine döngü içerisinde genelde yeniden bir atama yapılmaz. Bu durumlarda sonsuz bir döngüye neden olabilirsiniz.**

Örnek:

1. 1 den 20 ye kadar olan sayıların toplamını alan ve her toplamı ekrana yazan programı yazınız.

```
Private Sub Form_Load()
    Dim topla
    topla = 0
    Show
    For i = 1 To 20
        topla = topla + i
        Print topla
    Next
End Sub
```

For next yapısında iç içe for next kullanılabilir. Bu durumda dıştaki döngünün bir sefer dönmesi içteki döngünün ise kendi sayısı kadar dönmesi anlamına gelir. Genellikle sıralama işlemlerinde iç içe döngüler kullanılır.

İç içe döngü:

```
For i= 1 to 10
  For j=1 to 10
    For k=1 to 10
      .....
    next k
  next j
next i
```

örnek: 1 den 5 kadar olan sayıların faktöriyel toplamalarını alıp ekrana yazan programı yazınız.

$5!=5*4*3*2*1$ anlamına gelir.

$4!=4*3*2*1$ dir.

Buna göre iç içe döngü yapılması gerekir. 1 döngü toplama ikinci döngüde ise faktöriyel hesabını almamız gerekir.

```
Private Sub Form_Load()
Dim topla, fakt As Long
topla = 0
fakt = 1
Show
For i = 1 To 5
  fakt = 1
  For j = 1 To i
    fakt = fakt * j
  Next
  Print fakt
  topla = topla + fakt
Next
Print topla
End Sub
```

Örnek: Köpük Sıralama

Verilerin sıralanması işlemidir. Çok sayıda sıralama algoritması vardır. Bunlardan biriside köpük algoritmasıdır. Örneğin klavyeden girilen 10 adet sayıyı büyükten küçüğe doğru sıralayınız. Burada sıralanacak elemanlar bir dizi olarak düzenlenir. Dizinin birinci elemanı ikinci elemanı ile karşılaştırılır eğer küçükse yer değiştirilir. Bu işlem sıralama bitene kadar devam eder.

DİKKAT TAMAMLA

Step Deyiminin Kullanılışı

Step deyimi döngü sayacının belirtilen değerde artması yada azalması için kullanılır.

For j=1 to 10 step 2	yada	for sayi=50 to 2 step -2
Komutlar		komutlar
Next		next

Bu iki döngü while-wend ve do while –loop , verilen koşul doğru olduğu sürece döngü işletilir. Bu tür döngüler for-next döngüsünden farklı olarak döngünün kaç defa çalışacağı önceden bilinmiyorsa kullanılır. Örneğin bir veri tabanı dosyasındaki kayıtların listelenmesi. Kayıtlar sürekli güncellendiğinde kayıt sayısı her an değişebilir. Buna göre bizim şartımız veri tabanı dosyası sonuna kadar olacaktır.

Yapısı: burada şart gerçekleştiği sürece döngü işletilir. Şart döngüye girerken kontrol edilir.

```
While şart
      Komutlar
wend
```

```
do while şart
      komutlar
loop
```

Do Until – Loop döngüsü

Bu döngü ise şart gerçekleşene kadar döngü işletilir. Yani şart gerçekleşmediği sürece döngü işletilir.

```
Do until şart
      Komutlar
Loop
```

Do- Loop while , Do – Loop Until Döngüsü

Bu döngüler do while – loop ile do until –loop döngüleri ile çalışma mantığı aynıdır. Ancak şart bu döngülerde döngüye girerken kontrol edilmez. Döngü sonunda şart kontrole edilir.

```
Do
      Komutlar
Loop until şart
```

```
Do
      Komutlar
Loop while şart
```

Döngü Kontrol Deyimleri

Exit Do: Exit do komutu do – loop, while wend döngülerinden birinde bazı şartlar gerçekleştiğinde döngüyü bitirmek için kullanılır.

Exit for: for next döngüsünde bazı şartlar gerçekleştiğinde döngü tamamlanmadan döngüden çıkmayı sağlar.

Exit Sub-Exit Function: Exit sub ve exit function deyimleri alt programları program sonuna ulaşmadan sonlandırmak için kullanılır.

End: Programı sona erdirir.

With –End With Deyimi

Bir kontrole ait birden fazla özelliği değiştirilmek isteniyorsa her defasında kontrolün adının yazılması gerekir. Bu gibi durumlarda with – end with deyimi kullanılarak kolaylık sağlanış olur.

Örnek: combo kutusuna birden fazla eleman eklenecek ise:

```
With combo1
    .clear
    .additem "kahramanamraş"
    .additem "Gaziantep"
    .additem "İstanbul"
    .additem "Ankara"
    .fontbold=true
    .fontname="courier new"
end with
```

NOT: Eğer Programınız sonsuz döngüye girdi ise ESC yada CTRL – BREAK tuşuna basarak durdurabilirsiniz.

Ödev Soruları:

1. 1 den 10 a kadar olan sayıları do while – loop döngüsünü kullanarak yazınız.
2. herhangi Access (mdb) uzantılı veri tabanı dosyasında kayıtların tamamını ekrana yazdıran programı yazınız.
3. 1 den 10 a kadar olan sayıların kareleri toplamını alan ve her değerin karesini ayrı ayrı list kutusuna yazan programı yazınız.
4. 1 den 10 kadar olan sayıların faktoriyelini alan ve her sayı değerinin faktoriyelini list kutusuna yazan programı yazınız.
5. form genişliğine göre iç içe 5 adet kare çizen programı yazınız.
6. 1 den 10 kadar olan sayıları sıralayınız.

OPERATÖRLER

Operatörler ve Öncelikleri

Aritmetik		Karşılaştırma		Mantıksal	
Üs Alma	^	Eşitlik	=	Not	Değil
Çarpma Bölme	*, /	Eşitsizlik	<>	And	Ve
Tamsayı Bölme	\	Küçüktür	<	Or	Veya
Mod(bölmede Kalan)	Mod	Büyüktür	>	Xor	
Toplama ve Çıkarma	+, -	Küçük yada eşittir	<=	Eqv	
String Birleştirme	&	Büyük yada eşittir	>=	Imp	
		Like			
		İs			

Üstel İşlemler

Örnekler:

$\sqrt[3]{81} = \text{print } 81^{(1/3)}$

$\sqrt{16} = \text{print } \text{sqr}(16)$ yada $\text{print } 16^{(1/2)}$

Mod İşlemi : (Bölmede Kalanı verir)

Örnekler

1 den 100 kadar olan tek sayıların toplamını bulan programı yazınız.

```
Private Sub Form_Load()  
    Dim topla  
    topla = 0  
    For i = 1 To 100  
        If i Mod 2 = 0 Then  
            topla = topla + i  
        End If  
    Next  
    Show  
    Print topla  
  
End Sub
```

Karşılaştırma İşlemleri:

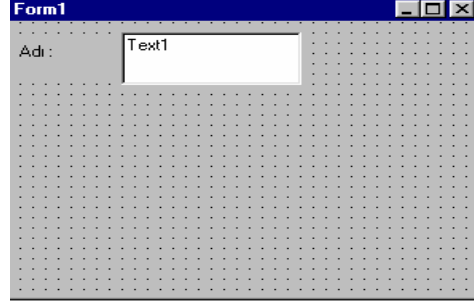
Like Operatörü: stringler üzerinde detaylı karşılaştırma yapmak için kullanılır.

Örnekler:

```
If adres like "*Burhaniye" then  
    Adres değişkeninin sonunda Burhaniye olup olmadığı  
If adres like "Burhaniye*" then  
    Adres değişkeninin başlangıcının Burhaniye olup olmadığı  
If adres like "**Burhaniye*" then  
    Adres değişkeninin herhangi bir yerinde Burhaniye olup olmadığı
```

10.1.Textbox (Metin Kutusu)

VB’de bilgi giriřleri TextBox nesneleri aracılıęı ile yapmaktadır. ToolBox penceresinde TextBox kontrolü iřaretlenir. Kontrol düęmesini seętikten sonra alıřılan form üzerinde mouse ile yeri iřaretlenerek kontrol eklenir.



řekil-10.1. TextBox eklenmiř Form penceresi

10.1.1.Properties

Text

Text kutusuna girilen metin nesnenin text parametresine atanır. Bu kullanılarak kullanıcının girdięi metin üzerinde iřlem yapılır. Text kutusuna text verilmeden sadece ismi kullanılırsa Text özellięi kullanılmıř olur.

```
Text.Text= "Elektronik-Bilgisayar Bölümü"  
Text1= "Elektronik-Bilgisayar Bölümü"
```

ToolTipText

Windows altında alıřan bir ok modern program öğelerinin birinin üzerinde fare ile kısa bir süre durduęunuzda açılan bir balonla onun ne iře yaradıęını yazan mesajlar görmüşsünüzdür. VB’ de ise bu iře o kontrolün **ToolTipText** özellięine atanacak metin ile yapılır. Bu özellięe verdięiniz metin, kullanıcının o kontrol üzerinde kısa bir süre durmasıyla, küçük bir kutu içerisinde gösterilecektir.

MultiLine

True ise text kutusuna birden fazla satır girilebileceęini gösterir. False ise tek satır girilebilir.

Alignment

Nesne içerisindeki yazının sola, saęa veya ortaya yazılmasını saęlar. 0 ise sola, 1 ise saęa, 2 ise yazıyı ortaya yaslar.

ScrollBars

Multiline özellięinin True olması durumunda etkili olan bu özellik Text kutusu içerisinde bu özellięin ařaęı-yukarı, sola-saęa kaydırmak için kaydırma ubuklarının

eklenmesini sağlar. 0 ise yok, 1 ise yatay, 2 ise dikey ve 3 ise hem yatay hem de dikey kaydırma çubukları eklenir.

MaxLenght

Text kutusuna girilebilecek maksimum karakter sayısını belirler. Buna 0 verilirse üst sınır 32 karakter olur.

PaswordChar

Text kutusuna şifre girmek için kullanılır. Yani kullanıcı girdiği karakterlerin ekranda görülmesini istemiyorsa bu özellikten yararlanır. **PasswordChar** özelliğine karakter girerek kullanıcının girdiği bütün karakterlerin bu karakterle görülmesini sağlar.

Locked

Text kutusunun bu özelliği True yapılırsa Text üzerinde hiçbir değişiklik yapılamaz.

Örnek olarak bir şifre programı yazalım.

```
Private Sub Form_Load()  
    If InputBox("Şifreyi Giriniz") <> "66KC" Then  
        MsgBox("Bilemediniz")  
        Text1.Locked=True  
    End If  
End Sub
```

BorderStyle

Nesnenin ekran üzerindeki sınırlarının çerçeve şeklini belirler.

0 : Çerçeve yok

1: Tek çizgili çerçeve

Appearance

Bu özellik kontrolün üç boyutlu görünümünü ayarlar. Bu özelliğin değerine 1 verilirse kontrol üç boyutluymuş gibi, 0 verilirse düz olarak gösterilir. Kontrolün üç boyutlu gösterilmesi daha güzel bir görüntü oluştururken, düz gösterilmesi daha hızlı olmasını sağlar.

SelStart, SelLength

Windows altında çalışan bir çok işlem için bloklama (seçme) yapmak gerektiğini biliyorsunuz. Kullanıcı Text kutularında seçme işlemi yapabilir. Ayrıca bu iki özelliği kullanarak siz de program kodu ile seçme işlemi yaptırabilirsiniz.

Bu özelliklerde SelStart seçmenin Text içinde başlayacağı konumu ve SelLenght ise uzunluğunu belirler.

SelText

Seçilmiş metin bu özellik ile öğrenilip değiştirilebilir. SelText özelliğine yapılan atamalarda, eğer seçilmiş bir kısım varsa o kısım silinir, yoksa kursörün bulunduğu noktaya sıkıştırır.

HideSelection

True ise Text kutusunda seçilen metnin, kontrol başka bir nesneye geçtiğinde seçilen kısmın gizlenmesini sağlar. False ise kontrol başka bir nesneye geçse de seçilen kısmın görülmesini sağlar.

Enabled

Nesnenin aktif veya pasif olmasını sağlar. False yani pasif olması durumunda kullanıcı nesneyi görür ancak üzerinde işlem yapamaz. Text kutusunun Enabled özelliği False ise kullanıcı giriş yapamaz.

Visible

True ise nesne görülür, False yapılırsa görülmez. Belli şartlar gerçekleştiğinde gözükmesini istediğimiz kontrollerin Visible özelliğini kullanabilirsiniz.

FontBold

True ise nesne için kullanılan yazıyı koyu yapar.

FontItalic

True ise nesne için kullanılan yazıyı eğik yapar.

FontStrikethrough

True ise nesne için kullanılan yazının ortasını çizer.

FontUnderline

True ise nesne için kullanılan yazıyı altını çizer.

FontName

Nesne için kullanılan yazının fontunu belirler. Bu font çalışma sırasında sistemde bulunan bir fontun numarasını veya ismi verilerek değiştirilebilir.

FontSize

Nesne için kullanılan yazının puntosunu belirler, maksimum 2048 olabilir.

BackColor

Nesnenin zemin rengini deęiřtirir. Bu özellięi oluřturan deęer 7 dijitalik bir hexadesimal sayıdır.(0 ile &HFFFFFF arasında) En düşük iki dijit kırmızı, sonraki iki dijit yeřil, ve dięer iki dijit mavi rengin yoğunluęunu gösterir. Bu ikili dijitlerin düşük deęerli olmaları rengin koyuluęunu saęlar. En yüksek seviyeli dijit deęeri 0 deęilse Windows' un system renklerinin kullanılacaęını gösterir. Properties penceresinde olmayan renkler, bu dijitlerin ayarlanması ile elde edilebilir. Kullanılabilecek renk sayısı $16^2 * 16^2 * 16^2 = 16.777.216$ renktir. Renkleri ayarlamak için **QBColor()** fonksiyonu da kullanılabilir.

ForeColor

Yazının rengini belirler. Yukarıda anlatılan renk bileřimleri burada da geçerlidir.

Left, Top

Nesnenin sol ve üst noktalarının koordinatlarını belirler. Bu koordinatlar ekran koordinatları olmayıp içinde bulunduęu nesneye baęlı koordinatlardır. Bir nesne Form, PictureBox, Frame içinde bulunabilir. İçinde bulunduęu nesnenin sol üst koordinatları 0,0 olmak üzere Left ve Top bu noktaya olan uzaklıktır.

Height, Width

Nesnenin boyu ve enini belirler. Bu iki özellik deęiřtirilerek kontrolün boyutları ayarlanabilir. VB' de ölçü birimi olarak twip kullanılır. Ancak istenirse formun ScaleMode özellięi deęiřtirilerek ölçü birimi olarak piksel veya metrik birimler verilebilir.

HelpContextId

Kontrolle ilgili yardım dosyasındaki konu numarası bu özellik ile belirlenir.

Index

Aynı isimli birden çok nesne oluřturulmuřsa VB bu nesneleri bir dizi olarak görür. Index parametresi bu nesnenin dizideki kaçıncı eleman olduęunu belirler, bu diziye bu dizi indexiyle ulařılır. **Index** özellięine bir sayı verilmiřse o kontrole isminden sonra bu sayı verilerek ulařılabilir. Örneęin Index özellięi 2 olan bir text kutusunun Text özellięine **Text1(2).Text** satırı ile ulařılabilir. Bir kontrolü dizi olmaktan çıkarmak için ise Index özellięini silmeniz gerekir.

MousePointer. MouseIcon

Mouse göstergesinin nesne üzerine geldięinde alacaęı řekli belirler. 0 ile 15 arası bir sayıdır. 0 normal hali 11 bekleme hali vb. Bu özellięe 99 deęerini vererek kendi tasarladığınız (ICO veya CUR dosyasından) bir kursörü seęebilirsiniz. Bu iřlem için kendi tasarladığınız kursörün bulunduęu dosyanın adını **MouseIcon** özellięine girmeniz gerekir.

Örnek: Aşağıdaki programı çalıştırarak mouse şekillerini görebilirsiniz. Text kutusunu her tıkladığınızda mouse göstericinin şekli değişecek ve numarası Text kutusunda görülecektir.

```
Private Sub Text1_Click ()  
Static i  
Text1.Text = str(i)+ " Numaralı mouse pointer"  
Text1.MousePointer = i  
i=i+1  
If i = 16 Then End  
End Sub
```

TabStop

True ise kullanıcı bu nesneye tab tuşuyla ulaşabilir. **False** ise Tab tuşuyla bu kontrol üzerine gelinmez, mouse ile, yazılım yoluyla veya varsa kısa yol tuşu ile gelebilir.

Yoğun bilgi girişi gerektiren formlarda sık kullanılmayacak kontrollerin TabStop özelliklerini **False** yaparsanız kullanıcının bilgi girişi daha kolay olacaktır. Çünkü genelde yoğun bilgi girişi olan yerlerde Tab tuşu sık kullanılır. Gereksiz yere ekrandaki bütün kontrolleri dolaşmamak için bazılarının bu özelliğini **False** yapmak gerekir.

TabIndex

Form üzerinde kullanıcının ulaşabileceği her nesnenin bir TabIndex' i vardır. TabIndex kullanıcının tab tuşuyla kontroller arasında dolaşırken bu kontrollerin sıralamasını belirler. Örneğin tab indexi 5 olan bir nesnede iken tab tuşuna basılırsa kontrol tab indexi 6 olan nesneye geçer.

Normalde her yerleştirilen kontrole otomatik olarak bir tab sırası verilir. Form tasarımınız bittikten sonra programı çalıştırıp tab tuşuyla bütün kontrolleri gezerek tab tuşunun sırayla gidip gitmediğini kontrol etmeniz gerekir. Çünkü araya yerleştirdiğiniz kontrollerin tab sıraları bozulacaktır.

HWND

Windows altında çalışan kontrollerin **handle** diye adlandırılan tanıtıcı bir numarası vardır. Windows bu numarayı kullanarak kontrolleri tanır. Visual Basic' te bu numara nesnenin hWnd özelliği ile programın çalışması esnasında öğrenilebilir. hWnd özelliği genellikle Windows API çağrılarında gereklidir.

Parent

Nesnenin üzerinde bulunduğu Form' a ulaşmayı sağlar. Nesnenin isminden sonra verilen Parent özelliği o nesneyi değil nesnenin üzerinde bulunduğu formu temsil eder.

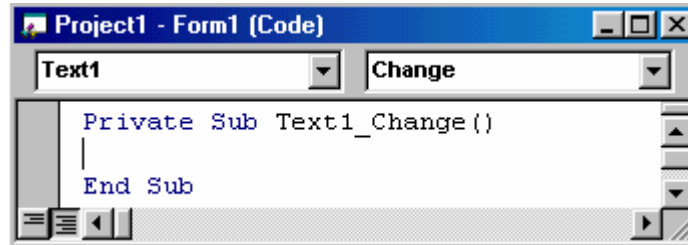
Örneğin **Text1.Parent.Print "Merhaba"** komutu Text1'in bulunduğu Forma "Merhaba" yazacaktır. Böyle bir şeye ihtiyaç duyulmasının sebebi; çoklu Formla çalışma

esnasında yazılacak alt programların bütün formlar için kullanılabilmesinin sağlanmasıdır.


10.1.2.Events (Olay)

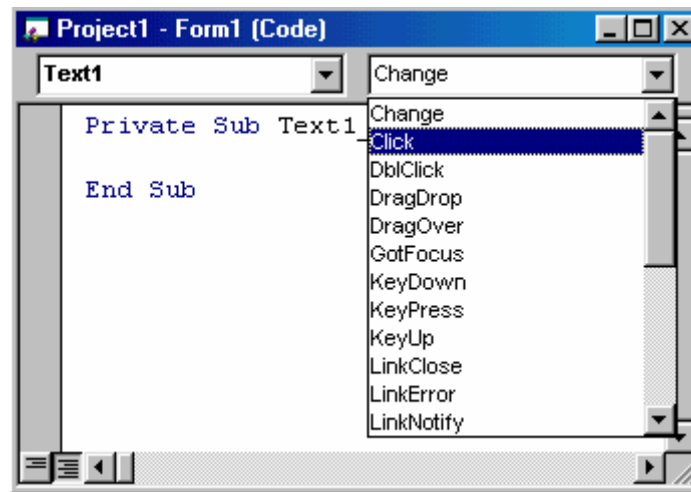
Events yani olay ifadesini şu şeklide açıklayabiliriz. Windows'ta; bir nesneyi tıklamak (Click, DblClick), Mouse'un bir nesne üzerine gelmesi halinde açıklama ifadelerinin görüntülenmesi (MouseMove), iconların taşınması olayı (Drag), Bir dosyanın fare ile tıklatılarak bir yerden başka bir yere alınması olayı (DragDrop), Mouse'un bazı bölgelerde gezinmesi halinde şeklinin değişmesi (DragOver), Mouse'un tuşları basılı tutularak çizim yapılması (MouseDown, MouseUp) vb. gibi yapılan her hareket bir olayı temsil etmektedir.

Form ve Form üzerinde bulunan her kontrol elemanı çift tıklatılarak o elemana ait kod penceresine ve **olay** prosedürlerine ulaşılır. Örneğin tasarım anında Form üzerinde bulunan text kutusunu çift tıklarsak Text kutusunun Change olayına ulaşırız.(Şekil-10.3)



Şekil-10.3 Text1 Change Events Kod Penceresi

Diğer olaylara ulaşmak için pencerenin sağ üst köşesindeki  kutusunu kullanabiliriz. Bu listeyi aşağı doğru açtığımızda o kontrolün olayları listelenecektir.(Şekil-10.4)



Şekil-10.4. Nesne için tanımlı diğer olaylar

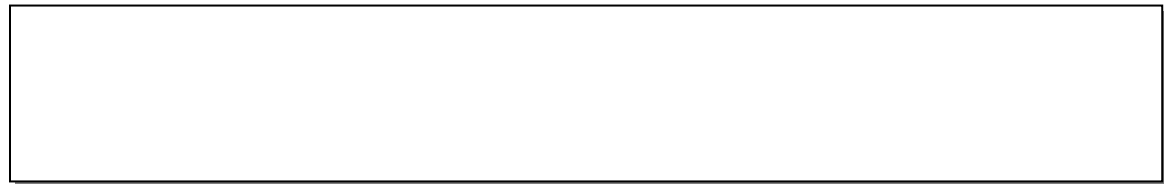
Yukarıdaki listeden istenen olaya ulaşarak (Mouse ile seçerek) ona ait kod

penceresini açıp kod yazabiliriz. Aşağıda her kontrol elemanına **ait olaylar (Events)** **tek tek** ele alınarak incelenmiştir.

Change()

Text kutusunun içerdiği metin değiştiği zaman bu olay meydana gelir. Bu olay kutu içerisinde tuş vuruşlarıyla kullanıcı tarafından meydana getirilebileceği gibi program vasıtasıyla Text kutusuna atama yapılırken de meydana gelir.

ÖRNEK:Örneğin aşağıdaki Formda Text1’de herhangi bir değişiklik olduğunda ekrana “Bu alanı değiştiremezsiniz” diye mesaj veren bir program için Text kutularına ait Change olayı kullanılabilir.



ÖRNEK:Aşağıda görülen form dizaynını kullanarak Text1 kutusu içerisindeki sayı değişir değişmez sayının karesini alın ve sonucu Label1’ e yazan programı hazırlayalım.

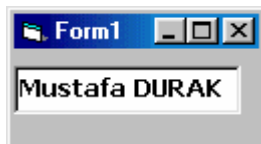


```
Private Sub Text1_Change()  
    Label1 = Val(Text1) * Val(Text1)  
End Sub
```

Click ve DblClick()

Genellikle farenin tıklanmasıyla meydana gelen bir olaydır. Aynı olay bazı nesneler için klavye ile de gerçekleşebilir. Bu durum, farenin yaptığı iş klavyeden de yapılabilirse meydana gelir. DblClick olayı ise nesnenin üzerine mouse ile çift tıklanması durumunda meydana gelir.

ÖRNEK: Click olaylarına örnek olarak bir text kutusu üzerine tıklıldığında yazı renginin, çift tıklıldığında da Zemin renginin değişmesini sağlayacak bir program yazalım.



```
Private Sub Text1_Click( )  
    Text1.ForeColor = QBColor(Rnd * 15)  
End Sub  
Private Sub Text1_DblClick( )  
    Text1.BackColor = QBColor(Rnd * 15)  
End Sub
```

KeyDown (KeyCode As Integer , Shift As Integer)

Bu olay klavyeden bir tuşa basıldığında meydana gelir ve basılı tutulduğu müddetçe devam eder. KeyPress olayından önce meydana gelir ve KeyPress olayının tanımadığı tuşları da tanır. Dışardan iki parametre almaktadır. Birinci parametre hangi tuşa basıldığını belirler(KeyCode), ikinci parametre ise herhangi bir tuşa basılırken Alt, Ctrl veya Shift tuşlarından hangisine basıldığını belirler(Shift). KeyCode ve Shift' e ait genel bir liste aşağıda verilmiştir.

KeyCode : Basılan tuşun Scan kodudur. Klavyedeki tuşların Scan kodları aşağıdaki gibidir.,

Shift: Shift, Alt ve Control tuşlarının durumunu belirtir.

1: vbShiftMask: Shift tuşu maskesi,

2: VbCtrlMask: Kontrol(Ctrl) tuşu Maskesi,

4: VbAltMask: Alt tuşu Maskesi,

Shift parametresi bu maskelerden biri ile AND işlemine tabi tutularak bu tuşlardan hangilerinin basıldığı anlaşılabilir.

KeyCode	Tuş	KeyCode	Tuş
VbKeyBack	Backspace	VbKeyTab	Tab
VbKeyReturn	Enter	VbKeyShift	Shift
VbKeyControl	Ctrl	VbKeyPause	Pause
VbKeyEscape	Esc	VbKeyInsert	Insert
VbKeyDelete	Delete	VbkeyMultiply	*
VbKeyAdd	+	VbKeySubtract	-
VbKeyPageUp	PageUp	VbKeyPageDown	PageDown
VbKeyEnd	End	VbKeyHome	Home

Text1 = " "

If shift AND VbShiftMask Then Text1 . Text= "Shif Tuşu Basılı"

**If shift AND VbCtrlMask Then Text1 = Text1+"Control
Tuşu Basılı"**

Yukarıdaki gibi bir kod ile bu tuşlardan hangisinin veya hangilerinin basılı olduğu anlaşılabilir.

ÖRNEK: Örneğin Enter tuşuna basıldığında Text1 dolu ise imleci Text2 kutusuna getiren program kodunu yazınız.

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case vbKeyReturn
If Text1.Text <> "" Then Text2.SetFocus
Else Text1.SetFocus
End If
End Select
End Sub
```

KeyUp (KeyCode As Integer, Shift As Integer)

Bu olay basılan tuş bırakıldığı anda meydana gelir. Her klavye eventin' de oluşmaz. Şöyle ki A harfine uzun bir süre basarak birden çok A girişi yapılır. Bu giriş esnasında Keyup olayı sadece son A harfinin yazılmasından sonra meydana gelir. Mantığı Keydown olayı ise aynıdır.

KeyPress (KeyAscii As Integer)

Bu olayı klavyedeki tuşların bir kısmı meydana getirir. Bu olay basılan tuşların ASCII kodları varsa meydana gelir. KeyDown olayından sonra, KeyUp olayından ise önce meydana gelir. Klavye eventi sırasıyla şu olayları meydana getirir. **KeyDown, KeyPress, Change, Keyup**

ORNEK:KeyPress olayını kontrol ederek küçük harfleri büyük harfe çeviren ve rakamları kabul etmeyen bir örnek aşağıda verilmiştir.

```
Private Sub Text1_KeyPress (KeyAscii As Integer)
  If (KeyAscii > Asc("0")) AND (KeyAscii <= Asc("9")) Then
    KeyAscii = 0 'Rakam ise iptal et
  Else
    KeyAscii = Asc (Ucase (Chr(KeyAscii))) 'büyük harfe çevir
  End If
End Sub
```

KeyPress, KeyDown, KeyUp Olaylarının tanıdıkları tuş grupları

Tuş Grubu	KeyPress	KeyDown	KeyUp
Num, Caps, Scroll	Hayır	Evet	Evet
Control, Alt, Shift	Hayır	Evet	Evet
Alt + Bir harf	Hayır	Evet	Evet
Ctrl + Bir harf	Evet	Evet	Evet
Kısa yol Tuşları	Hayır	Evet	Hayır

MouseDown, MouseUp (Button As Integer , Shift As Integer , X As Single , Y As Single)

Bu olaylar; bir kontrolün üzerinde iken mouse düğmelerinden birinin basılıp bırakılması esnasında meydana gelir. Burada MouseDown farenin basılmasına, Mouseup ise farenin basılmış tuşunun bırakılmasına karşılık gelir.

Button

Button parametresi farenin basılı tuşlarını ifade eder.

- 1: VbLeftButton:Farenin sol tuşunu,
- 2: VbRightButton:Farenin sağ tuşunu,
- 4: VbMiddleButton:Farenin orta tuşunu ifade eder.

Shift

Basılan control, alt ve shift tuşların durumlarını gösterir. Bu parametre Keydown

olayında açıklanmıştır.

X,Y:

Fare göstergesinin bulunduğu konumun koordinatlarını verir.

ÖRNEK: Aşağıdaki örnekte kullanıcı Text kutusunda farenin sol tuşuna bastığında Text kutusunun yazı rengini, sağ tuşuna bastığında zemin rengini değiştiren program kodlarını MouseDown ve MouseUp olaylarını kullanarak tasarlayalım.

```
Private Sub Text1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Select Case Button
Case vbLeftButton 'Sol tuş basılı ise
Text1.BackColor = QBColor(Rnd * 15)
Case vbRightButton 'Sağ tuş basılı ise
Text1.ForeColor = QBColor(Rnd * 15)
End Select
End Sub
```

MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)

Mouse'un bir form yada bir kontrol elemanı üzerinde dolaşması sonucu MouseMove olayı meydana gelir. Dolaşma olayı devam ettiği müddetçe bu olay meydana gelir.

ÖRNEK: İki tane Text kutusu içeren bir Formda, Mouse hangi Text kutusu üzerinde ise o Text kutusunun zemin rengini sarı yapan ve Text kutusu üzerinden gittiğinde de rengini eski haline getiren bir programı MouseMove olayı ile gerçekleştirelim.

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Text1.BackColor = &H80000005
Text2.BackColor = &H80000005
End Sub

Private Sub Text1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Text1.BackColor = &HFFFF& 'QBColor(Rnd * 15)
End Sub

Private Sub Text2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Text2.BackColor = &HFFFF&
End Sub
```

GotFocus()

Klavye kontrolünün bu nesneye geçmesi halinde meydana gelen bir olaydır. Bir nesneye kontrolün geçebilmesi için görülebilir ve aktif olması gerekir.

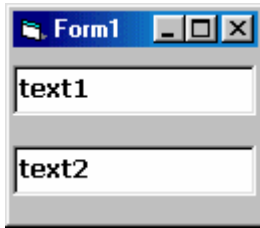
ÖRNEK: Örneğin imleç (Kursör) Text kutusuna geldiğinde Text kutusu içerisindeki metnin seçilmesini sağlayan program kodunu GotFocus olayına kullanarak gerçekleştirelim.

```
Private Sub Text1_GotFocus()  
    Text1.SelStart = 0 'İmleç' i Text' in ilk harfine getirir.  
    Text1.SelLength = Len(Text1)  
End Sub
```

LostFocus()

Bu olay herhangi bir kontrol elemanının kontrolü kaybetmesi esnasında meydana gelir. Bir eleman, kontrolü ya Tab tuşuyla yada fare tarafından tıklanmasıyla alır. Tab tuşu elemanlar arasında geçişi sağladığından dolayı elemanlara kontrolü verme veya geri alma olaylarında rol oynar. LostFocus olayının gerçekleşmesi için form üzerinde birden fazla kontrol elemanı yer almalıdır. Form üzerindeki menülerin seçilmesi LostFocus olayını meydana getirmez. Bu olay daha çok kontrolü kaybeden elemanın içeriğini değerlendirmek için kullanılabilir.

ÖRNEK: Örneğin kullanıcının girdiği sayı istediğimiz bir aralıkta değilse elemanın kontrolü kaybetmesini önleyelim. Bu örneğin çalışmasını görmek için Form üzerine birden fazla eleman koymanız gerekir. Öncelikle aşağıda görülen Formu tasarlayalım.

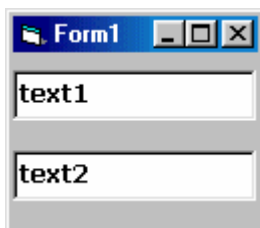


```
Private Sub Text1_LostFocus()  
    If Val(Text1) < 1 Or Val(Text1) > 10 Then  
        Text1.SetFocus  
    End If  
End Sub
```

Burada kontrol Text kutusuna geçtikten sonra 1 ile 10 arası bir sayı girilmediği sürece kontrol Form üzerinde (varsa menüler hariç) başka bir elemana geçmez. SetFocus bir metod olup imleci ilgili elemanın üzerine getirir.

Validate (Cancel as Integer)

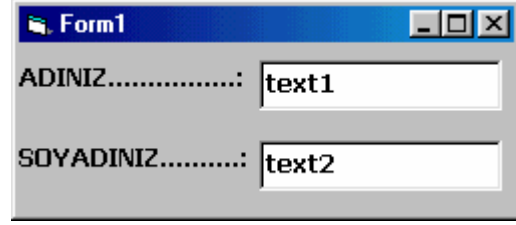
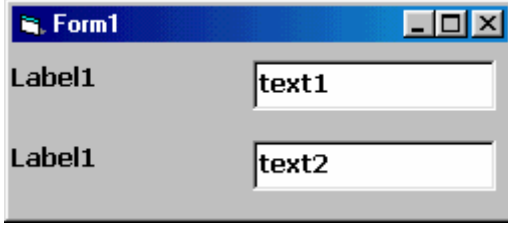
6.0 versiyonu ile eklenen bu özellik LostFocus olayında anlattığımız kontrol olayını yapmaya yarar. Eğer nesnenin CausesValidation özelliği True ise bu olaya yazdığımız kod çalışacak ve nesnenin içeriğini kontrol edebilmemizi sağlayacaktır. Eğer nesnenin içeriği uygun değilse Cancel parametresine True atanarak kontrolün tekrar aynı nesneye bırakılmasını sağlar.



```
Private Sub Text1_Validate(Cancel As Boolean)  
    If Val(Text1) <> 2000 Then 'Text1 2000' den farklı ise  
        Cancel = True          'Text2' ye geçemez.  
    Else  
        Cancel = False  
    End If  
End Sub
```

10.2.Label Kontrol Elemanı

Daha çok kullanıcıya Form üzerinde bilgi vermek için kullanılır. Bu nesneye kullanıcı tarafından giriş yapılamaz. Label kontrol kullanıcının giriş yapamamasına rağmen Text kutularına çok benzer.



10.2.1.Properties

Caption

Nesnenin üzerindeki yazı bu özellik ile belirlenir. Tasarım anında properties penceresi aracılığı ile Label üzerindeki yazı belirlenebileceği gibi çalışma anında da aşağıda verilen komut satırı (kod satırı) ile de Caption özelliği değiştirilebilir.

```
Label1.Caption = "ADINIZ.....:"  
Label2.Caption = "SOYADINIZ.....:"
```

Caption özelliğinde & karakterinin kullanılması bu karakterden sonra gelen karakterin nesneye ait kısa yol olarak atanacağını gösterir.

```
Label1.Caption = "&ADINIZ.....:"  
Label2.Caption = "&SOYADINIZ.....:"
```

Yukarıdaki satır A harfini kısa yol tuşu yapar. Alt ve A tuşlarına birlikte basılması durumunda klavye kontrolü TabIndex'i(imleci)kendisinden bir sonra gelen elemana getirir.

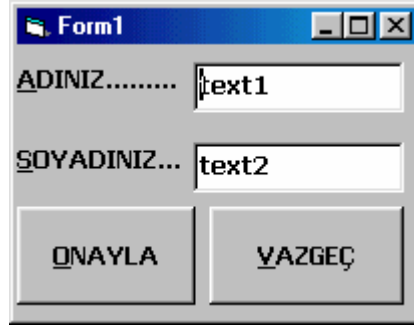
Bu kontrol çoğunlukla diğer kontrollere açıklama yazmak için kullanılır. Örneğin değişik bilgilerin girileceği Text kutularının veya diğer kontrollerin önüne Label elemanları konularak bunlara neyin girileceği belirlenir.

AutoSize

Bu özellik True ise nesnenin boyutları içeriğinin boyutlarına göre yeniden ayarlanacaktır. İçerik değiştiği an Label' in de boyutları otomatik olarak içerik uzunluğuna göre değişecektir.

10.3.Command Button (Komut Düğmesi)

Bir olayın kullanıcı tarafından başlatılması için programlarda çok kullanılan kontrollerden biridir. Kullanıcı istediğinde çalıştığı gibi program içerisinde de bir alt program gibi çalıştırılabilir.



10.3.1.Properties

Caption

Komut düğmesi üzerindeki yazılacak olan mesajı içerir. Diğer nesnelerin Caption özelliklerinde de olduğu gibi altı çizgili harf komut düğmesinin kısa yolunu belirtmektedir. Kısa yol tuşunu belirlemek için Caption' da kısa yol olacak harfin önüne & karakteri yazılır. Örneğin aşağıdaki gibi bir kod ile “ONAYLA” ve “VAZGEÇ” komut düğmelerinin kısa yol tuşları sırası ile Alt+O ve Alt+V yapılabilir. Tasarım anında properties penceresi aracılığı ile Command Button üzerindeki yazı belirlenebileceği gibi çalışma anında da belirlenebilir.

```
Command1.Caption = "&ONAYLA"  
Command2.Caption = "&VAZGEÇ"
```

Style

Bu özellik komut düğmesinin yazılı veya resimli olmasını sağlar. Değeri 0 ise komut düğmesinin üzerinde Caption özelliği ile belirlenen yazı bulunur. Değeri 1 ise komut düğmesinin üzerinde Picture özelliği ile belirlenen resim bulunur.



Style özelliğinin değeri 0 olan bir komut düğmesi.



Style özelliğinin değeri 1 olan bir komut düğmesi.

Picture, DisabledPicture, DownPicture

Eğer komut düğmesinin Style özelliği 1 verilerek resimli bir komut düğmesi olacağı belirtilmiş ise bu üç özellik ile komut düğmesinin üzerinde bulundurulacak resim belirlenir.

Picture özelliği ile belirlenen resim komut düğmesi aktif iken,

DisabledPicture özelliği ile belirlenen resim komut düğmesi pasif iken,

DownPicture ile de komut düğmesine basılı iken gösterilecek resmi belirlememizi sağlar.

Default

Bir komut düğmesinin Default özelliği True ise o düğmenin bulunduğu Form üzerinde Entere basılmasını durumunda o düğmeye ait program kodları çalıştırılmış olur.

Cancel

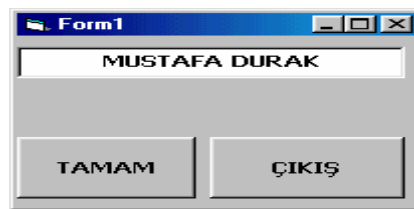
Bu özellik Default özelliği gibidir ancak **Esc** tuşu ile aktif hale gelir. Cancel özelliği True olan bir komut düğmesinin bulunduğu Form üzerinde Esc tuşuna basılırsa o komut düğmesine ait program kodları icra edilir.

10.3.2.Events

Click ()

Komut düğmesinin en önemli olayı budur. Düğme tıklatıldığında yapılması istenen işlem bu olaya yazılır.

ÖRNEK: Çıkış ve Mesaj Caption 'lı iki komut düğmesi ve 1 adet de Text kutusu içeren bir Formda Çıkış çalıştırıldığında "Güle Güle" Tamam çalıştırıldığında ise Text kutusu içerisindeki ifadeyi mesaj olarak gösteren programı tasarlayalım. Programımız için gerekli Form aşağıdaki gibi olacaktır.



```
Private Sub Command1_Click( )  
    MsgBox Text1.Text, vbInformation  
End Sub  
  
Private Sub Command2_Click( )  
    MsgBox " Güle güle ", vbInformation  
End Sub
```

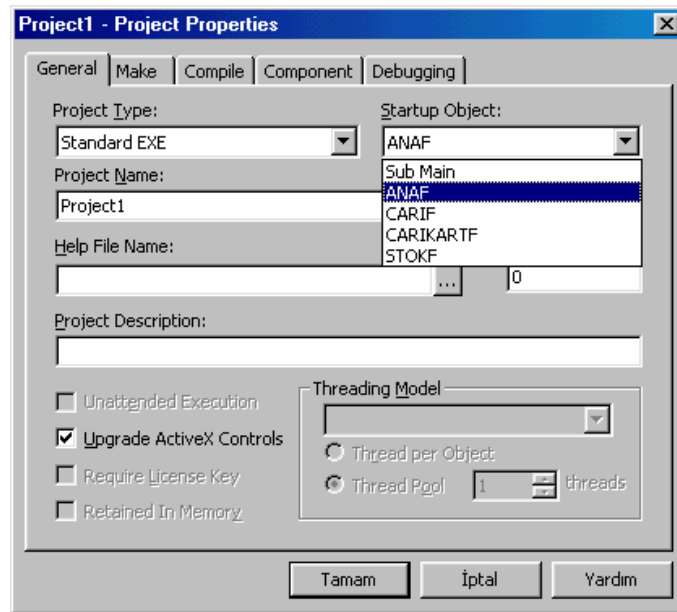
Windows arabiriminin en temel kontrolü formlardır. Windows'ta hemen her program Formlar üzerinde çalışır. Zaten Windows kelimesinin Türkçe anlamı Pencereleler de bu formlardır. Boyutlandırılabilir özelliği sayesinde aynı ekranda tek bir program olmak zorunda değildir.

Formun Properties kısmını Formun alt programlarında yazarken Formun ismini kullanmak zorunda değilsiniz, direk properties ismini vermeniz yeterlidir. Yani **Form1.Caption** ile sadece Caption, Form1' in alt programlarında aynı etkiye sahiptir.

Programınızda birden fazla Form bulunacaksa ilk olarak ana Form çalışacaktır. Diğer Formları programınızda kullanacağınız **Form2.Show** gibi bir yöntemle aktif hale getirmelisiniz.

Bir formdan başka bir formun bir özelliğine ulaşabilmek için kontrol adından önce formun adı da verilmelidir. Örneğin Form2 üzerindeki Label1'e ulaşmak için **Form2.Label1.Özellik** şeklinde kullanılır.

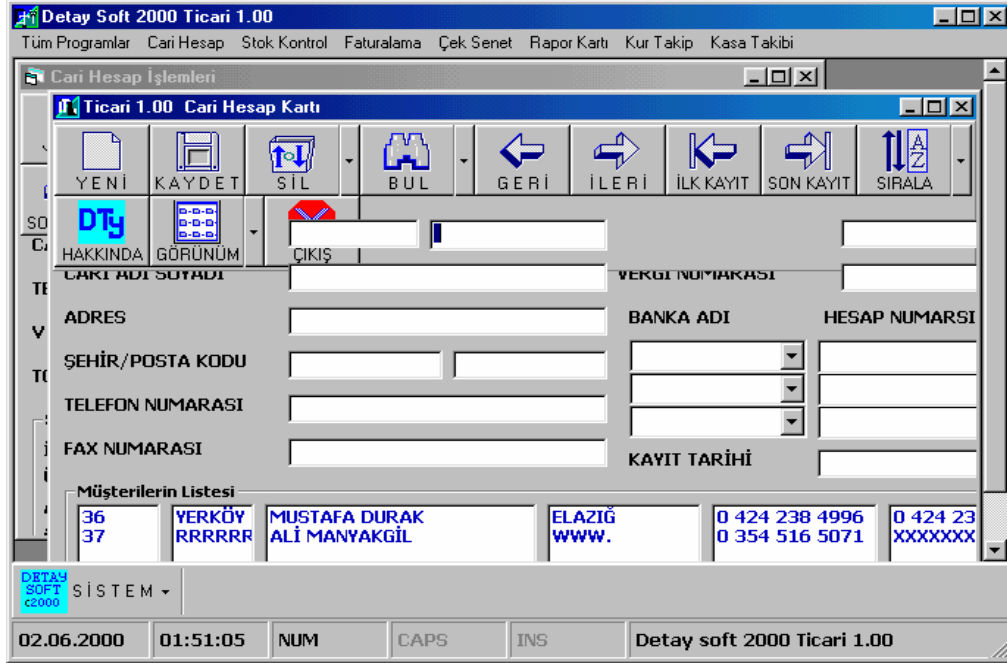
Programınızda birden fazla Formunuz varsa ilk oluşturduğunuz Form ana Formdur ve program çalışmaya o Form ile başlar. Eğer başlangıçta çalışacak Formu değiştirmek istiyorsanız; Project-Properties menüleri ile açılan aşağıdaki pencerenin Startup Object listesinden istediğiniz Formu seçebilirsiniz.



Yukarıdaki pencerede görüldüğü gibi program çalışmaya Form' dan başlayabileceği gibi ismi Main olan bir alt programdan da başlayabilir. Bunun için bir Modül oluşturulur ve Sub Main ile başlayan bir alt program yazılır ve program buradan başlatılabilir.

Formlar ikiye ayrılır. Biri programlarda kullandığımız tek başına çalışan SDI (Single Document Interface) formlarıdır. Şu ana kadar örneklerimizi üzerinde yaptığımız formların tamamı SDI form tipindeki formlardır.

İkincisi de MDI (Multi Document Interface) formlarıdır. MDI formlar içinde kendisine bağlı formlar bulundurulur. Bu formlar (Child Form) MDI forma bağlıdır. MDI formla birlikte hareket ederler. Aşağıda bir MDI form ve onun içinde bulunan Child formlar görülmektedir.



SDI Formlar

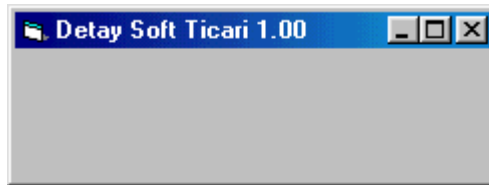
Tek başına çalışan formlara SDI Form denir. Normalde yeni bir projeye başlarken projede bulunan form daha sonra eklenen formlar SDI formlardır. Bu formlar herhangi bir forma bağlı olmaksızın kendi başlarına çalışırlar.

10.12.1.Properties

Caption

Formun başlığına yazılacak yazıyı belirler. Aşağıdaki satır programın Form_Load() yordamına yazılırsa aşağıdaki görüntüyü elde ederiz.

```
Form1.Caption= "Detay Soft Ticari 1.00"
```



Icon

Formu tanıttak iconu (ikon) belirler. Seçilecek ikon Form minimize edildiğinde, Alt+Tab geçişlerinde bu formu temsil eder. Ayrıca programa ait bir kısa yol oluşturulduğunda bu ikon kullanılır. Properties penceresinden bu özellik çift tıklanarak açılan pencereden GRAPHICS\ICONS dizinine geçilerek çok sayıda ikon bulunabilir.

BorderStyle

Formun çerçeve şeklini belirleyen bu özellik formun boyutlarının değiştirilmesini veya kapatılabilmesini vereceğiniz değerler engelleyebilirsiniz.

0 : vbBSNone

Çerçevesi, Başlığı, Kontrol kutusu, Ekranı Kapla (Maximize), Simgeler durumuna Küçült (Minimize) düğmeleri olmayan bir form oluşturur. Bu değer verildiğinde kullanıcı formu boyutlandıramaz, taşıyamaz ve kapatamaz.

1 : VbFixedSingle

Boyutları değiştirilmeyen, fakat konumu kullanıcı tarafından değiştirilebilen yani taşınabilen bir form oluşturur.

2 : VbSizable

Default değer budur ve formun tüm özelliklerini sunabilir. Formun kapatılması, boyutlandırılabilmesi ve taşınması mümkündür.

3 : VbFixedDouble

Form kullanıcı tarafından boyutlandırılmaz ama kullanıcı Formu kapatabilir veya taşıyabilir. 1 değerinden farkı; yalnızca maximize ve minimize düğmelerinin olmamasıdır.

4 : VbFixedToolWindow

Normal Forma göre başlığı daha küçük olan, kontrol menüsü olmayan ve boyutlandırılmayan bir form oluşturur. Bu form daha çok ToolBox pencereleri gibi araçların bulunduğu formlar için uygundur.

5 : VbSizableToolWindow

4 değerindeki gibi bir Form oluşturur. Farklı olarak bu pencerelerin kullanıcı tarafından değiştirilebilir.

MaxButton, MinButton

Formun sağ üst köşesindeki maximize ve minimize düğmelerinin görüntülenmesini veya görüntülenmemesini sağlar. Bu değerler True ise formun maximize veya minimize edilmesi herhangi bir koda gerek kalmadan gerçekleşir.

ControlBox: Formun sol üst köşesindeki kontrol kutusunun görüntülenmesini veya görüntülenmemesini sağlar.

WhatsThisButton: Form üzerinde ? düğmesinin bulunmasını sağlar. Form üzerinde bu düğmenin bulunabilmesi için;

1. ControlBox özelliği True
2. BorderStyle özelliği Fixed Single, Sizable veya Fixed Dialog
3. MinButton ve MaxButton özellikleri False

Moveable

Bu özellik formun kullanıcı tarafından taşınıp taşınamayacağını belirler.

ShowInTaskbar

Bu özellik formun Windows görev çubuğunda yer alıp almayacağını belirler.

AutoRedraw

Bu özellik False ise form üzerinde başka bir form geldiğinde veya form boyutlandırıldığında form üzerine yapılan yazım ve çizimler yenilenmeyecektir. True ise form boyutlandırılırken veya üzerine kapatılırken formun içeriği kaydedilerek formun içeriğinin kaybolmasını önleyecektir.

FillColor, FillStyle

Circle ve Line metodu ile form üzerine çizilen çember ve kutuların iç boyama rengini ve desenini belirler.

FontTransparent

True ise form üzerine Print komutu ile yazılacak yazıların zemin rengi yoktur. Dolayısıyla yazı altında resim veya başka bir yazı varsa bunu gösterecektir.

WindowState

Formun çalışması üç şekilde olur. Bu durumlar WindowState özelliği ile belirlenir.

0 : vbNormal: Normal, ekranın herhangi bir kısmında pencere içinde.

1 : vbMinimized: Simge durumunda.

2 : vbMaximized: Ekranın tamamını kaplayacak şekilde.

StartupPosition: Form yüklenirken ekrandaki koordinatlarının neye göre belirleneceğini bu özellik etkiler.

0: Form tasarlanırken bulunduğu koordinatlarda açılır.

- 1: Form içinde bulunduğu formun ortasında açılır.(MDI Child formlar için)
- 2: Form ekranın ortasında açılır.
- 3: Formun koordinatları Windows tarafından belirlenir.

KeyPrew

Form aktifken basılan tuşlardan formun haberdar edilip edilmeyeceğini belirler.

False : Herhangi bir kontrol üzerinde iken basılan tuşlar yalnız o kontrolün KeyPress, KeyDown, KeyUp olaylarını meydana getirir.

True: Her hangi bir kontrol üzerinde basılan tuşlar önce Form'un daha sonra o kontrolün KeyPress, KeyDown, KeyUp olaylarını da meydana getirir.

Örneğin ESC tuşuna basıldığında programın kapanmasını istiyorsak bu özelliği True yapıp Formun Keypress olayına aşağıdaki komut satırını yazmamız gerekir.

```
if KeyAscii=27 Then End
```

Count: Formun içindeki menüler dahil kontrol sayısını verir.

ActiveControl: Form üzerinde aktif kontrolün ismi gibi davranır.

Örnek: Form1.ActiveControl.Left=0 satırı form üzerindeki aktif kontrolü sağlar.

Burada aktif kontrol TextBox ise bu satır;

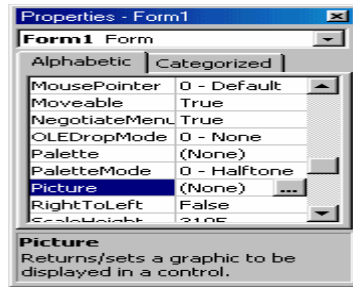
```
Form1.TextBox.Left=0
```

ListBox ise;

```
Form1.ListBox.Left=0
```

Picture

Form üzerinde gösterilecek resmi belirler. Formun properties penceresinde Picture seçeneği işaretlendiğinde, bu seçeneğin karşısında üç nokta çıkar.



Bu üç nokta tıklandığında Load Picture penceresi açılır.

Bu pencerede istenilen resim dosyası işaretlenerek “Aç” düğmesi mouse ile tıklanarak dosya forma yerleştirilir.

Image

Formun AutoRedraw özelliği True ise form üzerine yapılan yazım ve çizimlerin kaybolmadığından bahsetmiştik. Formun AutoRedraw özelliği True ise form üzerine yapılan çizim ve yazımlar Formun Image özelliğine kaydedilebilir. Bu özellik yalnız okunabilir bir özelliktir. Eğer form üzeri kaydetmek, panoya kopyalamak veya başka bir kontrol için göstermek istiyorsak bu özellik kullanılır.

MDIChild

Windows'un formlara verilebildiği diğer bir özellik de form içinde form oluşturulabilmesidir. Programımızda bir MDI form varsa diğer formların MDIChild özelliğini True yaparak Child formlar oluşturulur.

Hdc

Windows altında oluşturulan kontrollerin birer handle numarası vardır. Windows kontrollere bu numarayla ulaşır. Visula Basic'te bu numaralar hWnd özelliği ile öğrenilebilir. hDC numarası da bir handle numarasıdır. Fakat özel bir numaradır. Bu numara Windows uygulamaları ile Device Driver (birim sürücüleri) arasında bir bağlantı kurar. bU numara hWnd özelliği gibi API'lerle kullanılır.

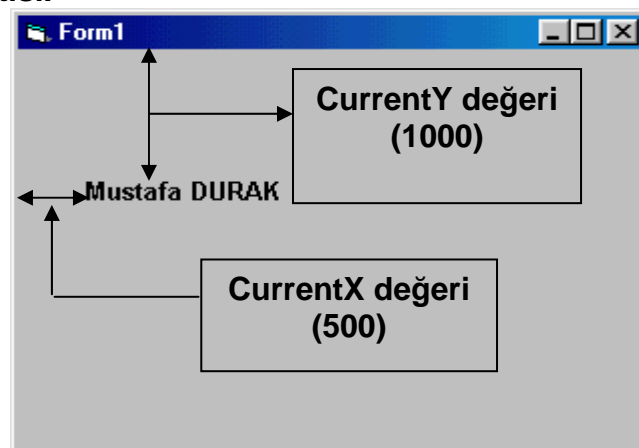
CurrentX, CurrentY

Form üzerine yazılan başlangıç noktası olmayan çizim ve yazımlar aktif pixelin bulunduğu bölgeden başlar. Bu aktif pixelin x ve y koordinatlarını CurrentX ve CurrentY özellikleriyle belirlenir. Özellikle form üzerine yazdığımız Print modu ile yazılarda bu özellik yazının koordinatlarını belirler.

Örnek:

```
Private Sub Form_Load()  
    Show  
    FontName = "Arial"  
    FontSize = 10  
    FontBold = True  
    CurrentX = 500  
    CurrentY = 1000  
    Print "Mustafa DURAK"  
End Sub
```

Programın çıktısı.



10.12.2. Events

Load()

Form ilk defa belleğe yüklenirken bu olay meydana gelir. Formun Visible özelliği ile gizlenip tekrar gösterilmesi bu olayı meydana getirmez. Çünkü form bellektedir. Programımızda birden çok form varsa program çalışmaya başladığında sadece ana form belleğe yüklenir. Diğer formları yazacağımız kodlarla bizim yüklememiz gerekmektedir. Formlar aşağıdaki şekilde belleğe yüklenir ve Load olayları meydana gelir.

- Load Formismi ile form direk olarak belleğe yüklenir. Ancak kontrolü ele almaz.
- Formadı.Show ile form belleğe yüklenir ve görüntülenir.
- O anda bellekte olmayan formla ilgili bir koda rastlandığında o form belleğe yüklenir ve daha sonra işlem görür.

Örnek:

Form2.Text1.Text= "12"

Yukarıdaki satırla Form2'nin yüklenmesini, dolayısıyla Form2'nin Load olayının çalışmasını ve Text1 kontrolüne "12" atamasının yapılması sağlanacaktır.

Activate(), Deactivate ()

Programımızda birden fazla form varsa, aynı anda sadece biri aktiftir. Aktivitenin programdaki formlardan diğerine geçmesi durumunda aktiviteyi kaybeden formun, Deactivate olayı, aktif olan formun da Activate olayı meydana gelir.

Unload (Cancel As Integer)

Form herhangi bir şekilde kapanırken bu olay meydana gelir. Bu olaya yazacağımız kodla form kapanmadan önce yapılması gereken işleri yapabiliriz. Ayrıca Cancel=True yazarsak formun kapatılmasını önleyebiliriz. Formun Unload olayı şu hallerde meydana gelir.

- Formun sol üst köşesindeki kontrol kutusunda "kapat" seçildiğinde,
- Windows görev yöneticisinden "Göreve son ver" düğmesi seçilmesiyle,
- Windows'tan çıkılmaya başlandığında,
- Form bir MDIChild ve MDI formun kapatılmasıyla.

- Programın herhangi bir yerinde End komutu ile sonlandırılması durumunda Unload olayı gerçekleşmez. Bu yüzden Unload olayına kod yazdığınız programlarınızda End komutu ile Programı sona erdirmemeliyiz.

QueryUnload (Cancel As Integer, UnloadMode As Integer)

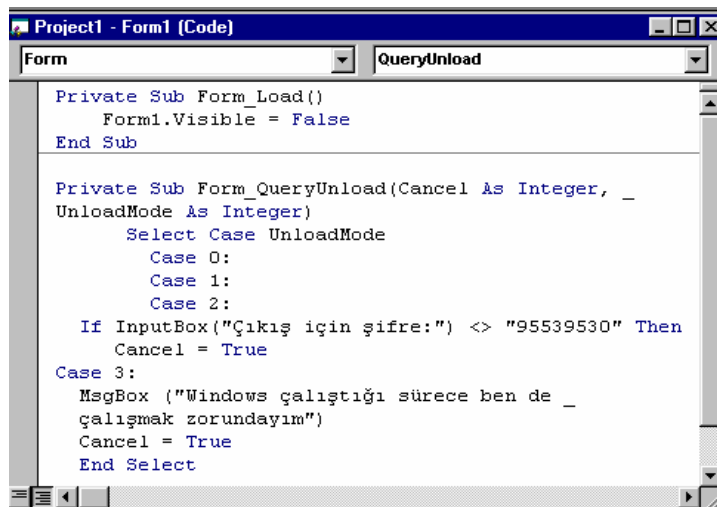
Unload olayıyla aynı işi yapar ancak formun kimin tarafından kapatılmaya çalışıldığı da öğrenebilirsiniz. Ayrıca bu olay Unload olayından önce meydana gelir ve burada Cancel = True ile kapatma olayı iptal edilirse Unload olayı meydana gelmez.

Unload olayında bir formun 5 değişik şekilde kapanacağını belirtmiştik. QueryUnload olayındaki UnloadMode parametresiyle formun hangi şekilde kapatılmaya çalışıldığını öğrenebiliriz. Bu parametrenin alacağı değerler şöyledir.

- 0: vbFornControlMenu** (Kontrol kutusunda “kapat” seçildi)
- 1: vbFormCode** (Programda Unload komutu kullanıldı)
- 2: vbAppWindows** (Windows’tan çıkılmaya çalışıldı)
- 3: vbATaskManager** (Task Managerden “Göreve Son Ver”)
- 4: vbFormMDIForm** (MDI form kapatıldığında)

Örnek olarak doğru şifre girilmediği sürece Windows’tan çıkışı önleyecek bir program yapalım.

Windows’u kapatmak istediğinizde, Windows açık olan bütün uygulamalara kendini kapatmalarını bildirir. Yani programların QueryUnload ve Unload olayları aktif hale getirilir. O halde biz Windows’tan çıkılacağını anlayabilmek için yazacağımız programın QueryUnload olayına gerekli kodu yazmamız gerekir. Windows kapatılmak istendiğinde programımız o anda bellekte ise QueryUnload olayı aktif hale gelecek ve doğru şifre girilmezse Windows kapatılmayacaktır. Programı sürekli bellekte tutmak içinde “Başlangıçta” grubuna koymamız gerekir. Ayrıca programın ekranda görülmemesi içinde formu gizlemeliyiz.



Resize()

Formun boyutlarının değişmesi halinde bu olay meydana gelir. Formun genişliğinin veya yüksekliğinin değişmesi ayrıca formun minimize edilmesi bu olayı meydana getirir.

10.12.3. Methods

Line: Bu komutla çizgi veya dikdörtgen çizilebilir.

Line (X1,Y1)-(X2,Y2),[renk]

x 1,y1 koordinatlarında **x2,y2** koordinatlarına verilen renk ile bir çizgi çizer.

Line (x2,y2),[renk] şeklinde kullanılarak en son kalınan noktadan x2,y2 noktasına çizim yapılabilir.

Line (x1,y1)-(x2,y2),[renk],B

şeklinde kullanılırsa x1,y1 köşesinden x2,y2 köşesine bir dikdörtgen çizer.

Line (x1,y1)-(x2,y2),[renk],BF

F parametresi ile kullanıldığında ise dikdörtgenin içi renk parametresi ile belirlenen renkle boyanır. **F** parametresi olmaksızın kullanıldığında ise dikdörtgenin için FillColor ve FillStyle özellikleri ile belirlenen modda boyanır.

Circle: Çember, Elips ve yay çizmek için kullanılır.

Circle (mx,my),r,renk

mx,my merkezli r yarıçaplı çemberi verilen renkte çizer.

Circle (mx,my),r,renk,a,b

mx, my merkezli r yarıçaplı yayı a açısından b açısına verilen renkte çizer.

Pset: Verilen koordinatlar arasına nokta koyarak şekil çizer.

Pset (x,y),renk

Point: Verilen koordinatların rengini belirler.

Renk=Point (x,y)

Print: Form üzerine kontrolden bağımsız olarak yazı yazmak için kullanılır.

Yazının;

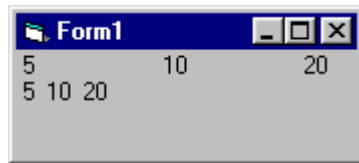
- Koordinatları CurrentX, CurrentY özellikleri ile
- Biçimi FontName, FontSize, FontBold, FontItalic, FontUnderLine, FontStrikeThru özellikleriyle
- Boyu FontSize özelliği ile

- Rengi ForeColor özelliği ile
- Zemin renginin olup olmayacağı FontTransparent özelliği ile belirlenebilir.

Print metodunda birden fazla değişken araya noktalı virgül koyularak yan yana, virgül koyularak bir tab mesafesinde aralıklı olarak yazdırılabilir.

Örnek :

```
Dim x,y,z  
Show  
x=5 :y=10: z=20  
Print x,y,z  
Print x;y;z
```



Clis: Form üzerine yukarıdaki yöntemlerle yazılmış yazı ve çizimleri siler. Forn üzerindeki kontrollerde bir değişiklik olmaz.

TextHeight (Metin), TextWidth(Metin)

Bu özellikler, Metin parametresiyle verilen yazının formu, picturebox veya yazıcıya print metoduyla yazılması durumunda yazının yüksekliğini ve genişliğini nesnenin ScaleMode özelliğiyle belirlenen birim cinsinden verir. Bu değerlere nesnenin FontName, Fontsize, FontItalic, FontBold gibi fon özellikleri etki eder.

Bu özellikleri, özellikle yazıcıda tablo gibi aynı hizada bulunması gereken çıktılarda kullanmak gerekli olabilir.

Move sol, üst, genişlik, yükseklik: Formun veya herhangi bir kontrolün konumunu ve boyutlarını tek sefere değiştirir.

Scale (x1,y1,x2,y2)

Bu özellikte kontrolün kullanacağı koordinat sistemi yeniden tanımlanabilir. Örneğin matematiksel grafikler için bilgisayarda kullanılan koordinat sistemi uygun değildir. Scale metodu ile bu koordinat sistemini yeniden belirleyebiliriz.

Örneğin formun ortasını orijin, sol üst köşeyi -10,10 ve sağ alt köşeyi de 10,10 kabul edecek yeni bir koordinat sistemi oluşturmak için

```
Form1.Scale (-10,10,10,-10)
```

şeklinde kullanılabilir. Bu işlemden sonra uygulayacağınız yazım ve çizim metodları bu koordinat sistemi referans alınarak yapılacaktır.

Hide

Formun Visible özelliğinin False yapılmasını yani formun ekrnda görülmemesini sağlar.

Show[stil]

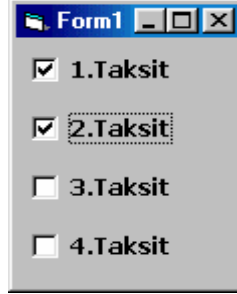
Formun Visible özelliğinin False yapılması gibidir. Parametre ile kullanılırsa formun gösterim şeklini de etkiler.

Burada stil kullanılmazsa veya sıfır kullanılırsa Visible=True ile aynı işi yapar.

Stil değeri 1 ise form Modal olarak gösterir. Form modal ise, form gizlenmeden veya kapatılmadan program içindeki diğer formlara ulaşım engellenir.

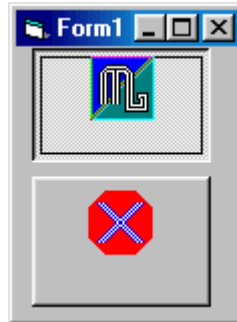
10.4.CheckBox (İşaret Kutusu)

Windows' ta çok kullanılan kontrollerden biri de kullanıcının belirli özellikleri aktif veya pasif hale getirmek için kullanıldığı eleman CheckBox' lardır. Anlaşırılığı ve kullanımı kolay olması sebebiyle kolay kullanılabilir arabirimler oluşturmak için oldukça faydalı bir kontroldür. Form üzerindeki görüntüsü aşağıdaki gibidir.



Komut düğmesinde olduğu gibi Style özelliği ile CheckBox kontrolünün resimli olması da sağlanabilir. Picture, DisabledPicture ve DownPicture özellikleri ile de kontrolün üç durumu için farklı resimler yüklenebilir.

Aşağıdaki Formda Style özelliği 1 (Graphical) yapılmış ve resim yüklenmiş halde CheckBox' lar görülmektedir.



10.4.1.Properties

Aligment

0 değeri için ☐ Check1 işaret solda, 1 değeri için ☒ Check1 işaret sağdadır.

Value

İkisi kullanıcı tarafından değiştirilebilen üç değer almaktadır.

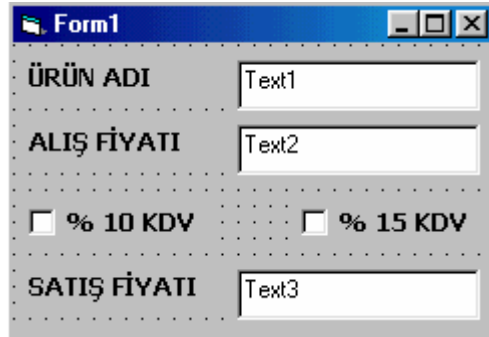
0 değeri için CheckBox işaretsiz ☐

1 değeri için CheckBox işaretli ☒

2 değeri için CheckBox belirsiz ☐ olarak karşımıza çıkar.

İlk iki değer kullanıcı tarafından kontrol tıklanarak değiştirilebilir. Üçüncü değer ise belirsizlik durumunu gösterir ve program tarafından bu durum aktif hale getirilebilir. Belirsizlik Value özelliğinin iki olması ile ilgili değildir sadece o kontrolün ifade ettiği değer belirsiz olduğunu gösterir. Örneğin CheckBox'ın bir metnin Kalın olup olmadığını temsil ettiğini düşünürsek metnin tamamı kalın ise işaretli, tamamı kalın değilse işaretli ve bir kısmı kalın bir kısmı kalın değilse belirsiz olması gerekir.

ÖRNEK: Aşağıdaki Formda yapılan satışa %10 veya %15 KDV' nin dahil edilip edilmeyeceğini CheckBox kontrolü kullanarak gerçekleştirelim.



Program için gerekli program kodu ise aşağıda verilmiştir.

```
Private Sub Check1_Click()  
    Text3.Text = Val(Text2) + (Val(Text2) * 10) / 100  
End Sub
```

```
Private Sub Check2_Click()  
    Text3.Text = Val(Text2) + (Val(Text2) * 15) / 100  
End Sub
```

Program çalıştırıldığında ise aşağıdaki gibi görülecektir.

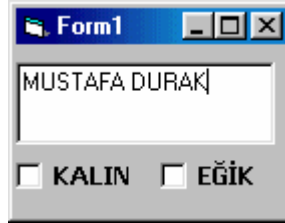


10.4.2.Events

Click ()

CheckBox' da Click olayını boşluk tuşu, kısa yol tuşu ve Mouse tıklaması sonucu meydana gelir. Check kutusunun işareti değiştirilirse bu olay meydana gelir.

ÖRNEK: Örneğin bir Text kutusu içerisindeki yazıyı kalın ve eğik yapılmasını CheckBox ile sağlayalım. Örneğimiz için Formun tasarım görüntüsü aşağıdaki gibi olacaktır.



Program için gerekli program kodları ise şu şekilde olur.

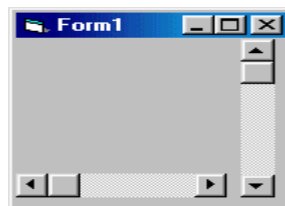
```
Private Sub Check1_Click()  
    Select Case Check1.Value  
        Case 1  
            Text1.FontBold = True  
        Case 0  
            Text1.FontBold = False  
    End Select  
End Sub
```

'Check kontrolünün Value değerine göre
' Case yapısı oluşturuldu. Eğer değer 1
' ise (Check kutusu işaretli) yazı tipi kalın,
' 0 ise yazı tipi normal olacaktır.

```
Private Sub Check2_Click()  
    Select Case Check2.Value  
        Case 1  
            Text1.FontItalic = True  
        Case 0  
            Text1.FontItalic = False  
    End Select  
End Sub
```

10.5.Horizontal & Vertical ScrollBar (Kaydırma Çubuğu)

Biri yatay diğeri dikey olmak üzere her iki kaydırma çubuğu da aynı özelliklere sahiptir ve kullanım amacına göre amaçları da değişir. Örneğin bir veri tabanında kayıtlar arasında ileri geri gitmek için kullanılabilir.

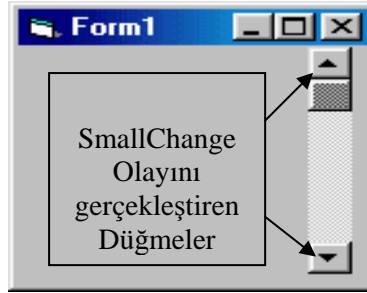


10.5.1.Properties

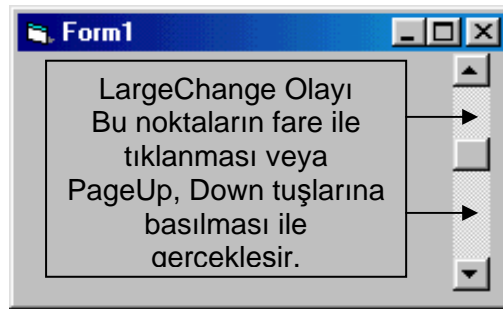
Value

Kaydırma çubuğunun temsil ettiği değeri gösterir, bu değer max ile min arasında bir sayıdır ve kaydırma çubuğu üzerinde değişik şekillerde değiştirilebilir.

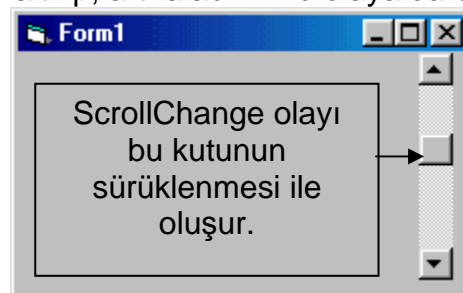
- Kaydırma çubuğunun uçlarındaki iki ok ile azaltılıp çoğaltılabilir. Bu olay SmallChange olayıdır.



- Direkt kaydırma çubuğu üzerinde mouse ile tıklanarak değer artırılıp azaltılabilir. Bu olay LargeChange olarak adlandırılır.



- Kaydırma çubuğunun üzerinde hareket eden kutucuk mouse ile istenen konuma getirilerek değer azaltılıp, artırılabilir. Bu olaya da ScrollChange olayı denir.



Max, Min

Kaydırma çubuğunun alabileceği (temsil edeceği) maksimum ve minimum değerdir.

LargeChange

Kaydırma çubuğu üzerine tıklanması durumunda **Scroll.Value**' nin değişim miktarının ne kadar olacağına belirler.

SmallChange

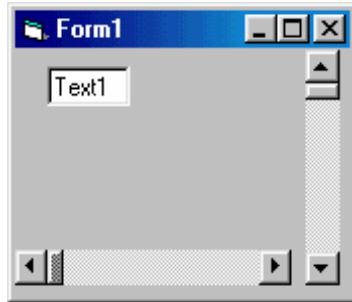
Kaydırma çubuğunun iki kenarındaki oklarla **Scroll.Value**' nin ne kadarlık bir değişime tabi tutulacağını belirler.

10.5.2.Events

Change()

Kaydırma çubuğunun Value özelliğinin değişmesi sonucu Change olayını meydana getirir. Max ve Min değerlerinin değişmesi normalde bu olayı meydana getirmez. Ancak Max veya Min değeri değiştirildiğinde Value bu sınırlar dışında kalıyor ise Value değeri kendiliğinden bu sınır içerisine alınacaktır. Dolayısı ile Change olayı meydana gelecektir. Programlarda bu duruma dikkat etmek gerekmektedir.

ÖRNEK: Örnek olarak Form üzerinde bulunan bir Text kutusunun konumunu yatay ve dikey kaydırma çubukları ile ayarlayabileceğimiz bir program yazalım. Program için gerekli Form ve program kodu aşağıdaki gibi olacaktır.



```
Private Sub HScroll1_Change()  
    Text1.Left = Text1.Left + HScroll1.Value  
End Sub
```

```
Private Sub VScroll1_Change()  
    Text1.Top = Text1.Top + VScroll1.Value  
End Sub
```

10.6.Timer (Zamanlayıcı) Kontrolü

Programda belirli bir süre aralıklar ile aktif hale gelip belirli işleri yapabilmek için kullanılan bir kontroldür. Bu kontrol tasarım zamanda ekranda görülmesine rağmen çalışma esnasında görülmez.

10.6.1.Properties

Bu özellik False verilerek Timer nesnesinin çalışması durdurulur. Tekrar True yapılıncaya kadar Timer nesnesi çalışır duruma gelmez.

Interval

Timer olayının gerçekleşeceği mili saniye cinsinden zaman periyodudur. Alabileceği değerler 1-65535 arasındır. 0 değeri Timer nesnesini pasif hale getirir.

10.6.2.Events

Timer()

Timer kontrolünün Interval özelliği ile belirtilen süre içerisinde periyodik olarak meydana gelir. Bu olay içerisine yazılacak kodun hızlı olması gerekir. Bu olay periyodik olarak sürekli meydana geleceği için bu olay içerisine yazılacak kodun uzun olması Windows altında çalışan diğer programlarında yavaşlamasına sebep olacaktır.

ÖRNEK: Timer kontrolü kullanarak Form başlığında saati gösteren bir program yazalım. Bunun için Timer nesnesinin Interval özelliğini 1000 (1 saniye) yapın.

```
Private Sub Timer1_Timer ( )  
    Form1.Caption = "Saat:" + Timer  
End Sub
```

ÖRNEK: Timer kontrolü kullanarak bir Text veya Label üzerinden kayan bir yazı programı yazalım. Program için 1 adet Timer, 1 adet Text veya Label gerekmektedir.

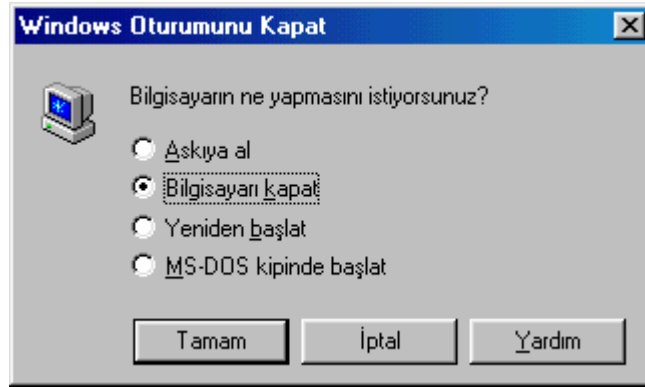
```
Option Explicit  
Dim yazi As String  
  
Private Sub Form_Load()  
    yazi = String(5, " ") + " Mustafa DURAK"  
End Sub  
  
Private Sub Timer1_Timer()  
    yazi = Mid(yazi, 2) & Left(yazi, 1)  
    Text1.Text = yazi  
End Sub
```

ÖRNEK: Timer kontrolü ile ayarlanan saat için alarm çalan bir program kodu yazalım. Bu kod için bir Timer ve adı ZAMAN olan bir Text kutusu gerekmektedir.

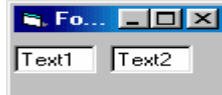
```
Option Explicit  
Dim SAAT,DAKK,SANY As Byte  
Private Sub Form_Load()  
    SAAT= Inputbox("Alarm Saatini Giriniz")  
    DAKK= Inputbox("Alarm Dakikasını Giriniz")  
    SANY= Inputbox("Alarm Saniyesini Giriniz")  
    ZAMAN.Text= SAAT + ":" + DAKK + ":" + SANY  
End Sub  
  
Private Sub Timer1_Timer()  
    If (Hour(Now) = Val(Trim(SAAT))) And (Minute(Now) =  
    Val(Trim(DAKK))) And (Second(Now) = Val(Trim(SANY))) Then  
        beep: beep: beep  
    Else  
  
    End If
```

10.7.OptionButton (Seenek Dğmesi) Kontrol

OptionButton kontrol ChechkBox kontrolnden farklı olarak birkaç seenekten sadece birini seme imkanı veren bir kontroldr. Bu kontroln tek baına kullanılması anlamsızdır. Birkaç seenekten birini seme imkanı veren bir kontrol olduėu iin en az iki tane birlikte kullanılmalıdır. Gruptaki OptionButton dğmelerinden biri seildiėinde diėeri kendiliėinden seilmiř zelliėini kaldırır. Yani anda bir grupta iki tane iřaretli dğme bulunmaz. rneėin bilgisayarımızı kapatmak iin kullandığımız Bilgisayar Kapat komutu verildikten sonra karřımıza ıkan pencereden yalnızca bir seenek seebiliriz ve bilgisayarımızda o seeneėe gre hareket eder.



RNEK: OptionButton kontrol kullanarak seime gre Text1 ve Text2 kutularında bulunan sayıları toplayan, ıkaran, arpan veya blen programı tasarlayınız. (Text1' in deėeri Text2' nin deėerinden byk olmalıdır.) Program iin gerekli Form ařaėıdaki olmalıdır.



```
Private Sub Option1_Click()  
    Text3 = Val(Text1) + Val(Text2)  
End Sub  
  
Private Sub Option2_Click()  
    If Val(Text1) > Val(Text2) Then  
        Text3 = Val(Text1) - Val(Text2)  
    Else  
        End If  
End Sub  
  
Private Sub Option3_Click()  
    Text3 = Val(Text1) * Val(Text2)  
End Sub  
  
Private Sub Option4_Click()  
    If Val(Text1) > Val(Text2) Then  
        Text3 = Val(Text1) / Val(Text2)  
    Else  
        End If  
End Sub
```

ÖRNEK: Herhangi bir kullanıcının adını, medeni hali ve tahsil durumunu öğrenen ve bu bilgileri Form üzerinde bulunan Text kutularına yazan bir program tasarlayalım. Program için aşağıda görüldüğü gibi bir Form hazırlayın.

Option Explicit

Dim isim As String

Private Sub Form_Load()

 isim = InputBox("Adınızı ve Soyadınızı giriniz..:") ' Kullanıcının ismini sor

 Text1.Text = isim 'İsmi Text1 kutusunda göster

End Sub

Private Sub Option1_Click()
 Text2.Text = Option1.Caption
End Sub

Evli düğmesi seçilirse evli düğmesinin
Captionı' nı Text2' ye yaz.

Private Sub Option2_Click()
 Text2.Text = Option2.Caption

End Sub

Private Sub Option3_Click()
 Text2.Text = Option3.Caption
End Sub

Private Sub Option4_Click()
 Text3.Text = Option4.Caption
End Sub

Private Sub Option5_Click()
 Text3.Text = Option5.Caption
End Sub

Private Sub Option6_Click()
 Text3.Text = Option6.Caption
End Sub

10.7.1.Properties

Value

Bu özellik True ise option düğmesinin seçilmiş olduğunu gösterir. Option düğmelerinin yalnızca iki durumu vardır. Bunlar seçili olma ve olmamadır.

ÖRNEK: Örnek olarak bir Text kutusu içerisindeki yazıyı sağa sola ve ortaya hizalayan bir program yapalım. Program için gerekli Form ve program kodları şu şekilde olmalıdır. (Hizalama için gerekli hizalama numaralarını Text' in Aligment özelliğinden aldım.)



```
Private Sub Option1_Click()  
Text1.Alignment = 1 'Sağa hizala  
End Sub
```

```
Private Sub Option2_Click()  
Text1.Alignment = 0 'Sola Hizala  
End Sub
```

```
Private Sub Option3_Click()  
Text1.Alignment = 2 'Ortaya Hizala  
End Sub
```

10.8.Frame (Çerçeve) Kontrolü

Bu kontrol tek başına değil, diğer kontrolleri gruplamak için kullanılır. Kontrolleri bu kontrolle gruplamanın bir çok avantajı vardır.

Bu çerçeveler içine konan kontroller, çerçeveye bağımlıdırlar ve konumları bu çerçeve dışına çıkamaz. Özellikle birkaç kontrolü birden görünür veya görünmez yapmak için hepsinin Visible özelliğini değiştirmek yerine çerçevenin Visible özelliğini değiştirerek aynı işlemi bir adımda yapmış oluruz. Aynı şekilde taşınma özelliğinde de çerçeve taşınarak işlem tek adımda gerçekleştirilebilir. Yani kullandığımız Frame kontrolünü Form içerisindeki başka bir Form gibi düşünebiliriz.

Frame' lerle gruplanan kontrollerin koordinatları artık forma göre değil grup kontrolünün sol üst köşesine göre belirlenir.

□ Form üzerine yerleştirilmiş bir kontrolü taşıyarak bir Frame kontrolü üzerine getirmekle o kontrol gruplanmış olmaz. Frame kontrolü içerisine bir kontrol yerleştirirken önce Frame kutusunu seçin. Eğer Form üzerinde bulunan kontrolleri Frame kontrolü içerisine almak istiyorsanız o zaman Form üzerindeki kontrolleri kesin (Cut) ve Frame kontrolünü seçtikten sonra buraya yapıştırın (Paste).

Ayrıca Frame, kontrolleri sadece bazı şartlarda gösterilmesi gereken kontrolleri de bir arada tutarak bunların kolayca gizlenip gösterilmesini sağlarlar.

ÖRNEK:Örneğin yukarıdaki Form dizaynında “Metin Hizalama” seçeneği seçilirse “Metin Hizalama” Caption’ ılı Frame2, “Yazı Tipi Seçimi” seçeneği seçilirse “Yazı Tipi Seçimi” Caption’ ılı Frame3 kontrolünün görülmesi için gerekli kodu yazalım. Bu program kodunu yazarken öncelikle sadece Frame1 kontrolü görülecek diğerleri ise görülmeyecektir. Frame2 ve Frame3 yalnızca seçildikleri zaman görüneceklerdir. Bunun için başlangıçta Frame2 ve Frame3’ ün Visible özellikleri False yapılacaktır. Bunun için gerekli program kodu aşağıdaki gibi olacaktır.

```
Private Sub Form_Load()  
    Frame2.Visible = False  
    Frame3.Visible = False  
End Sub
```

```
Private Sub Option1_Click()  
    Frame2.Visible = True    'Frame2 aktif iken Frame3 görünmez yapılır.  
    Frame3.Visible = False  
End Sub
```

```
Private Sub Option2_Click()  
    Frame3.Visible = True    'Frame3 aktif iken Frame2 görünmez yapılır.  
    Frame2.Visible = False  
End Sub
```

10.9.Line (Çizgi) Kontrolü

Bu kontrol elemanı Form üzerine bir çizgi şeklinde alınır. Alınan bu çizgiyi yatay, dikey ve eğik bir biçimde göstermek, istenilen boyuta getirmek mümkündür. Bu kontrolün yerine Line(X1,Y1) – (X2,Y2) metodunu kullanabiliriz.

X1, Y1, X2, Y2

X1 ve Y1 noktaları çizgi kontrol elemanın başlangıç koordinatlarını, X2 ve Y2 ise bitiş koordinatlarını belirler. Yatay kontrolleri X1 ve X2, dikey kontrolleri ise Y1 ve Y2 belirler.

10.10.PictureBox (Resim Kutusu) Kontrolü

Bu kontrol elemanı Bitmap, Icon, Metafile, Jpeg ve Gif gibi resimleri görüntülemek için kullanılır. Ayrıca metotlar kullanılarak PictureBox içine çizimlerde yapılabilir. Bu kontrolün bir diğer özelliği ise aynen Frame kontrolünde olduğu gibi diğer kontrolleri gruplandırmasıdır.

10.10.1.Properties

Picture

Bu özellik kullanılarak Picture kutusunun göstereceği resim belirlenir. Bu resim BMP, ICO, JPG, GIF veya WMF türünden olabilir.

Picture kutusuna resim yüklenmesi:

❑ Tasarım anında Properties penceresinde yer alan Picture özelliği kullanılarak karşımıza çıkan diyalog penceresinden istediğimiz resmi seçerek PictureBox içerisine yerleştirebiliriz.

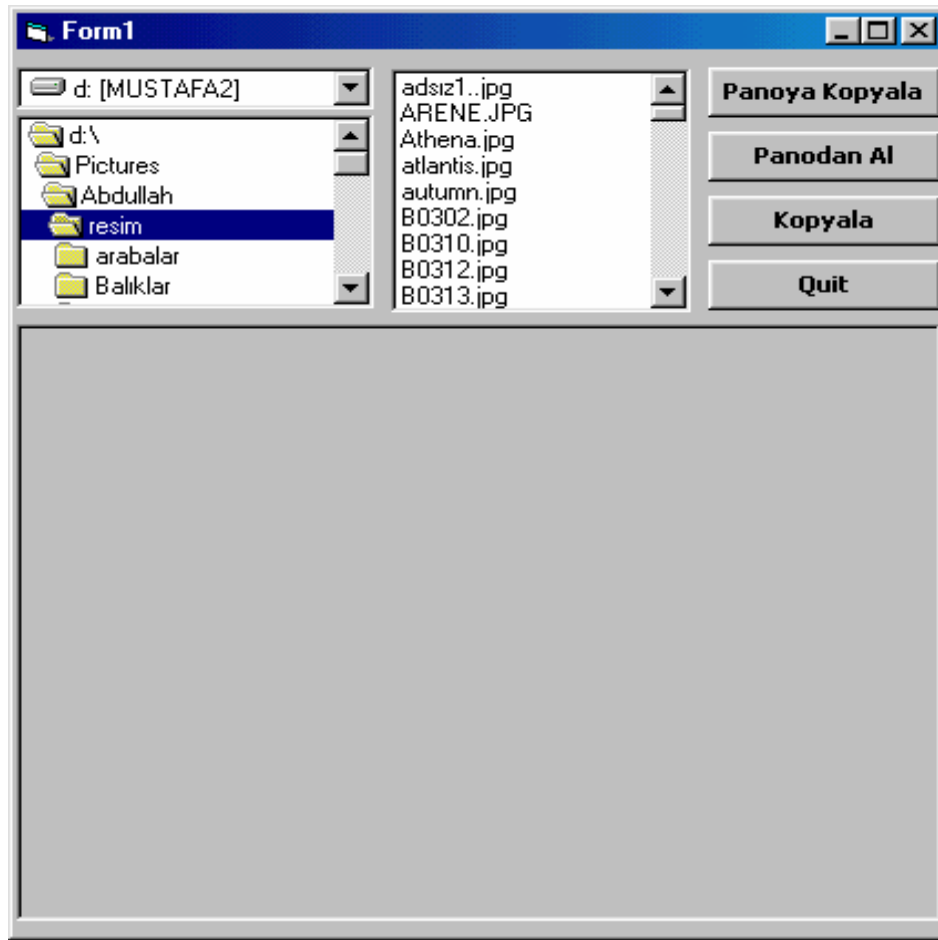
❑ **LoadPicture** komutu ile programın çalışması esnasında dosyadan yüklenebilir.

```
Picture1.Picture = LoadPicture ("C:\Windows\Kur.Bmp")
```

❑ **Clipboard**’ dan yapılabilir.

```
Picture1.Picture = Clipboard.GetData( )
```

ÖRNEK:Picture özelliğini kullanarak resim kutusuna dosyadan veya panodan resim yükleyecek ve kaydedecek bir program yapalım. Bunun için Form üzerine bir PictureBox, bir DriveListBox, bir FileListBox, bir DirectoryBox ve dört tanede komut düğmesi yerleştirerek aşağıdaki Formu oluşturun.



Programın çalışması için gerekli kodlar aşağıda verilmiştir. Program kodlarını genişleterek bir adet resim editörü elde edebiliriz. Burada kullanılan DriveBox, FileListBox ve DirectoryListBox elemanları kitabımızın ilerleyen bölümlerinde ayrıntıları ile verilecektir.

```
Private Sub Command1_Click()  
    Clipboard.Clear  
    Clipboard.SetData Picture1.Picture  
End Sub  
  
Private Sub Command2_Click()  
    Picture1.Picture = Clipboard.GetData  
End Sub  
  
Private Sub Command3_Click()  
    On Error GoTo hata  
    SavePicture Picture1.Picture, InputBox("Dosya adı")  
    hata:  
End Sub  
  
Private Sub Command4_Click()  
    End  
End Sub  
  
Private Sub Dir1_Change()  
    File1.Refresh  
    File1.Path = Dir1.Path  
    ChDir (File1.Path)  
End Sub
```

10.11.Image

Bu özellik sadece okunabilir bir özelliktir ve Picture kontrolü içine değişik metotların (Print, Line, Circle) kullanımı ile yapılan yazım ve çizimleri temsil eder. Bu metotlar ile yapılan yazım ve çizimlerin kaydedilebilmesi veya başka bir nesne içine alınabilmesi için bu özellik kullanılır. Bu özelliğin etkili olabilmesi için AutoRedraw özelliğinin True olması gerekir.

```
Picture1.Circle (0,0),150  
Picture2.Picture = Picture1.Image
```

Yukarıdaki satırlar ile Picture1 içine bir daire çizilmekte ve bu daire Picture2 nesnesi içerisinde gösterilmesi sağlanmaktadır.

10.10.1.Properties

AutoRedraw

Bu özelliğin aldığı True veya False değerleri ile nesnenin kendini otomatik olarak yenilenmesi veya yenilenmemesi sağlanır. Bu özellik sayesinde arka plana düşmüş bir Form yada üzeri kapatılmış bir nesnenin üzerinin açılması ile nesnenin kendisini yeniden çizdirilmesi sağlanır. Picture kutusunda metotlar kullanılarak yapılmış yazım ve çizimler varsa PictureBox' un kendini yenilenebilmesi için bu özellik True olmalıdır.

ScaleLeft, ScaleTop

Bu özellikler Left ve Top özelliklerinden farklıdır. Left ve Top nesnenin Form içindeki koordinatlarını belirlerken ScaleLeft ve ScaleTop çizimin yapılacağı bölgenin koordinatlarını belirler. Default değeri 0,0' dır, yani yapılacak çizimler nesnenin sol üst

köşesi orijin olmak üzere yapılır. Pozitif değerler orijini sol yukarı taşırken, negatif değerler sağ aşağı taşır.

ScaleHeight, ScaleWidth

ScaleHeight ve ScaleWidth özellikleri çizimin yapılacağı bölgenin genişliğini belirler. Nesne içine yapılacak çizimler bu koordinatlar dışına taşamaz.

Bu özellikler nesne içinde Picture özelliği ile yüklenen resmi etkilemez. Sadece nesnenin metotları ile yapılmış yazım ve çizimleri etkiler.

Align

Bu özellik kullanılarak Picture kontrolünün sürekli olarak Formun bir kenarında kalması sağlanabilir. Örneğin bir araç çubuğunun sürekli olarak Formun en üstünde kalması istenir veya durum çubuğunun Formun sürekli en altında kalması istenir. İşte bu özellik kullanılarak herhangi bir kod yazmaya gerek kalmadan kendisini sürekli aynı kenarda tutması sağlanabilir. Align özelliğinin seçenekleri aşağıda verilmiştir.

0-None	Yerleştiği koordinatlarda sabit kalır
1-Align Top	Sürekli üstte
2-Align Bottom	Sürekli altta
3-AlignLeft	Sürekli solda
4-AlignRight	Sürekli sağda

Align özelliğine 0 olamayan bir değer verildiğinde, Formun boyutlarının değişmesi ile kontrol elemanı kendini otomatik olarak boyutlandırır. Form boyutları büyüdüğünde kontrol elemanı da büyüme uygun olarak kendini boyutlandır.

10.10.2.Events

Resize()

Kontrolün boyutlarının değişmesi halinde bu olay meydana gelir. Picture boyutları iki şekilde değişir. Birincisi, programın herhangi bir yerinden kontrolün genişliğinin veya yüksekliğinin değiştirilmesi ile, ikincisi de PictureBox' ın AutoSize özelliği True ise Picture kontrolünün içindeki resmin değişmesi ile PictureBox' ın boyutları resmin boyutlarına ayarlanacağından Resize olayı meydana gelir.

10.10.3.Methods

PaintPicture

Resim işleme için geliştirilmiş bir metottur. Bu metot ile bir resmin ölçeklenmesi (büyütülüp, küçültme), taşınması, ters çevrilmesi ve üzerinde işlem yapılması (negatifinin alınması gibi işlemler) mümkündür.Yazım formatı aşağıdaki gibidir.

PaintPicture(KaynakPicture,HedefX,HedefY,HedefGenişlik,HedefYükseklik, KaynakX,KaynakY,KaynakGenişlik,KaynakYükseklik,İşlem)

KaynakPicture parametresi ile işleme girecek resim belirlenir. PaintPicture metodunda hedef resim ise kontrolün kendisidir. Örneğin Picture1' in PaintPicture metodu kullanılıyorsa hedef resim Picture1 içindeki resimdir. Kaynak resim olarak ta aynı kontrolün Picture özelliği verilebilir.

KaynakX, KaynakY Parametreleri ile işlenecek resmin sol üst köşe koordinatları ve KaynakGenişlik, KaynakYükseklik parametreleri ile de boyutları belirlenebilir.

HedefX, HedefY, HedefGenişlik, HedefYükseklik parametreleri ile de işlenecek resmin gösterileceği kontroldeki koordinatları ve boyutları belirlenir.

ÖRNEK: Kaynak resmin boyutları hedef resmin boyutlarından daha küçük verilerek resim küçültülebilir veya daha büyük verilerek resim büyütülebilir.

Private Sub Picture1_Click()

```
Picture2.PaintPicture Picture1.Picture,0,0,Picture1.ScaleWidth/2,  
Picture1.ScaleHeight/2,0,0,Picture1.ScaleWidth,Picture1.ScaleHeight,
```

Kod yazıldıktan sonra Picture1 nesnesine Properties penceresinden bir resim yükleyiniz. Programı çalıştırdıktan sonra Picture1 nesnesi üzerine mouse ile tıklama yaptıktan sonra aşağıdaki görüntüyü elde ederiz.



ÖRNEK: Hedef resmin boyutlarını negatif vererek resmi ters çevirmek mümkündür. Sadece hedefin genişliği negatif verilirse yatay olarak, yüksekliği negatif verilirse dikey olarak resim ters çevrilmiş olur. Her ikisi birden ters verilerek resim her iki yönde de ters çevrilebilir. Aşağıda konu ile ilgili program ve çıktısı görülmektedir.

Private Sub Picture1_Click()

```
Picture2.PaintPicture Picture1.Picture, Picture1.ScaleWidth, Picture1.ScaleHeight, -  
Picture1.ScaleWidth, -Picture1.ScaleHeight, 0, 0, Picture1.ScaleWidth,
```

```
Picture1.ScaleWidth, Picture1.ScaleHeight, &H550009
```



İşlem parametresi ile resme uygulanacak işlem belirlenir. Bu parametreye aşağıdaki değerler verilmektedir.

&H550009 : Hedef resmin renklerini tersine çevirir.

&H330008: Kaynak resmin tersi alınarak hedefe kopyalanır.

&H1100A6: Kaynak resim ile hedef resim OR işlemine tabi tutulur ve sonucun tersi alınarak hedefe gösterilir.

&H8800C6: Kaynak resim ile hedef resim AND işlemine tabi tutularak hedefte gösterilir.

&HCC0020: Kaynak resim hiçbir işleme tabi tutulmadan hedefte gösterilir.

&H660046: Kaynak resim ile hedef resim XOR işlemine tabi tutularak hedefte gösterilir.

ÖRNEK: PaintPicture metodunu kullanarak bir resmi Form üzerine döşeyecek programı yazalım. Örneğimiz için Formumuzun üzerine bir PictureBox yerleştirin ve içine döşenmesini istediğiniz resmi alın.

Private Sub Form_Load()

```
Picture1.Visible = False
```

```
End Sub
```

Private Sub Form_Paint()

```
Dim i, j
```

```
For i = 0 To ScaleWidth Step Picture1.Width
```

```
For j = 0 To ScaleHeight Step Picture1.Height
```

```
PaintPicture Picture1.Picture, i, j, Picture1.Width, Picture1.Height, 0, 0
```

```
Next
```

```
Next
```

```
End Sub
```

Private Sub Picture1_Click()

```
Picture2.PaintPicture Picture1.Picture, Picture1.ScaleWidth,
```

```
Picture1.ScaleHeight, -Picture1.ScaleWidth, -Picture1.ScaleHeight, 0, 0,
```

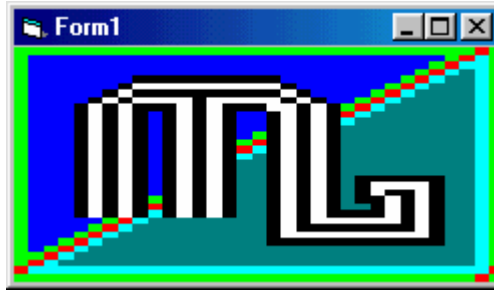
```
Picture1.ScaleWidth, Picture1.ScaleHeight, &H550009
```

```
End Sub
```

Program çalıştırıldığında aşağıdaki pencere görüntüsünü elde ederiz.



Eğer PictureBox' ın boyutu Form ile aynı boyutta ise görünümü şu şekilde olur.



10.11.Image (Resim Gösterme) Kontrolü

Bu eleman grafiksel bir kontrol olup resimleri görüntülemek, boyutlandırmak ve taşımak için kullanılır. PictureBox' dan daha hızlı ve daha az sistem kaynağı kullanmasına rağmen PictureBox kadar gelişmiş özelliklere sahip değildir. PictureBox' tan farklı olarak gruplandırma yapamaz ve metotları kullanarak yazım ve çizim yapamaz.

10.11.1.Properties

Picture

BMP, ICO, WMF, DIB, JPEG, GIF resim formatlarını destekler. Bu özellik, PictureBox kısmında detaylı olarak açıklanmıştır.

Stretch

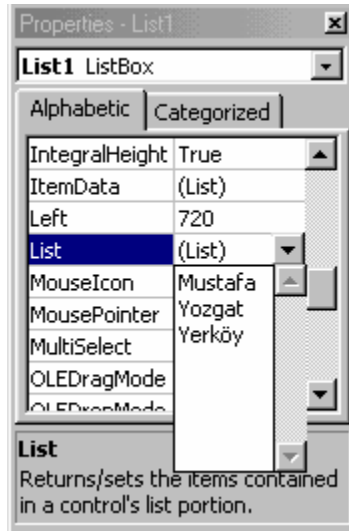
Bu özellik resmin Image kutusu kadar büyütölüp, büyütölmeyeceğini belirler. True ise resim, kutunun içine sığdırılır. Yani resim büyük ise küçültölür, küçük ise büyütölür. False ise Image kutusunun boyutları resmin boyutlarına ayarlanır.

10.13.ListBox (Listeleme Kutusu) Kontrolü

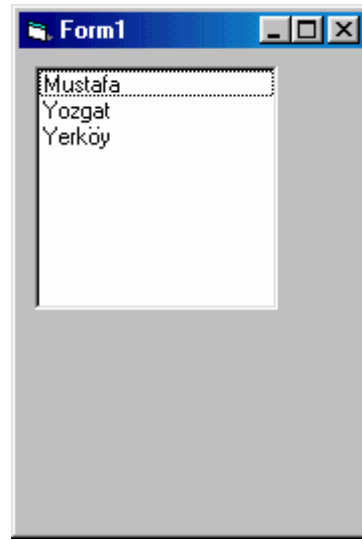
Visual Basic' in sağladığı dizilerinizi gösterebileceğiniz kontrollerdendir. Elemanları listelemek , sıralamak amacı ile kullanılan bir kontroldür. DriveListBox, DirectoryListBox, FileListBox gibi türleri vardır.

Listeye eleman eklemek için AddItem metodu kullanılabileceği gibi, tasarım anında da List özelliği kullanılabilir.

Eğer listeye tasarım anındayken elemanlar eklemek isterseniz Properties penceresinden List özelliğini aşağı doğru açın ve elemanları yazın. Ancak elemanlar arasında Enter tuşunu değil Ctrl+Enter tuşunu kullanın.



Listeye eleman ekleme



Eklenen Elemanların Görüntüsü

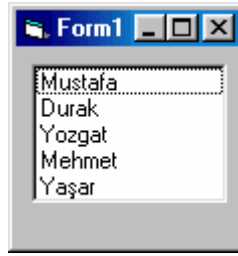
10.13.1.Methods

AddItem Eleman, SıraNo

Listeye eleman ilave etmek için kullanılan bir metottur. Liste sıralı değilse elemanlar eklendiği sıraya göre eklenir. Elemanı sona değil de belirli bir sıraya yerleştirmek isterseniz son parametre olarak yerleştirileceği sıra numarasını verebilirsiniz.

```
List1.AddItem "Mustafa"  
List1.AddItem "Durak"  
List1.AddItem "Mehmet"  
List1.AddItem "Yaşar"  
List1.AddItem "Yozgat",2
```

Bu kodlar aracılığı ile, aşağıdaki gibi bütün elemanlar eklendiği sırayla yerleştirecektir. Bu kodlardan sonra aşağıdaki ekran görüntüsü elde edilir.



Bu koddaki ilk dört satır satırda sıra numarası verilmediği için ilk dört eleman eklenir ancak son satırda “Yozgat” elemanın 2. sıraya eklenmesi istenmektedir. İlk sıra numarası 0 olduğu için bu listede 3. sıraya eklenecektir. Çünkü “Mustafa” elemanı listenin 0. elemanıdır.

RemoveItem Index

Listedeki index nolu elemanı listeden çıkarmaya yarayan bir metottur. İlk elemanın index numarası 0 olduğu için elemanların listedeki yerinin bir eksiği index olarak verilmelidir.

List1.RemoveItem 1 ‘ 2 nolu elemanı listeden çıkarır.
List1.RemoveItem List1.ListIndex ‘Seçili elemanı listeden çıkarır.

Clear

Listedeki bütün elemanları siler yani listeyi boşaltır.

List1.Clear ‘Listedeki elemanları çıkarır.

10.13.2.Properties

ListCount

Liste içindeki eleman sayısını verir.

A=List1.ListCount+1
Msgbox “Listede” + Str(A) + “tane eleman var. ”

ListIndex

Listedeki aktif (seçili) elemanın liste içindeki numarasını verir. Ayrıca listedeki aktif yapılacak elemanı belirler. Bu numara ilk eleman için 0, son eleman için ListCount-1’ dir.

List1.ListIndex=5 ‘Listedeki 6. elemanı seçer.

List(Index)

Liste içindeki Index numaralı elemanın değerini öğrenmek veya değiştirmek için kullanılır. Bu özellik tasarım zamanında listeye eleman ilave etmek için de kullanılabilir.

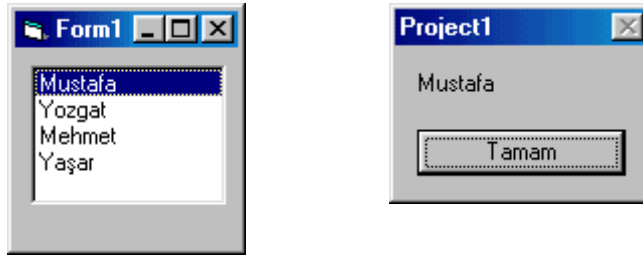
```
List1.List(1) = "Mustafa" '2. elemanı Mustafa yapar.  
AltinciEleman = List1.List(5) ' 6. elemanı değişkene atar.
```

Text

Liste içinde seçili olan elemanın içeriğini verir. Sadece okunabilir bir özelliktir.

```
Private Sub List1_Click()  
MsgBox List1.Text 'Seçili elemanı ekrana mesaj kutusu şeklinde gösterir.  
End Sub
```

Eğer "Mustafa" isimli eleman seçili ise aşağıdaki mesaj kutusu karşımıza çıkar.



MultiSelect

Bu özellik bir kutu içinde birden fazla elemanı seçme imkanı verir. Üç değişik biçimi vardır.

- 0: Birden fazla eleman seçimi yapılamaz.
- 1: Mouse ile tıklanan her eleman seçilir veya seçilmiş ise seçilmişliği kaldırılır.
- 2: Bu modda Shift veya Ctrl tuşu basılı tutularak eleman seçimi yapılabilir.

Bu özellik sadece tasarım zamanında değiştirilebilir. Çalışma zamanında yapılacak atama ile listenin MultiSelect özelliği değiştirilemez.

SelectCount, Selected(Index)

MultiSelect özelliği 0 olmayan listelerde birden fazla eleman seçilebileceği için **SelectCount** özelliği ile seçili eleman sayısı, **Selected(Index)** özelliği ile de index numaralı elemanın seçili olup, olmadığı öğrenilebilir.

Seçili elemanları öğrenebilmek için listeyi baştan sona taratıp **Selected** özelliğinin kontrol edilmesi gerekmektedir.

Aşağıdaki gibi bir kod ile **MultiSelect** özelliği verilmiş bir listedeki elemanlardan seçili olanları öğrenebiliriz. Bunun için aşağıdaki Formu oluşturun.



```
Private Sub Command1_Click()  
Dim i  
For i = 0 To List1.ListCount - 1  
If List1.Selected(i) = True Then  
List2.AddItem List1.List(i)  
Else  
End If  
Next  
End Sub
```

Yukarıdaki kod yazıp programı çalıştırdıktan sonra bazı elemanlara seçer ve “Aktar” Caption’ lı komut düğmesine basarsak aşağıdaki gibi bir görünüm elde ederiz.

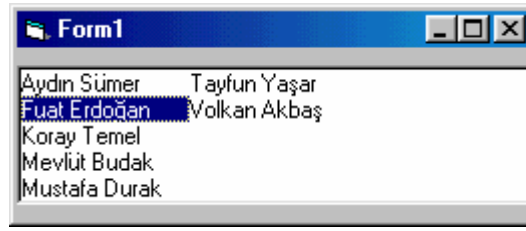


Sorted

Listenin alfabetik sıralı olmasını sağlar. Listeye eklenen elemanları sona değil alfabetik sıraya yerleştirir. Bu özellikte tasarım anında ayarlanabilen bir özelliktir. **Sorted** özelliğinin değeri True yapılırsa listeye eklenen elemanlar sıralanır.

Columns

Bu özellik ile liste kutusu birkaç kolon yapılabilir. Eğer değeri 0 ise liste kutusu tek sütundur ve listenin ekranda görülen kısmının dolması ile listeye dikey kaydırma çubuğu eklenir. 0’ dan farklı bir değer ise listenin genişliği verilen sayıda sütuna bölünür ve bir sütunun dolması ile ikinci sütuna geçilir. Listenin görülen kısımlarının dolması ile listeye yatay kaydırma çubuğu eklenir. Örneğin **Columns** özelliği 3 olan bir liste aşağıdaki gibi görünür.



IntegralHeight

Bu özelliğin değeri True ise liste kutusu içindeki elemanları tam gösterebilmek için kendisini yeniden boyutlandırır. Normalde bu özellik True değerindedir ve liste kutusunun boyutu belirli aralıklar ile değiştirilebilir.

Style

Liste kutusu bir **Checkbox** dizisi gibide kullanılabilir. **Style** özelliğine 1 değeri verildiğinde listedeki elemanlar birer **CheckBox** gibi davranır. Yani kullanıcı listedeki elemanları işaretleyerek seçebilir. Listedeki bir eleman seçildiğinde **ItemCheck** olayı meydana gelir.



10.13.2.Events

Click, DblClick ()

Listeden bir eleman seçildiğinde **Click**, bir eleman üzerinde mouse ile iki kere tıklandığında ise **DblClick** olayı meydana gelir.

ItemCheck (Item As Integer)

Listenin Style özelliğine 1- **CheckBox** verilmiş ise listedeki elemanların birer **CheckBox** gibi olacaklarını yukarıda açıklamıştık. Bu durumda listeden bir eleman işaretlendiğinde **ItemCheck** olayı meydana gelir. Buradaki **Item** parametresi işaretlenen elemanın listedeki index numarasıdır.

ÖRNEK: Genel bir örnek olarak öğrenci kimlik bilgilerinin girilebileceği bir program yazalım. Örneğimiz için aşağıdaki formu hazırlayın.

Ekle düğmesi, Text kutularına girilenleri ayrı ayrı listelere yazacak.
Sil düğmesi, seçili olan elemanı silecek.
Bul düğmesi ile Text1' e ismi girilen elemanı listede bulup seçeceğiz.
Değiştir düğmesi ile listede seçili elemanı Text kutularındakilerle değiştireceğiz.
Araya Ekle düğmesi ile Text kutularındakileri seçili elemanın önüne ekleyeceğiz.

```
Private Sub Command1_Click()
```

```
List1.AddItem Text1.Text
```

```
List2.AddItem Text2.Text
```

```
List3.AddItem Text3.Text
```

```
List4.AddItem Text4.Text
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
If List1.ListIndex < 0 Then
```

```
MsgBox "Önce silinecek elemanı seçiniz", vbInformation
```

```
Exit Sub
```

```
End If
```

```
Dim cevap, ind
```

```
cevap = MsgBox(List1.List(ind) & "Silinsin mi?", vbYesNo +  
vbExclamation + vbDefaultButton2, "Sil")
```

```
ind = List1.ListIndex
```

```
If cevap = vbNo Then Exit Sub
```

```
List1.RemoveItem ind
```

```
List2.RemoveItem ind
```

```
List3.RemoveItem ind
```

```
List4.RemoveItem ind
```

```
End Sub
```

```
Private Sub Command3_Click()  
Text1 = "": Text2 = "": Text3 = "": Text4 = ""  
Text1.SetFocus  
Dim i  
For i = 0 To List1.ListCount - 1  
If UCase(Text1.Text) = UCase(List1.List(i)) Then  
List1.ListIndex = i  
Exit Sub  
Else  
MsgBox "Aranan kayıt bulunmadı", vbInformation  
Exit Sub  
End If  
Next  
End Sub
```

```
Private Sub Command4_Click()  
Dim ind  
If List1.ListIndex < 0 Then  
MsgBox "Değiştirilecek kayıttı seçin", vbInformation  
Exit Sub  
End If  
ind = List1.ListIndex  
List1.List(ind) = Text1  
List2.List(ind) = Text2  
List3.List(ind) = Text3  
List4.List(ind) = Text4  
End Sub
```

```
Private Sub Command5_Click()  
Dim ind  
If List1.ListIndex < 0 Then  
MsgBox "Öğrencinin nereye ekleneceğini seçiniz", vbInformation  
Exit Sub  
End If  
ind = List1.ListIndex  
List1.AddItem Text1.Text, ind  
List2.AddItem Text2.Text, ind  
List3.AddItem Text3.Text, ind  
List4.AddItem Text4.Text, ind  
End Sub
```

```
Private Sub Command6_Click()  
End  
End Sub  
Private Sub List1_Click()  
List2.ListIndex = List1.ListIndex  
List3.ListIndex = List1.ListIndex  
List4.ListIndex = List1.ListIndex  
End Sub
```

```
Private Sub List1_Click()  
List2.ListIndex = List1.ListIndex  
List3.ListIndex = List1.ListIndex  
List4.ListIndex = List1.ListIndex  
End Sub
```

```
Private Sub List2_Click()  
List1.ListIndex = List2.ListIndex  
List3.ListIndex = List2.ListIndex  
List4.ListIndex = List2.ListIndex  
End Sub
```

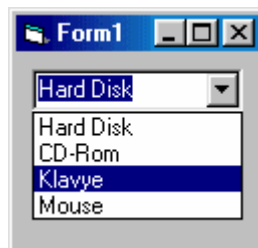
```
Private Sub List3_Click()  
List1.ListIndex = List3.ListIndex  
List2.ListIndex = List3.ListIndex  
List4.ListIndex = List3.ListIndex  
End Sub
```

```
Private Sub List4_Click()
```

```
List1.ListIndex = List4.ListIndex  
List2.ListIndex = List4.ListIndex  
List3.ListIndex = List4.ListIndex
```

10.14.ComboBox (Açılan Liste) Kontrolü

Genellikle değerleri daha önceden belli olan elemanların seçimi için kullanılır. Liste kutusuna benzer ancak listedeki elemanlardan sadece seçileni ekranda görüntülenir. Aşağı doğru açılabilen bir liste kontrolüdür. ComboBox kontrolüne eleman ekleme ve silme işlemi ListBox' ta olduğu gibidir.



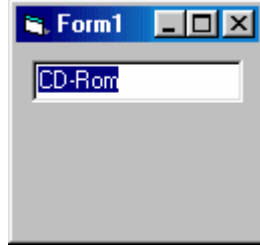
10.14.1.Properties

Style

0,1,2 değerlerinden birini alabilen bu özellik ile ComboBox' un tipi belirlenir.

0: Aşağı doğru açılan ve içeriği kullanıcı tarafından değiştirilebilen ComboBox oluşturur.

1: Aşağı doğru açılmayan fakat içeriği kullanıcı tarafından değiştirilebilen ComboBox oluşturur. Listedeki elemanı seçmek aşağı ve yukarı tuşlarıyla yapılır, listedeki bütün elemanlar aynı anda görüntülenemez.



2: Aşağı doğru açılabilen fakat kullanıcı tarafından giriş yapılamayan ComboBox oluşturur. Kullanıcı listedeki değerlerden birini seçebilir ama değiştiremez.

Text

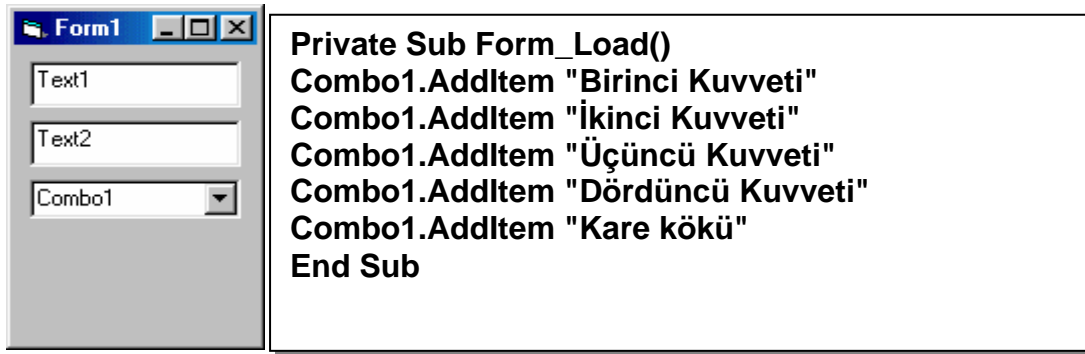
Combobox kutusundaki aktif olan elemanın içeriği bu özellik ile öğrenilebilir.

ListIndex

ComboBox kutusundan seçilen elemanın numarasını verir.

ComBox kutuları ile **ListBox** kutularının özellikleri hemen hemen aynıdır. Bunları ListBox kutularında açıkladığımız için bu kısımda tekrar anlatılmadı.

ÖRNEK: Bir ComboBox kullanarak Text1' de yer alan sayının birinci, ikinci, üçüncü, dördüncü kuvvetleri ile kare kökünü alıp Text2 kutusuna yazan bir program yazalım. Önce aşağıdaki formu hazırlayalım ve verilen kodları uygun alt yordamlara yazalım.



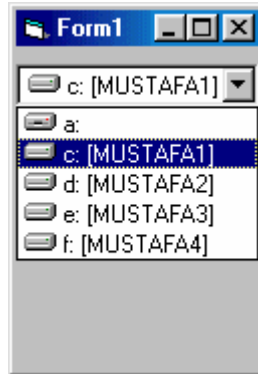
Burada yaptıracağım işlemleri ComboBox kontrolünden seçeceğim için Form_Load() yordamında yaptıracağım işlemlerin adını AddItem metodu ile ComboBox içerisine ekledim. Şimdi ComboBox' un Click yordamına aşağıdaki kodu ekleyerek programımızı bitirelim. (Click yordamına ekleme sebebi değer değiştirilebilmek için mutlaka Click olayı gerçekleştirilmelidir.)

```
Private Sub Combo1_Click()  
Select Case Combo1.Text  
Case "Birinci Kuvveti"  
Text2 = Val(Text1)  
Case "İkinci Kuvveti"  
Text2 = Val(Text1) ^ 2  
Case "Üçüncü Kuvveti"  
Text2 = Val(Text1) ^ 3  
Case "Dördüncü Kuvveti"  
Text2 = Val(Text1) ^ 4  
Case "Kare Kökü"  
Text2 = Sqr(Val(Text1))  
End Select  
End Sub
```

Bu program kodunda ise ComboBox' ın Text özelliğine göre bir Case yapısı kullanılarak istenilen işlem yaptırılmaktadır. Kullanıcı mecburen ComboBox kutusu içerisindeki değerlerden birini seçeceği için seçtiği ComboBox elemanın Text içeriğine göre Case yapısından hangisi uyuyor ise o blok işletilir.

10.14.DriveListBox (Sürücü Listeleme Kutusu) Kontrolü

Sistemde bulunan sürücüler i listelemeye yarayan, ComboBox kontrolünden türemiş bir kontrol elemanıdır. Bunun vasıtası ile programın çalışması esnasında istenilen sürücü seçimi yapılır. Bu seçim sürücüye geçişi sağlamaz. Sürücüye geçiş işlemi **Chdrive** gibi bir komutla yapılması gerekir.



10.14.1.Properties

Drive

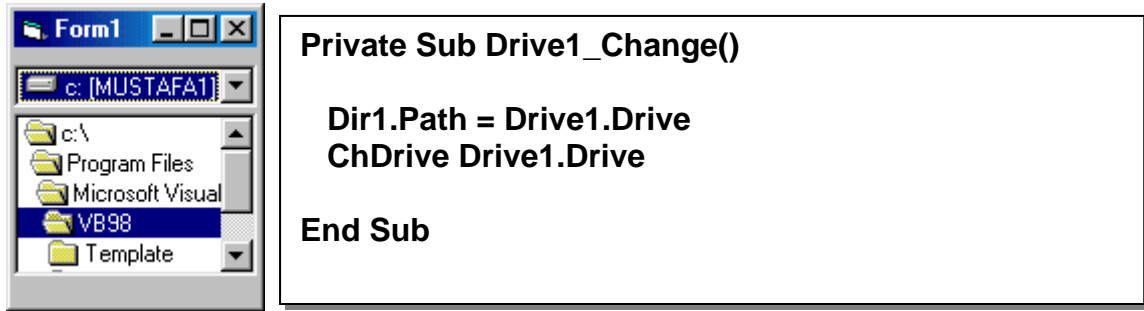
Bu özellik kullanılarak kontrolün gösterdiği aktif sürücü öğrenilebilir veya değiştirilebilir. Drive özelliğinin gösterdiği değer disket sürücüler için sürücü harfi ve ":" işareti iken, hard disk sürücüleri için ise buna sürücü etiketi de dahildir. Yani A sürücüsü A: ile gösterilirken C sürücüsü C:[Etiket] şeklinde gösterilmektedir. Bu durum genellikle dikkate alınmaz, çünkü sürücü atamaları için iki karakter dikkate alınır.

Chdrive "c:abcd" gibi bir komut C sürücüsüne geçmeyi sağlar. Aynı durum Drive değerini değiştirirken de geçerlidir. **Drive1.Drive="c:"** atanması C sürücüsüne geçmeyi sağlamaktadır.

10.14.2.Events

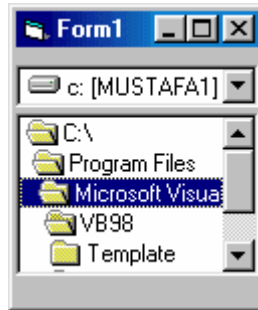
Change()

Kontrolden bir sürücü seçilmesi sırasında bu olay meydana gelir ve seçim den sonra yapılması istenen kodlar bu kısma yazılır. Bir DirectoryListBox kullanarak DriveListBox içerisindeki sürücü değiştiği zaman DirectoryListBox içerisindeki dosyalarında değişmesi için aşağıdaki gibi bir kod yazmalıyız. Bu işlemi gerçekleştirmek için aşağıdaki formu oluşturun.



10.15.DirectoryListBox (Dizin Listeleme Kutusu) Kontrolü

Sistemde bulunan sürücülerdeki dizinleri listelemeye yarar. Genellikle DriveListBox kontrolü ile birlikte kullanılır. Seçilen sürücüye ait dizin yapısını bir liste şeklinde göstermektedir. Dizin seçme işlemini ChDir gibi bir komut ile sağlar.



10.15.1.Properties

Path

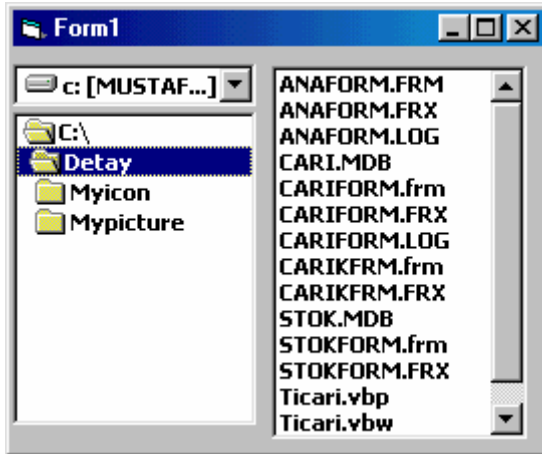
Bu özellik kullanılarak seçilmiş dizin öğrenilebilir ve değiştirilebilir. Bu özellikten dönen değer sürücü ismi dahil tam yoldur. Kontrolün göstermesi istenen dizin de kontrolün bu özelliği ile gerçekleştirilebilir. Örneğin program başlatıldığında DirectoryListBox 'ın içeriğinin "C:\Windows\Desktop" olması için Formun Load olayına aşağıdaki kodu yazmalıyım.

```
Private Sub Form_Load()  
Dir1.Path = "C:\Windows\Desktop"  
End Sub
```

10.15.2.Events

Change

Listeden bir dizinin seçilmesi durumunda bu olay gerçekleşir. Dizin seçimi yapıldığında yazılması gereken kod buraya yazılmalıdır. Bu olay gerçekleştiğinde değişiklikleri görmek için bir tane de FileListBox kontrolü kullanarak aşağıdaki gibi bir kod yazmalıyım.



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

```
Private Sub Drive1_Change()
```

```
Dir1.Path = Drive1.Drive  
ChDrive Drive1.Drive  
End Sub
```

Eğer bir dizinde yalnızca belli dosyaları görmek istersem FileListBox kontrolünün Pattern özelliğini kullanabilirim. Örneğin sadece Doc uzantılı dosyaları görmek istiyorsam yukarıdaki formun Load yordamına aşağıdaki satırı yazmam yeterli olacaktır.

```
Private Sub Form_Load()  
File1.Pattern = "*.doc"  
End Sub
```

10.16.FileListBox (Dosya Listeleme Kutusu) Kontrolü

Her hangi bir dizindeki dosyaları listelemeye yarayan ve liste kutusu kontrolünden yapılmış bir kontrol elemanıdır. Hangi dizindeki dosyaların gösterileceği Path özelliği ile belirlenir. Örneğin "C:\Windows\Desktop" içerisindeki dosyaları görmek istiyorsam formun Load olayına aşağıdaki kodu yazmalıyım.

```
Private Sub Form_Load()  
File1.Path = "C:\Windows\Desktop"  
File1.Pattern = "*.*)"   
End Sub
```

Eğer listelenecek dosyalara kısıtlama getireceksem bunu da FileListBox' ın Pattern özelliği ile sağlayabilirim.

10.16.1.Properties

FileName

Bu özellik ile seçili dosyanın ismi öğrenilebilir. Bu özelliğten dönecek değer dosya adı ve uzantısıdır.

Archive, Hidden, Normal, ReadOnly, System

Bu özellikler kullanılarak listeleme kutusunda istenen özelliğe sahip dosyalar görüntülenebilir. Bu özelliklerden herhangi birine True değeri verildiğinde o özelliğe sahip dosyalar listelenir. False verildiğinde listelenmez.

Path

Bu özellik seçeceğimiz herhangi bir dizindeki dosyalar listelenebilir. Listelenecek dosyalar bulundukları dizin ve sürücü ismi ile birlikte yazılmalıdır.

Pattern

Bu özellik ile listelenecek dosyalar seçilebilir. Bu işlem DOS işletim sisteminden tanıdığımız "? , *" karakterleri ve dosya niteliğini belirleyen dosya uzantıları sayesinde yapılır.

Örneğin Exe uzantılı dosyaları görmek için File1.Pattern="*.exe" yazmalıyım.


10.16.2.Events

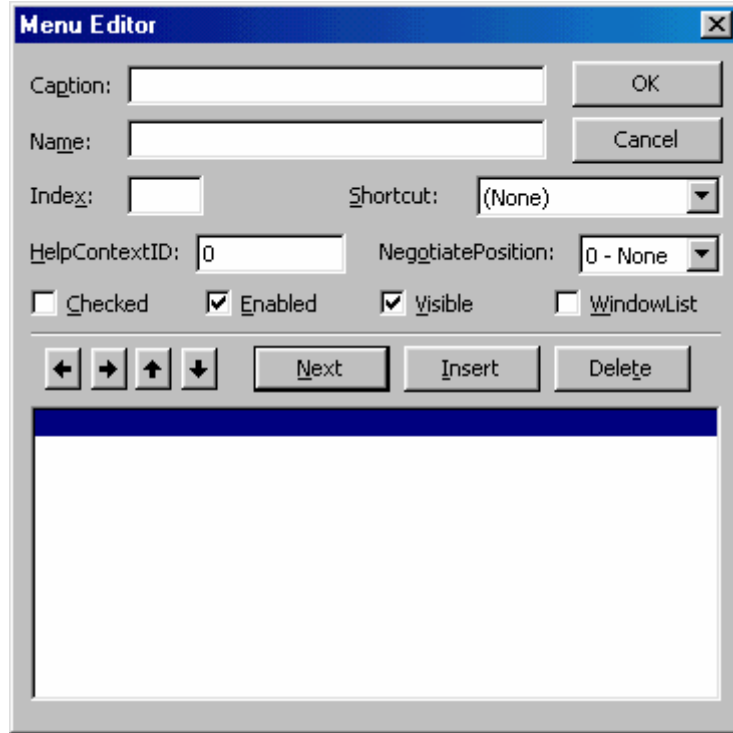
PathChange()

Bu olay Path özelliğinin değiştirilmesi sonucu meydana gelir. Yani FileListBox' ın içeriği değiştiğinde meydana gelir.

10.25.MENÜ TASARIMI

Visual Basic'te oluşturacağınız menülerin de bir kontrol gibi Properties ve Eventsleri vardır. Bu özellikleri vermeden önce menü tasarımı nasıl yapılır onu görelim.

Menü oluşturmak veya oluşturulmuş menüleri değiştirmek VB' nin Tools menüsü altındaki Menu Editor seçeneği ile yapılır. Kısaca bu seçeneğe  ikonundan veya Ctrl-E tuşları ile ulaşabilirsiniz. Bu seçenek seçildiğinde aşağıdaki gibi bir pencere açılır.



Bu pencerede bulunan alanlar şu anlama gelmektedir.

Name

Bu kutuya menünün ismi yazılır. Diğer kontrollerin Name özelliğinden bir farkı yoktur. Prensip olarak bu ismin önüne Mn gibi bir tanıtıcı harf koyarsanız bunun bir menü ismi olduğunu hatırlamanız daha kolay olur. Örneğin Kaydet yerine MnKaydet ismini vermek bu ismin bir menüye ait olduğunu hatırlamak için daha uygundur.

Caption

Menü başlığıdır. Bu kutuya yazılan yazı menü üzerinde görüntülenir. Bu aynı zamanda menünün bir propertiesidir. Çalışma zamanı da değiştirilebilir. Buraya yazılacak metnin herhangi bir harfinin başına Bt işaretinin konması o harfin altı çizili

Farklı olarak bu özelliğe — (eksi işareti) verilirse bu durumda o eleman bir menü elemanı olarak değil bir kesme çizgisi olarak gösterilir.

Aşağıdaki menüde, biri Kaydet menüsü altında, diğeri Yazdır menüsü altında ol-

mak üzere iki tane kesme çizgisi görmektesiniz. Bu kesme çizgilerini oluşturmak için Caption — olarak verilir.



Index

Bu da bir propertiesdir. Ve menü bir dizi olarak tanımlanacaksa yani aynı isimli bir kaç menü oluşturulacaksa bu kutuya index numarası verilir. Bu kutu içine herhangi bir yazılırsa menü dizi olarak tanımlanmış kabul edilir ve bu menüye ulaşım menü ismiyle birlikte parantez içinde bu numara verilerek yapılır.

Shortcut

Menüye bir kısa yol tuşu bu kutudan seçilebilir. Örneğin F2 tuşuna basıldığında kaydet menüsünün aktif hale gelmesini sağlayabilirsiniz.

WindowList

Bu özellik Form'un bir MDIChild form olması durumunda kullanılır. Bir menü elemanına bu özellik verildiğinde o anda yüklü olan MDIChild formların bir listesi bu menünün alt elemanları olacak şekilde otomatik olarak eklenir. Bu durumu Program Yöneticisinin Pencere menüsünde görebilirsiniz. (Örnek için MDI Formlar bölümüne bakınız)

Enabled, Visible

Bu iki özelliğin etkisini artık biliyorsunuz. Biri menüyü aktif-pasif yaparken diğeri de görünür veya görünmez yapar.

NegotiatePosition

Form üzerine OLE 2.0'ı destekleyen nesneler eklediğinizde, eklediğiniz uygulamanın menüleri formunuzda yerini alır. Bu özellikle, bir OLE nesnesi aktif hale geldiğinde formunuzda bulunan menülerin nasıl konumlandırılacağını belirleyebilirsiniz.




0: Bir OLE nesnesi aktif hale geldiğinde forma ait menüler gösterilmeyecek.

1: Bir OLE nesnesi aktif hale geldiğinde forma ait menüler menü çubuğunun solunda yer alacak.

2: Bir OLE nesnesi aktif hale geldiğinde forma ait menüler menü çubuğunun ortasında yer alacak.

3: Bir OLE nesnesi aktif hale geldiğinde forma ait menüler menü çubuğunun sağında yer alacak.

10.25.1.Menü tasarımı

Bir menü bir başka menünün alt menüsü olarak tanımlanacaksa pencere üzerindeki  düğmesine basılır. Bunu yaptığınızda bu menünün başına - işareti konacaktır. Bu durumda bu menü bir üstteki menünün alt menüsü olur. Bir menüyü başka bir menünün alt menüsü olmaktan çıkarmak için de  düğmesine basılır.Araya bir menü eklemek  için, Insert, silmek için de Delete düğmesini kullanabilirsiniz.

Checked


Bu özelliğe True verilerek menünün seçili (işaretli) olması sağlanabilir. Geneline Checked özelliği verilen menüler, işaretli iken seçildiklerinde işaretinin kalkmaz işaretli değilken seçildiğinde ise işaretlenmesi istenir. Bunu yapmak için o menünün Click olayında şu kod bulunmalıdır.

ÖRNEK:Örnek olarak aşağıdaki menüyü adım adım oluşturalım.



Önce Dosya menüsü için aşağıdaki değerleri girip Entere basın.

Caption:	<input type="text" value="Dosya"/>
Name:	<input type="text" value="MnDosya"/>

Aç ve Kaydet menüsünü Dosya menüsünün bir alt menüsü olarak tanımlayacağımız için önce  düğmesine basın ve kısa yol olarak F3 ve F2'yi atayın.

Caption: &Kaydet OK
 Name: MnKaydet Cancel
 Index: Shortcut: F2
 HelpContextID: 0 NegotiatePosition: 0 - None
☐ Checked ☒ Enabled ☒ Visible ☐ WindowList
 ← → ↑ ↓ Next Insert Delete
 &Dosya
&Aç F3
&Kaydet F2

Şimdi aradaki kesme çizgisini oluşturmamız gerekiyor. Bunun için menünün başlığını aşağıdaki gibi vermemiz gerekiyor.

Caption: - OK
 Name: MnKesme1 Cancel
 Index: Shortcut: (None)
 HelpContextID: 0 NegotiatePosition: 0 - None
☐ Checked ☒ Enabled ☒ Visible ☐ WindowList
 ← → ↑ ↓ Next Insert Delete
 &Dosya
&Aç F3
&Kaydet F2

Yazdır menüsünü de diğerleri gibi oluşturduktan sonra buna ait alt menüyü oluşturmak için tekrar → düğmesini kullanacağız.

.....Ekran
Yazıcıya

Yazdır menüsünün alt menüsü olmaktan çıkmak için ← düğmesine basın ve aşağıdaki kesme çizgisini de yine diğerleri gibi tanımladıktan sonra Çıkış menüsünü de tanımlayalım. Menü tasarım penceremiz aşağıdaki gibi olmalıdır.

Menu Editor
 Caption: Çıkış OK
 Name: MnÇıkış Cancel
 Index: Shortcut: F12
 HelpContextID: 0 NegotiatePosition: 0 - None
☐ Checked ☒ Enabled ☒ Visible ☐ WindowList
 ← → ↑ ↓ Next Insert Delete
 &Dosya
&Aç F3
&Kaydet F2

Yazdır
Ekran
Yazıcıya

Çıkış F12

Yukarıdaki elemanları doğru şekilde oluşturduktan sonra OK düğmesine basarsanız formunuzun üzerinde aşağıdaki menü oluşacaktır. Bu aşamadan sonra istediğiniz menüyü seçerek ona ait kodu Click olayına yazabilirsiniz.

Pencere üzerindeki diğer seçenekler birer properties olup ve aşağıda anlatılmıştır.

Checked

Bu özelliğe True verilerek menünün seçili olduğu olması sağlanabilir. Genellikle Checked özelliği verilen menüler, işaretli iken seçildiklerinde işaretinin kalkması, işaretli değilken seçildiğinde ise işaretlenmesi istenir. Bunu yapmak için o menünün Click olayında şu kod bulunmalıdır.

```
Private Sub Menu_Click ()  
Menu.Checked=Not Menu.Checked
```

Enabled

Bu özelliğe False verilerek menünün pasif olması sağlanır.

Visible

Bu özelliğe False verilerek menünün görülmez olması sağlanır. Menü nesnesinin diğer özellikleri daha önce anlatılan kontrollerdeki gibidir.

10.25.2.Events

Menünün sadece Click olayı vardır ve menü seçildiğinde bu olay aktif hale gelir.

Click

Bu olay koduna ulaşabilmek için tasarladığınız menüyü form üzerinden seçmeniz gerekir. Alt menüleri olan bir menünün ise Click olayına ulaşabilmek için Vb bir yol sunmaz. Bu kodu kendiniz yapabilirsiniz.

Alt menüleri olan bir menünün Click olayı, daha çok alt menüler açılmadan önce gerekli menüleri aktif-pasif duruma getirmek için kullanılır.

Örneğin Düzen menüsü ve altında Kes-Kopyala-Yapıştır seçenekleri varsa, Düzen menüsünün Click olay kodunu yazarak, panoda bilgi yoksa, menüler henüz açılmadan Yapıştır menüsünü pasif hale getirebilir, seçili bir kısım yoksa da Kes ve Kopyala seçeneklerini pasif hale getirebilirsiniz.

10.25.3.POPUP MENÜLER

Sağ fare tuşuna basınca ekranın herhangi bir yerinde açılan menüleri bir çok

programda görmüşsünüzdür. VB'de de tek komutla bu tip menüleri oluşturmak mümkündür.

Popup menü tasarımı aynen normal menü gibi tanımlanır. Bu menünün menü çubuğunda gösterilmesi istenmiyorsa visible özelliği istenen false yapılır. Ve açılması istenen yerde **PopupMenu**., metodu ile aktif hale getirilir.

PopupMenu [Menu_Adı] , flags x , y, bold

Menu_Adı:Menü Design kısmında tanımlanan menünün adı. Bu menünün mutlaka alt menüleri bulunmalıdır.

x,y:Menünün gösterileceği koordinatlar

flags: Flags parametresi, x koordinatının nasıl kullanılacağını ve farenin hangi tuşunun Click olayını meydana getireceğine dair iki bilgi içerir. **Flags** parametresini şöyle formülize edersek

$$\text{flags} = \text{Koorx} + \text{Fare}$$

Burada Koorx in olacağı değerler ve anlamları şöyledir.

0: vbPopupMenuLeftAlign: Menü x koordinatının solunda açılacak

4: vbPopupMenuCenterAlign: Menü x koordinatını ortalayacak.

8: vbPopupMenuRightAlign: Menü x koordinatının sağında açılacak

Fare değişkeninin alacağı değerler ise söyler.

0:vbPopupMenuLeftButton:Sol fare tuşu ile menü elemanı seçilir.

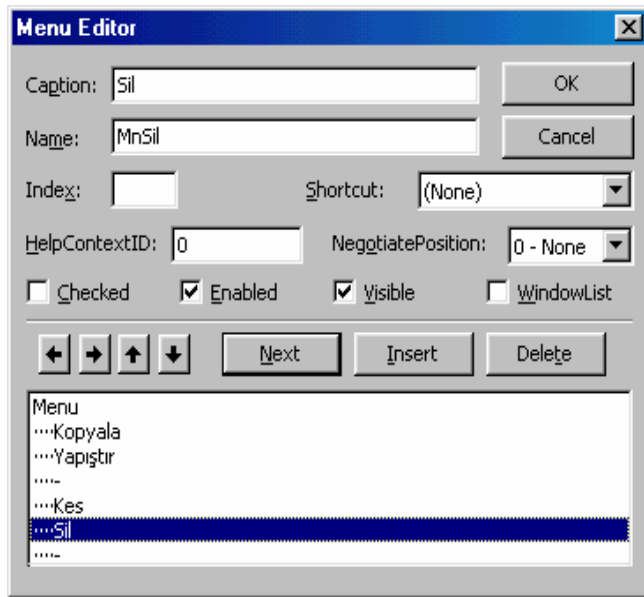
2:vbPopupMenuRightButton:Sağ fare tuşu ile menü elemanı seçilir.

Bold

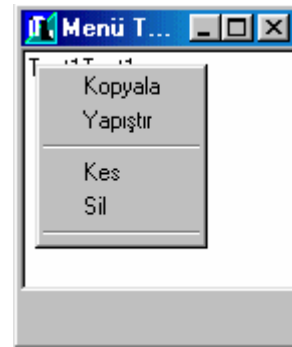
Bu parametre ile açılacak menüdeki seçeneklerden biri kalın yapılabilir. Kalın olarak yazılması istenen menünün ismi bu özelliğe verilirse popup menü açıldığında menü bold olarak yazılacaktır.

ÖRNEK:Örnek olarak bir text kutusu içinde kesme, kopyalama, silme, yapıştırma ve büyük küçük harf'e çevirme özelliklerini destekleyecek bir menü tasarlayalım.

Örneğimiz için formunuzun üzerine bir text kutusu yerleştirin ve Menü Editor penceresi yardımı ile aşağıdaki menü yapısını oluşturun. (Caption'ların önüne Mn koyarak Name özelliklerine yazın. Yani Menü Caption'lı menünün ismini MnMenu olarak verin)



Menünün Tasarım Görünümü



Program Çalıştıktan
Sonra Menü Görünümü

Örnek Program Popupmenu

```
Private Sub Form_Load()  
    MnMenu.Visible = False  
End Sub  
Private Sub MnKes_Click()  
    Clipboard.SetText (Text1.SelText)  
    Text1.SelText = ""  
End Sub  
Private Sub MnKopyala_Click()  
    Clipboard.SetText (Text1.SelText)  
End Sub  
Private Sub MnSil_Click()  
    Text1.SelText = ""  
End Sub  
Private Sub MnYapıştır_Click()  
    Text1.SelText = Clipboard.GetText()  
End Sub  
Private Sub Text1_MouseDown(Button As Integer, Shift As  
Integer, X As Single, Y As Single)  
    If Button = 2 Then  
        PopupMenu MnMenu, 4+2, Text1.Left + X, Text1.Top + Y  
    End If  
End Sub
```

10.17.Common Dialog (Dialog Penceresi) Kontrolü

Windows tarafından sağlanan standart diyalog kutularını kullanımını sağlayan kontroldür. Bu kutular “Aç”, “yeni Adla Kaydet”, “Renk”, “Help”, “Yazdır” ve “Font” dur. ToolBox penceresinden ilgili kontrol seçilir ve Formda istenilen alan seçilerek kontrol eklenir.

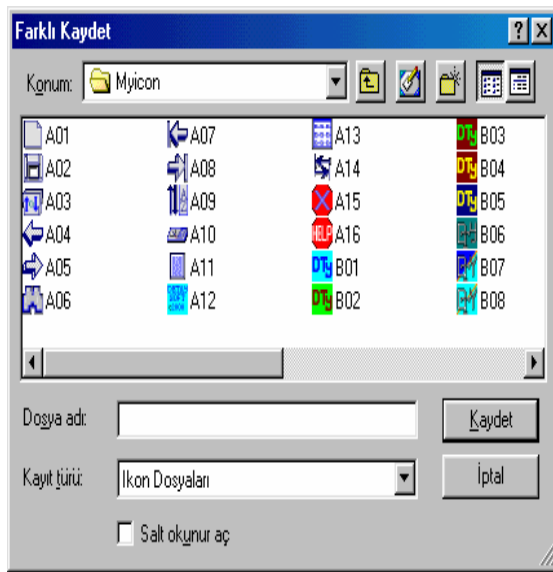
Form üzerinde kontrolleri istediğim büyüklükte ayarlayabiliyorduk. Bu kontrolde böyle bir imkan yoktur. Sabit bir büyüklüğe sahiptir. Çalışma anında Common Dialog kutusu formun üzerinde görünmez.

10.17.1.Properties

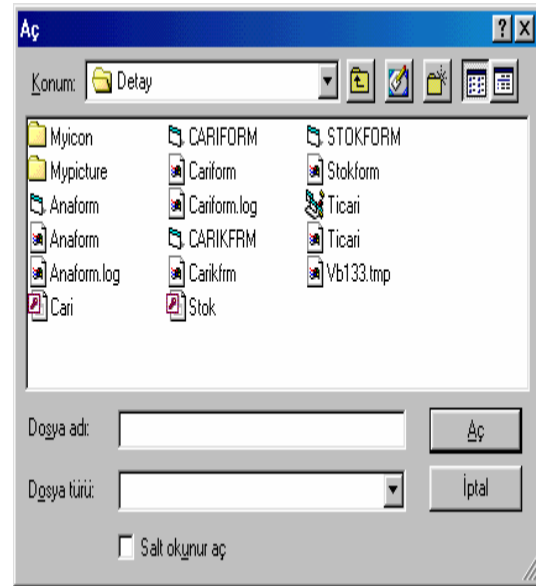
Action: 1 ile 6 arası değer verilerek ilgili diyalog kutusunun açılmasını sağlar.

1: Aç diyalog kutusunu aktif hale getirir.

CommonDialog1.Action=1 veya CommonDialog1.ShowOpen



Farklı Kaydet/Kaydet Diyalog Penceresi



Aç Diyalog Penceresi

2: Yeni Adla Kaydet diyalog kutusunu aktif hale getirir.

CommonDialog1.Action=2 veya CommonDialog1.ShowSave

3: Renk diyalog penceresini aktif yapar.

CommonDialog1.Action=3 veya CommonDialog1.ShowColor

4: Font diyalog penceresini aktif yapar.

CommonDialog1.Action=4 veya CommonDialog1.ShowFont

5: Yazdır diyalog penceresini aktif yapar.

CommonDialog1.Action=5 veya CommonDialog1.ShowPrint

6:Windows Help programını çalıştır.

CommonDialog1.Action=6 veya CommonDialog1.ShowHelp

Örneğin kullanılan 2 adet komut düğmesi ile Aç ve Kaydet diyalog pencereleri şu şekilde aktif hale getirilebilir.

```
Private Sub Command1_Click()  
CommonDialog1.Action = 1 'Aç Diyalog penceresini açar.  
End sub  
  
Private Sub Command2_Click()  
CommonDialog1.Action = 2 'Kaydet diyalog penceresini açar.  
End Sub
```

FileName

Aç ve Kaydet Diyalog pencereleri için Dosya Adı kutusunda yazılması istenen veya kullanıcı tarafından yazılan değerdir.

DefaultExt

Kullanıcı dosya isminde dosya uzantısı yazmaz ise bu özellik ile belirlenen uzantı dosya adına eklenir.

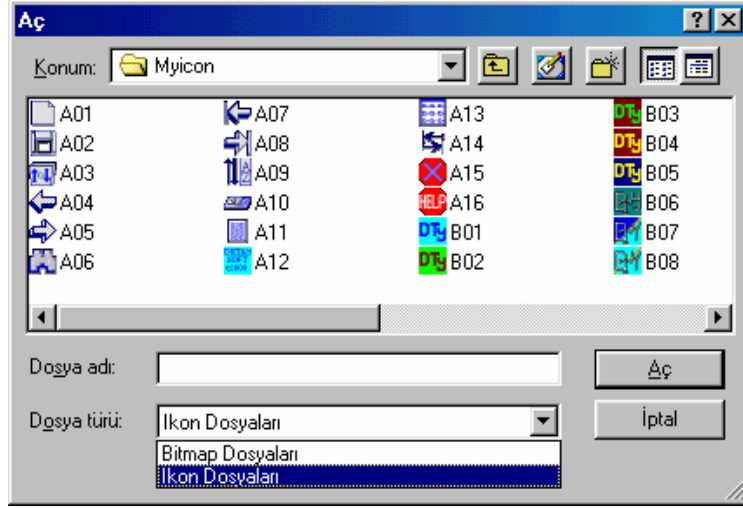
CommonDialog1.DefaultExt = "Doc"

Filter

Aç ve Kaydet diyalog pencerelerinde gördüğümüz Dosya Türü ve Kayıt Türü kutularına listelenecek dosyaların tiplerini seçmeye yararak açıklamalar yazılmasını sağlar. Kullanım formatı şöyledir.

CommonDialog1.Filter= "Açıklama1 |Filtre1|Açıklama2|Filtre2|...|....|"

Yukarıdaki komut satırı yazılıp ardından da Aç diyalog penceresi çalıştırılırsa aşağıdaki görüntüyü elde ederiz.



Buradaki | karakteri Ascii kodu 124 olan karakterdir. Klavyeden Alt Gr tuşu ile birlikte < tuşuna basılarak çıkartılabilir. AçıklamaN parametresi Dosya Türü kutusunda görülmesi istenen yazıyı, FiltreN parametresi ise listedeki elemanın seçilmesi ile dosya kutusunda gösterilecek dosyaların filtresidir.

```
CommonDialog1.Filter= "Bitmap Dosyaları|*.BMP|İkon Dosyaları|*.ico|"
```

CancelError

Dialog pencerelerinde "İptal" düğmesinin seçilmesi halinde yakalanabilir bir hata oluşturup oluşturmayacağı bu özellik ile belirlenir. Değer True ise "İptal" düğmesi seçildiğinde hata oluşturur, False ise oluşturmaz.

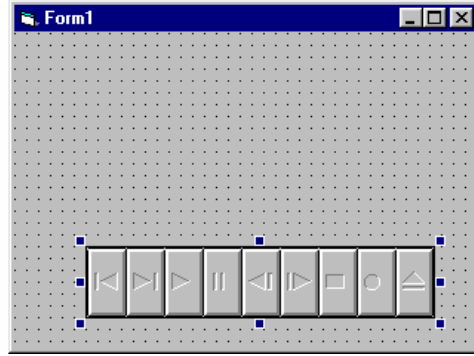
Dialog Title

Açılacak pencerenin başlığında yazılması istenen metni belirler.

10.18.MultiMedia Kontrolü

VB projeleri dahilinde Windows ile birlikte verilen Ortam yürütücüsü uygulamasından yararlanmadan **MultiMedia** kontrolünden yararlanarak gerçekleştirebiliriz. Multimedia kontrolü ToolBox penceresinde bir düğme ile temsil edilmediği için Components diyalog penceresinden eklememiz gerekmektedir.

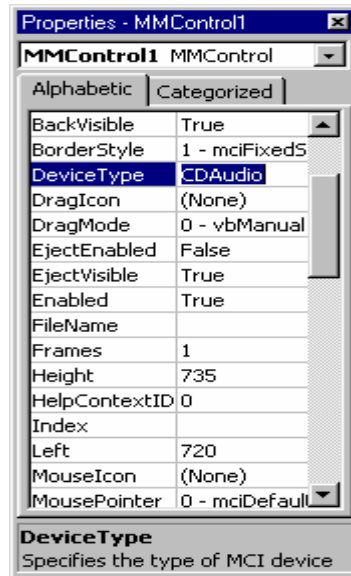
Components diyalog kutusunda Multimedia Control onay kutusunu seçip Tamam düğmesini tıklayınca Multimedia kontrolünü temsil eden ToolBox penceresine dahil edelim. Multimedia kontrolü seçiliyken mouse ile formun üzerine yerleştirelim.



MultiMedia Kontrolünün Görünüşü

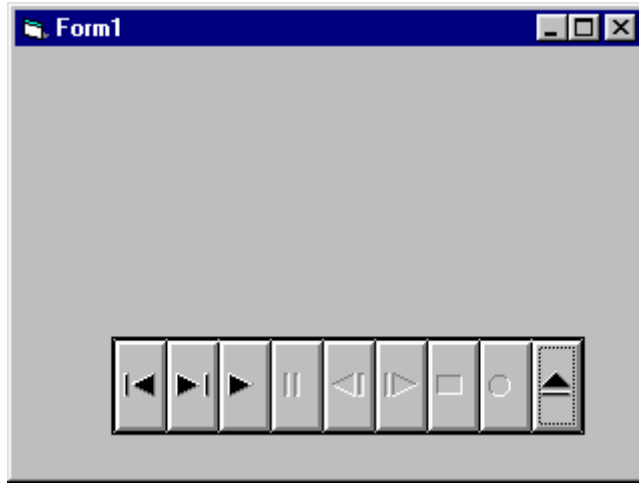
Multimedia nesnesinden hangi amaçla yararlanılacağı MultiMedia nesnesine ait Properties penceresindeki Device Type özelliği ile belirlenmektedir. Şimdi müzik CD'si çalmaya çalışalım.

Forma eklediğimiz Multimedia nesnesinden müzik CD'lerini çalmak için Command özelliğinden de yararlanmak gerekir. Command özelliğine o işlemle ilgili komutu aktarmamız gerekir. Command özelliğine Open komutunu aktaracak olursak forma eklemiş olduğumuz Multimedia nesnesi aktive edilir.



```
Private Sub Form_Load()  
    MMControl1.Command= "Open"  
End Sub
```


Bu satırı ekledikten sonra programı çalıştıralım.



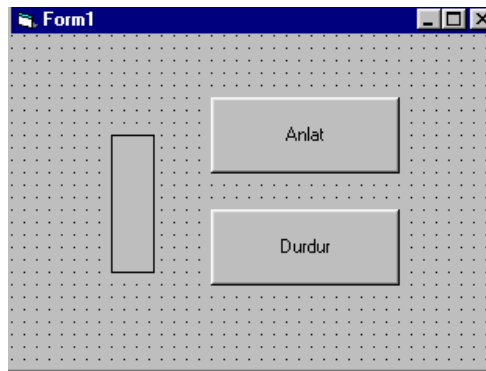
Artık müzik CD'sini çalmaya başlayabiliriz.

10.18.1.Ses Dosyalarını Seslendirmek

Programcı hazırladığı program hakkında yardım dosyası hazırlamanın yanında programını konuşarak anlatır ve WAV uzantılı bir ses dosyasına kaydedebilir.

Şimdi bu işlemin nasıl yapılacağı hakkında bilgi verelim. Üzerine çalıştığımız projenin formuna bir Multimedia kontrolü ekleyelim. Daha sonra forma yerleştirdiğimiz Multimedia kontrolünün bütün düğmelerini properties penceresinden yararlanarak formun üzerinde görünmelerini engelleyelim. Bunun için properties penceresinden BlackVisible,Enabled Visible, Next Visible, Pause Visible, Play Visible, Prev Visible, Recad Visible, Stop Visible, Eject Visible değerlerine FALSE yapalım.

Forma Anlat isminde iki tane Command Butonu atayalım. Bu buton bir bizim ses dosyamızın çalışmasını sağlayacaktır. Diğeri bu işlemi sonlandıracak.



Forma yerleřtirdiđimiz MultiMedia DeviceType özelliđine WaveAudio bilgisini aktarmak ve Formdaki düđmeye (Anlat) tıklama yapıldıđında seslendirilecek Wav uzantılı ses dosyasının adını FileName özelliđini aktarmak için ařađıdaki progman satırlarını yazalım.

```
Private Sub Command1_Click()  
    MMControl1.DeviceType = "WaveAudio"  
    MMControl1.filename= "c:\Windows\media\ctmelody.Wav"  
    MMControl1.Command = "Open"  
    MMControl1.Command = "Play"  
End Sub  
  
Private Sub Command2_Click()  
    MMControl1.Command = "close"  
End Sub
```

Artık Programımızı alıřtırabiliriz.

10.19.Animation (Film Gösterici) Kontrolü

Visual basic projeleri dahilinde Multimedia kontrolü sayesinde müzik CD'lerini, MID ve WAV uzantılı ses dosyalarını alabilir, AVI formatındaki video görüntü dosyalarını oynatabilirsiniz. Bu kontrol içerisinde ses olan Avi dosyaları ile sıkıřtırılmıř Avi dosyalarını desteklememektedir. Bu kontrolün temel amacı film içeren Avi dosyalarını göstermek deđildir. Bu kontrolün görevi ise küçük animasyonları göstermektir. Örneđin Windows iřletim sisteminde dosya kopyalarken veya silerken karřımıza ıkan animasyonun kontrolünü sađlamaktır.

10.19.1.Properties

AutoPlay

Bu özellik True yapılırsa, Open metodu ie açılan Avi dosyası otomatik olarak gösterilmeye başlar. Bu özellik False ise Open metodu ile açılan dosya daha sonra Play metodu ile başlatılmalıdır.

Center

True değeri verilirse gösterilecek Avi filmi Animation kontrolü içinde ortalanır.

BackStyle

Bu özelliđin normal değeri 0' dır ve Animation kontrolü transparan bir görünüm ile filmi oynatır. Deđeri 1 ise orijinal görünümü ile filmi oynatır.

10.19.2.Methods

Open

Animation kontrolü ile gösterilecek dosyanın açılmasını sağlar. Eğer Autoplay değeri True ise gösteri hemen başlar, değilse Play metodu ile gösteri başlar.

```
Private Sub Command1_Click ( )  
Animation1.AutoPlay=True  
Animation1.Open = "c:\basla.avi"  
End Sub
```

Play

Open metodu ile açılan dosyanın çalıştırılması için kullanılır. Play metodunu kullanmak için AutoPlay özelliğinin False olması gerekmektedir.

Stop

Gösterinin durmasını sağlamaktadır.

Close

Open metodu ile açılan dosyanın kapatılmasını sağlar.

ÖRNEK: Kullanıcının seçeceği bir Avi dosyasını çalıştıran bir program yazınız.
(Program için gerekli form aşağıda verilmiştir.)



```
Private Sub Command1_Click()  
On Local Error GoTo hata  
CommonDialog1.Filter = "Avi Dosyaları|*.Avi|"  
CommonDialog1.InitDir = "c:\windows"  
CommonDialog1.Action = 1  
Animation1.Open CommonDialog1.FileName  
Exit Sub  
hata:  
MsgBox "Bu dosya türü desteklenmiyor", vbInformation  
Exit Sub  
End Sub  
  
Private Sub Command2_Click()
```

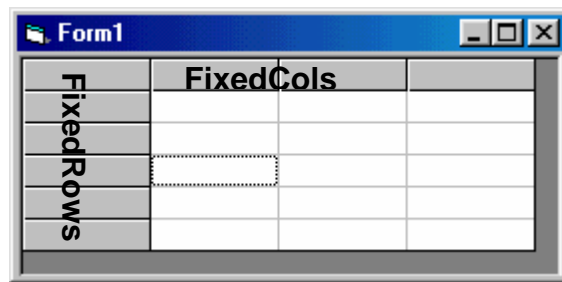
10.20.MsFlexGrid (Izgara) Kontrolü

Hücrelerden meydana gelen ve Excel sayfalarına benzeyen bir kontrol nesnesidir.

10.20.1.Properties

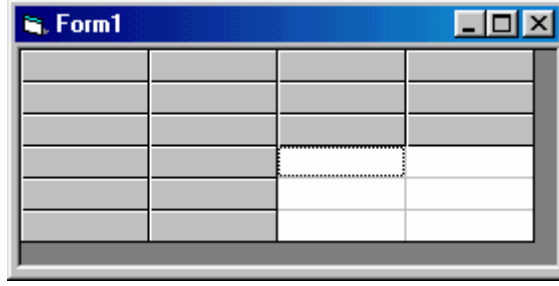
Cols, Rows

Bu özellikler sayesinde ızgara kontrolünün satır ve sütun sayısı belirlenir. Cols özelliği 4, Rows özelliği 6 olan MsFlexGrid kontrolünün görünümü aşağı verilmiştir.



FixedCols,FixedRows

Başlık olarak kullanılacak satır ve sütunlar belirlenir. Yukarda görülen şekilde her iki değeri de 1 olan bir MsFlexGrid kontrol görülmektedir. Eğer değerleri FixedCols için 2 FixedRows için 3 verecek olursam aşağıdaki gibi bir görüntü elde ederim.



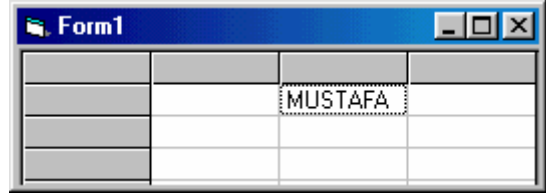
Col, Row

Bu özellikler ile aktif kolon ve satır öğrenilebilir ve değiştirilebilir. Yani üzerinde işlem yapılacak hücrenin adresini belirler.

Text

Aktif hücrenin içindeki metni temsil eder. Bu özellik ile aktif hücrenin içeriği öğrenilebilir veya değiştirilebilir. Örneğin kontrolün 1.satır 2. kolonuna adımızı yazdırmak için aşağıdaki kodu kullanmalıyım.

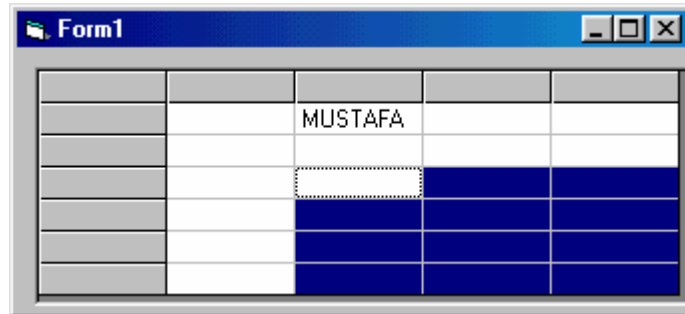
```
Private Sub Form_Load()  
MSFlexGrid1.Row = 1  
MSFlexGrid1.Col = 2  
MSFlexGrid1.Text = "MUSTAFA"  
End Sub
```



Programın ekran çıktısı

RowSel, ColSel

Grid içerisinde kullanıcı birden fazla hücre seçebilir. Seçili alanı öğrenmek veya değiştirmek için bu özellikler kullanılır. Seçili satırlar Row ve RowSel, sütunlar ise Col ve ColSel özellikleri kullanılarak belirlenir. Örneğin aşağıdaki gibi bir görünüme sahip Grid kontrol varsa ;



Row=3, RowSel=6 ve Col=2, ColSel=4 şeklinde değere sahip demektir.

Clip

Bir ızgara içindeki seçilmiş alının içeriğini verir. Bir çok hücreye birden atama yapılabilir. Hücrelerdeki bilgileri bir veri tabanı dosyasına veya veri tabanı dosyasında bulunan bilgileri de hücrelere yazabilir.

ÖRNEK:Örneğin aşağıdaki ders programı için Grid' in ilk iki satırına şu dersleri yerleştirmek istiyor ve aşağıdaki formu elde etmek istiyorum.

"VB 6.0", "Dbase", "Mikro İşlem", "Pascal"

"VB 6.0", "Mikro İşlem", "Dbase", "Pascal"

	Pazartesi	Salı	Çarşamba	Perşembe
1.Ders	VB 6.0	Dbase	Mikro İşlem	Pascal
2.Ders	VB 6.0	Mikro İşlem	Dbase	Pascal
3.Ders				
4.Ders				

Bu görüntüyü elde etmek için eğer iki yol vardır. Birincisi hücreleri tek tek Row ve Col özellikleri ile teker teker işaretleyip Text özelliğini kullanarak hücreleri doldurmak diğer yol ise Clip kontrolü ile gerekli hücelere bilgilerin yazılmasıdır. Clip kontrolü kullanılırken aynı satırda bulunan bir hücreden diğer bir hücreye geçerken Chr(9) karakteri, bir sonraki sütuna geçiş için de Chr(13) karakteri kullanılır. Şimdi yukarıda görülen ders programını her iki yolu kullanarak oluşturalım.

```
Private Sub Form_Load() 'Birinci yöntem ile ders programı yapımı.
```

```
MSFlexGrid1.Row = 1
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = "1.Ders"
MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = "2.Ders"
MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = "3.Ders"
MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = "4.Ders"
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 1
MSFlexGrid1.Text = "Pazartesi"
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 2
MSFlexGrid1.Text = "Salı"
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = "Çarşamba"
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 4
```

Private Sub Form_Load() 'ikinci yol ile ders programın elde edilmesi.

```
MSFlexGrid1.Row = 1
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = "1.Ders"
MSFlexGrid1.Row = 2
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = "2.Ders"
MSFlexGrid1.Row = 3
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = "3.Ders"
MSFlexGrid1.Row = 4
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = "4.Ders"
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 1
MSFlexGrid1.Text = "Pazartesi"
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 2
MSFlexGrid1.Text = "Salı"
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = "Çarşamba"
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 4
```

Clip özelliğinin asıl önemi ise Grid içerisindeki bilgileri bir dosyaya yazmak veya bir dosya içerisindeki bilgileri hücrelere yazmak olduğunu söylemiştik. Bu işlem için aşağıdaki kodu yazmak yeterlidir.

```
Private Sub Form_Load()  
Dim i,s  
Open "C:\takip.dat" For Input As #1 'Dosya açılır.  
MsFlexGrid1.Col=1: MsFlexGrid1.Row=1  
MsFlexGrid1.ColSel = MsFlexGrid1.ColSel-1  
MsFlexGrid1.RowSel= MsFlexGrid1.RowSel-1  
Input # 1,s  
MsFlexGrid1.Clip=s  
Close #1  
End Sub
```

Dosyaya bilgileri yazmak için ise aşağıdaki kod yeterlidir.

```
Private Sub Form_Load()  
Dim i,s  
Open "C:\takip.dat" For Output As #1 'Dosya açılır.  
MsFlexGrid1.Col=1: MsFlexGrid1.Row=1  
MsFlexGrid1.ColSel = MsFlexGrid1.ColSel-1  
MsFlexGrid1.RowSel= MsFlexGrid1.RowSel-1  
S=MsFlexGrid1.Clip  
Write # 1,s  
Close #1  
End Sub
```

TextMatrix(Satır,Sütun)

Grid içindeki bilgileri öğrenmenin veya değiştirmenin bir yolu da budur. Diğer yollara göre avantajı ise hem hücrenin adresinin hem de değerinin tek seferde değiştirmesine olanak sağlamasıdır.

Yukarıdaki ders programını TextMatrix özelliği ile gerçekleştirebilirsek.

```
Private Sub Form_Load()  
MsFlexGrid1.TextMatrix(1,1)= "VB 6.0"  
MsFlexGrid1.TextMatrix(1,2)= "Mikro İşlem"  
MsFlexGrid1.TextMatrix(1,3)= "Dbase"  
MsFlexGrid1 TextMatrix(1,4)= "Pascal"
```


Sort

Bu özelliğe içerdiği değerlerden biri verilerek Grid üzerinde sıralama yaptırılabilir. Eğer seçili bir alan varsa sıralama yalnızca seçili alanda yapılır. Seçili bir alan yoksa sıralama Grid kontrolünün tamamında yapılır. Sor özelliğinin alabileceği değerler aşağıda verilmiştir.

- 0 : Sıralama yok anlamındadır.
 - 1 : Artan sıralama anlamındadır.
 - 2 : Azalan sıralama
 - 3 : Artan sayısal sıralama
 - 4 : Azalan sayısal sıralama
 - 5 : Büyük – Küçük harf ayrımı yapmayan, artan sıralama yapar.
 - 6 : Büyük – Küçük harf ayrımı yapmayan, azalan sıralama yapar.
 - 7 : Büyük – Küçük harf ayrımı yapan, artan sıralama yapar.
 - 8 : Büyük – Küçük harf ayrımı yapan, azalan sıralama yapar.
 - 9 : Özel sıralama yapma imkanı sağlar. Sıralama için Compore olayı kullanılır.
- Şimdi bu açıklamaları daha iyi anlamak için bir örnek yapalım. Öncelikle aşağıdaki form görünümünü tasarlayalım.

İşçinin Adı	Aldığı Ücret

İŞÇİNİN ADI

İŞÇİNİN ÜCRETİ

☐ ARTAN ☐ AZALAN

Option Explicit
Dim i As Integer

```
Private Sub Form_Load()  
MSFlexGrid1.TextMatrix(0, 0) = "İşçinin Adı"  
MSFlexGrid1.TextMatrix(0, 1) = "Aldığı Ücret"  
MSFlexGrid1.ColWidth(0) = 250 * 8  
MSFlexGrid1.ColWidth(1) = 250 * 8
```

```
Private Sub Command3_Click()    'MAAŞA GÖRE düğmesi  
MSFlexGrid1.Col = 1  
If Option1.Value = True Then MSFlexGrid1.Sort = 3  
If Option2.Value = True Then MSFlexGrid1.Sort = 4  
If Option1.Value = False And Option2.Value = False Then  
MsgBox "Bir sıralama türü seçiniz", vbInformation :Exit Sub  
End If  
End Sub
```

Programı çalıştırıp değerleri girerek sıralama işlemlerini görmüş oluruz. Örneğin benim girdiğim veriler için maaş bilgilerine göre azalan sıralama sonucu aşağıda görüldüğü gibi oldu.

İşçinin Adı	Aldığı Ücret
MURAT ÖZTÜRK	300.000.000
MEVLÜT BUDAK	250.000.000
FUAT ERDOĞAN	250.000.000
YOLKAN AKBAŞ	200.000.000
MUSTAFA DURAK	200.000.000

İŞÇİNİN ADI

İŞÇİNİN ÜCRETİ

☐ ARTAN ☒ AZALAN

10.20.2.Methods

AddItem

Liste kutusunda gördüğümüz bu metot MsFlexGrid' e yeni bir satır eklemek için kullanılır. Eklenen satır normalde Grid' in sonunda yer alır. Eğer araya eklenmek istenirse parametre olarak ekleneceği yerin satır numarası verilir.

MsFlexGrid1.AddItem “ ” gibi kullanılırsa Grid' in sonuna boş bir satır eklenir.

MsFlexGrid1.AddItem “”,1 gibi kullanılırsa Grid' in başına boş bir satır eklenir. Bu şekilde kullanırken dikkat edeceğimiz husus ise FixedRow özelliğinin değeridir. Eğer FixedRow değeri 2 ise parametre olarak vereceğimiz satırın eklenme adresi en az 2 olmalıdır.

AddItem metodu ile eklenen satırlara ait hücreler yine aynı metot sayesinde doldurulabilir. Bunun için satır bilgileri arasına Chr(9) konularak diğer hücreye geçiş sağlanır ve hücre bilgisi çift tırnak arasına yazılır.

MsFlexGrid1.AddItem “Mustafa(1.Hücre)” +Chr(9)+”Durak(2. Hücre)”
RemoveItem

Aradan bir satırın silinmesi için kullanılır. Index numarası verilen satırı siler ve diğer satırları yukarı doğru kaydırır. Kullanırken yine FixedRow özelliği gös önünde bulundurulmalıdır. Çünkü FixedRow satırları silinemez.

MsFlexGrid1.RemoveItem 1 ‘1 nolu satırı siler.

Clear

Grid' in içindeki bütün bilgileri siler ancak satır ve sütunları silmez. Yani hücre içeriklerini siler.

10.20.3.Events

RowColChange()

Aktif olan hücre kontrolünün başka hücreye geçmesi halinde yani o hücrenin aktif olması halinde meydana gelen olaydır.

SelChange()

Seçili alanın değişmesi halinde bu olay meydana gelir.

EnterCell(), LeaveCell()

Bir hücre aktif hale getirildiğinde EnterCell, aktiviteyi kaybettiğinde ise LeaveCell olayı meydana gelir.

Compare (ByVal Row1 As Long, ByVal Row2 As Long, Cmp As Integer)

Sort özelliği ile sütun veya satırdaki verileri sıralayabiliyorduk. Ancak bazı bilgiler vardır ki bu azalan veya artan sıralama metodu ile de sıralanamazlar. Özellikle Hiyerarşik sıralama türlerinde bu sıralamalar yerine Compare olayına yazdığımız kod ile sıralama işlemini gerçekleştirebiliriz.

Bu işlem için Sort özelliğine 9 değeri verilir. Daha sonra satırlar karşılaştırılmaya başlanır. Hangi satırların karşılaştırıldığı Row1 ve Row2 parametreleri ile programa bildirilir. Yazılacak kod ile bu iki satırın hangisinin büyük, eşit veya küçük olduğu belirlendikten sonra Cmp parametresi ile sonucu bildirmeniz gerekir.

Cmp parametresi 1 (Row1 büyük), 0 (Eşit), -1 (Row2 büyük) değerlerini alabilir. Karşılaştırma aktif kolon üzerinde yapılacağı için karşılaştırılacak bilgiler şu şekilde alınabilir.

```
X1=MsFlexGrid1.TextMatrix(Row1, MsFlexGrid1.Col)  
X2=MsFlexGrid1.TextMatrix(Row2, MsFlexGrid1.Col)
```

Karşılaştırma işlemi bittikten sonra sonuç Cmp değeri ile bildirilir.
ÖRNEK: Öğretmen ile Müdür karşılaştırılması için aşağıdaki kodlar kullanılır.

```
X1=MsFlexGrid1.TextMatrix(Row1, MsFlexGrid1.Col)  
X2=MsFlexGrid1.TextMatrix(Row2, MsFlexGrid1.Col)  
If X1= "Müdür" And X2= "Müdür" Then Cmp=0  
If X1= "Müdür" And X2= "Öğretmen" Then Cmp=1  
If X1= "Öğretmen" And X2= "Müdür" Then Cmp=-1  
If X1= "Öğretmen" And X2= "Öğretmen" Then Cmp=0
```

ÖRNEK: Örnek olarak iki kare matrisin çarpımını gerçekleştiren programı tasarlayalım. (Program için (3x3) boyutunda matris düşünüldü ve A, B, C isimli 3 adet MsFlexGrid kullanıldı.) Tasarım için aşağıdaki form oluşturun.

Bu form üzerinde;

A MATRİSİNİ GİR butonu Command1

B MATRİSİNİ GİR butonu Command2

AxB İŞLEMİNİ YAP butonu Command3

MATRİSLERİ TEMİZLE butonu Command4

ÇIKIŞ butonu ise Command5' tir.

```
Private Sub Form_Load( )
```

```
A.ColWidth(1) = 10 * 80:A.ColWidth(2) = 10 * 80:A.ColWidth(3) = 10 * 80
A.TextMatrix(0, 0) = "A MATRİSİ" ' A MsFlexGrid biçimlendirildi.
```

```
B.ColWidth(1) = 10 * 80:B.ColWidth(2) = 10 * 80:B.ColWidth(3) = 10 * 80
B.TextMatrix(0, 0) = "B MATRİSİ" ' B MsFlexGrid biçimlendirildi.
```

```
C.ColWidth(1) = 10 * 80:C.ColWidth(2) = 10 * 80:C.ColWidth(3) = 10 * 80
C.TextMatrix(0, 0) = "AxB MATRİSİ" ' C MsFlexGrid biçimlendirildi.
```

```
A.ColWidth(0) = 13 * 80
```

```
B.ColWidth(0) = 13 * 80
```

```
C.ColWidth(0) = 15 * 80
```

```
End Sub
```

```
Option Explicit
```

```
Dim i, j, k As Integer
```

```
Private Sub Command1_Click( )
```

```
For i = 1 To 3
```

```
For j = 1 To 3
```

```
A.TextMatrix(i, j) = InputBox("A(" + Str(i) + ", " + Str(j) + ").eleman=")
```

```
If A.TextMatrix(i, j) = "" Then A.TextMatrix(i, j) = 0
```

```
Else
```

```
End If
```

```
Next: Next
```

```
End Sub
```

```
Private Sub Command2_Click()
```

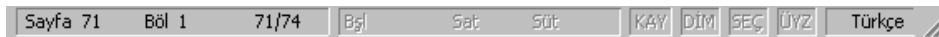
```
For i = 1 To 3
```

```
For j = 1 To 3
```

```
B.TextMatrix(i, j) = InputBox("B(" + Str(i) + ", " + Str(j) + ").eleman=")
```

10.21. Status Bar (Durum Çubuğu)

Bir çok programda pencerelerin en altında o anki durumu gösteren bazı bilgiler bulunur. Örneğin Word programının en alt satırında bazı bilgiler veren bir durum çubuğu bulunur.



Bu tip durum çubuklarını StatusBar kontrolünü kullanarak kolayca oluşturabilirsiniz. StatusBar kontrolü forma ilk yerleştirildiğinde bir adet panele(kanala) sahiptir.

10.21.1. Properties

Style

0, sbrNormal: Kontrol üzerinde birden fazla panel bulunabilir.

1, sbrSimple: Kontrol üzerinde sadece bir panel bulunabilir.

Panels

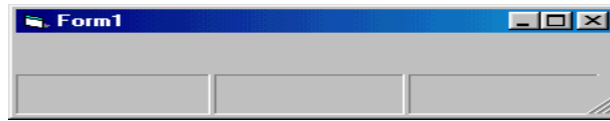
Kontrol üzerindeki paneller Panels özelliğinin aşağıdaki alt özellikleri ile yönetilirler.

Panels.Add

Kontrolle yeni paneller ekler. Eklenen bu panellere Panels(index) özelliği ile ulaşılır.

```
Private Sub Form_Load( )  
    StatusBar1.Panels.Add  
    StatusBar1.Panels.Add  
End Sub
```

Yukarıda yazılan satırlar ile kontrole iki adet panel daha eklenmiş olur.



Panels.Remove (index)

Index numaralı paneli siler. Panel numaraları birden başlar.

```
StatusBar1.Panels.Remove (1) 'Bir numaralı panel silinir.
```

Panels.Clear

StatusBar kontrolü üzerindeki bütün panelleri siler.

Panels.Count

StatusBar kontrolü üzerindeki toplam panel sayısını verir.

Panels(i).Style

Panellerin içeriklerini belirler. Panel içerikleri şunlardır.

0,sbrText: Yazı veya resim içerebilen bir panel

1,sbrCaps: Caps Lock tuşunu temsil eden bir panel

2,sbrNum: Num Lock tuşunu temsil eden bir panel

3,sbrIns: Insert tuşunu temsil eden bir panel

- 4,sbrScrl:** Scroll Lock tuşunu temsil eden bir panel
5,sbrTime: Saati gösteren bir panel
6,sbrDate: Tarihi gösteren bir panel

Eğer bunların hepsi aşağıdaki kod ile kontrole uygulanırsa;

```
Private Sub Form_Load ( )  
    Dim i  
    For i=1 to 6  
        StatusBar1.Panels.Add  
        StatusBar1.Panels(i).Style=i  
    Next i  
End Sub
```



Görüntüsünü elde ederiz.

Panels(index).Aligment

Index numarası verilen panelin içeriğinin yerleşimini ayarlar.

- 0,sbrLeft :** Sola dayalı
1,sbrCenter: Ortada
2,sbrRight: Sağa dayalı

Panels(index).AutoSize

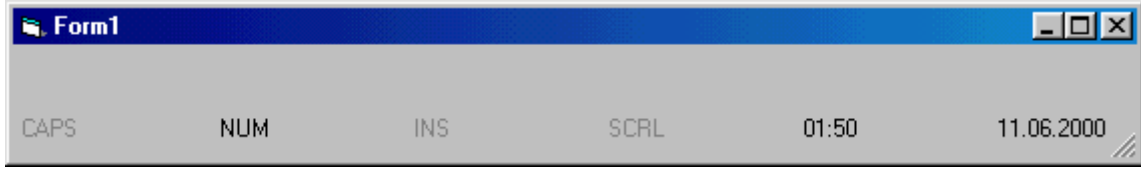
Index numarası verilen panelin boyutlandırılmasını sağlar.

- 0,sbrNoAutoSize:** Herhangi bir boyutlandırma işlemi uygulanmaz.
1,sbrSpring:Form boyutlandırıldığında, kontrolün kullanabileceğinden fazla yer açılıyor ise her panel bu fazlalığı paylaşır.
2,sbrContents: Her panel gerekiyor ise içeriğini tam gösterecek şekilde genişler.

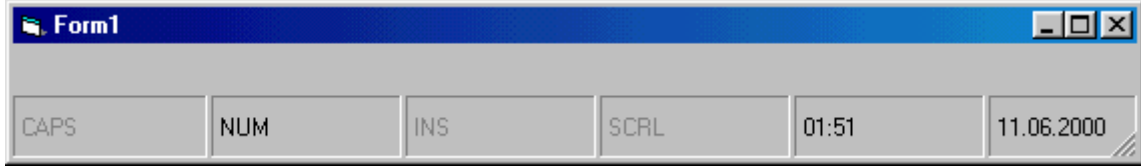
Panels(index).Bevel

Panellerin üç boyutlu olarak görünmelerini sağlar.

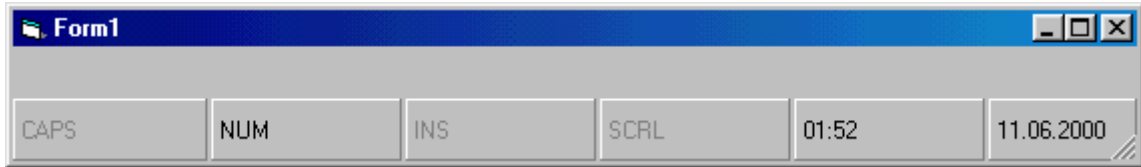
- 0,sbrNoBevel:** Üç boyut efekti uygulanmaz düz gösterilir.



1,sbrInset: İçe basık bir üç boyut efekti uygular.



2,sbrRaised: dışı basık bir üç boyut efekti uygular.



Panels(index).Visible

Index numaralı paneli gizlenmesini sağlar.

Panels(index).Enabled

Caps, Num, Ins ve Scrl tuşlarının durumlarını gösteren panellerde o tuş aktif ise Enabled özelliği True, pasif ise False olur. Bu özelliğe bakılarak o tuşların durumu öğrenilebilir.

Panels(index).Key

Paneller açıklayıcı bir isim vermek için kullanılır.

Panels(index).MinWidth, Panels(index).MaxWidth, Panels(index).Width

Bu özellikler ile her panellerin minimum, normal, maksimum boyutlar belirlenir.

Panels(index).Text

Panel üzerine yazılacak olan metin bu özellik ile belirlenir.

ÖRNEK: Form üzerinde bulunan OptionButton düğmelerinden hangisi seçilirse o butonun numarasına ait panelde butonun Caption' ı görülsün. Örnek için aşağıdaki formu tasarladım.

```
Private Sub Form_Load()  
Dim i  
For i = 1 To 3  
StatusBar1.Panels.Add  
Next i  
End Sub  
Private Sub Option1_Click()  
StatusBar1.Panels(1).Text = Option1.Caption  
StatusBar1.Panels(2).Text = "Seçili Değil"  
StatusBar1.Panels(3).Text = "Seçili Değil"  
StatusBar1.Panels(4).Text = "Seçili Değil"  
End Sub  
Private Sub Option2_Click()  
StatusBar1.Panels(2).Text = Option2.Caption  
StatusBar1.Panels(1).Text = "Seçili Değil"  
StatusBar1.Panels(3).Text = "Seçili Değil"  
StatusBar1.Panels(4).Text = "Seçili Değil"  
End Sub  
Private Sub Option3_Click()  
StatusBar1.Panels(3).Text = Option3.Caption  
StatusBar1.Panels(1).Text = "Seçili Değil"  
StatusBar1.Panels(2).Text = "Seçili Değil"  
StatusBar1.Panels(4).Text = "Seçili Değil"  
End Sub  
Private Sub Option4_Click()  
StatusBar1.Panels(4).Text = Option4.Caption  
StatusBar1.Panels(1).Text = "Seçili Değil"  
StatusBar1.Panels(2).Text = "Seçili Değil"  
StatusBar1.Panels(3).Text = "Seçili Değil"  
End Sub
```

Eğer kodu yazar ve programı çalıştırırsanız aşağıdaki gibi bir çıktı alırsınız.

Panels(index).Picture

Panellere metnin yanı sıra BMP, ICO ve WMF türündeki resimlerde yüklenebilir. Bu resimler tasarım anında açılan Custom özelliği ile açılan aşağıdaki pencereden yüklenebileceği gibi çalışma anında da LoadPicture komutu ile yüklenebilir.

StatusBar' ın Özellikler Penceresi

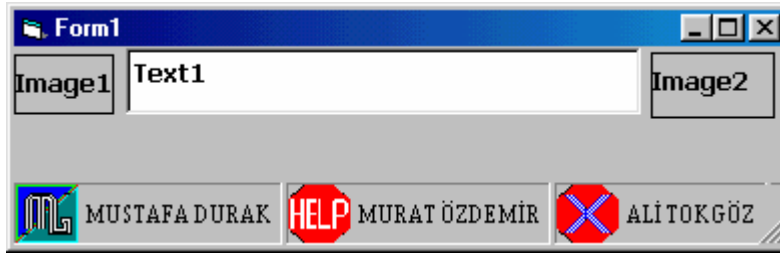
Resim eklendikten sonra StatusBar aşağıdaki gibi görülür.

10.21.2. Events

PanelClick(ByVal Panel As ComctlLib.Panel)

Panellerden birinin tıklanması sonucu bu olay meydana gelir. Buradaki Panel parametresi ile ilgili panelin bütün özellikleri öğrenilebilir ve değiştirilebilir.

ÖRNEK: Üç panelli ve resimli bir StatusBar'ın hangi paneline fare ile tıklanırsa o panelin yazısını Text1'e resmini ise Image1 ve Image2 nolu resim kontrolüne aktaran bir program yapalım. Program için aşağıdaki formu hazırlayalım.



Programın çalışması için ise aşağıdaki kod yazılmalıdır.

```
Private Sub StatusBar1_PanelClick(ByVal Panel As MSComctlLib.Panel)
    Select Case Panel
    Case StatusBar1.Panels(1)
        Text1.Text = StatusBar1.Panels(1).Text
        Image1.Picture = StatusBar1.Panels(1).Picture
        Image2.Picture = Image1.Picture
    Case StatusBar1.Panels(2)
        Text1.Text = StatusBar1.Panels(2).Text
        Image1.Picture = StatusBar1.Panels(2).Picture
        Image2.Picture = Image1.Picture
    Case StatusBar1.Panels(3)
        Text1.Text = StatusBar1.Panels(3).Text
        Image1.Picture = StatusBar1.Panels(3).Picture
        Image2.Picture = Image1.Picture
    End Select
End Sub
```

PanelDblick(ByVal Panel As ComctlLib.Panel)

Panellerden bir çift tıklandığı zaman bu olay meydana gelir. PanelClick olayı ile mantığı aynıdır.

ÖRNEK: Bir önceki örnekteki formu kullanarak bir panel üzerine çift tıklayınca Text1'in içeriğini ve Image1 ile Image2 kontrollerinde bulunan resimleri çift tıklanan panele aktaran bir program yapalım.(Örnek için aynı formu kullanacağız ama PanelClick için yazdığımız kodu sileceğiz.)

Private Sub Form_Load()

Text1.Text = StatusBar1.Panels(1).Text 'ilk anda Text kutusu doldurulur.

Image1.Picture = StatusBar1.Panels(1).Picture

Image2.Picture = Image1.Picture

End Sub

10.22. ProgressBar (İlerleme Çubuğu)

Windows altında bir çok programda gördüğümüz, yapılan işlemlerin ilerleme durumunu gösteren kontrollerdendir. Bu kontrolün Min ve Max özellikleri ile aralık belirlendikten sonra Value özelliği ile yapılan işlemin o anki değeri belirlenir. Özellikle uzun süren kodlarımıza bir PProgressBar ekleyerek kullanıcının sıkılmasını engelleyebiliriz.

10.22.1.Properties

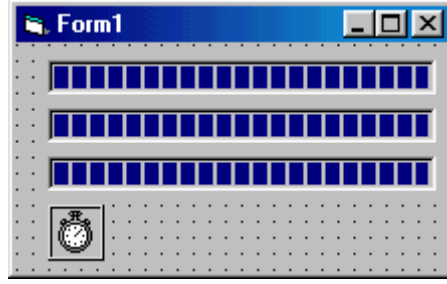
Min, Max

ProgressBar' ın alacağı maksimum ve minimum değerler belirlenir. Minimum değer olarak en az 0, maksimum değer olarak ise 10^{38} e kadar olan bir sayı verilebilir.

Value

ProgressBar' ın göstereceği değer bu özellik ile belirlenir. Yani ProgressBar' ın artım miktarını ayarlar.

ÖRNEK: Her saniye değiştiğinde dakikayı, her dakika değiştiğinde, saati değiştiren (saniye için ProgressBar1, dakika için ProgressBar2, saat için ProgressBar3 kullanıldı.) bir program yapalım. Program için aşağıdaki formu oluşturun.



Programın çalışması için ise aşağıdaki kodu programa ekleyin.

```
Private Sub Form_Load()  
ProgressBar1.Value = Hour(Now) 'Şimdiki saat  
ProgressBar2.Value = Minute(Now) 'Şimdiki dakika  
ProgressBar3.Value = Second(Now) 'Şimdiki saniye  
End Sub  
Private Sub Timer1_Timer()  
ProgressBar3.Value = ProgressBar3.Value + 1  
If ProgressBar3.Value = 60 Then  
ProgressBar2.Value = ProgressBar2.Value + 1  
ProgressBar3.Value = 0  
Else  
ProgressBar2.Value = ProgressBar2.Value  
End If  
If ProgressBar2.Value = 60 Then  
ProgressBar3.Value = 0  
ProgressBar1.Value = ProgressBar1.Value + 1  
Else  
ProgressBar1.Value = ProgressBar1.Value  
End If  
If ProgressBar1.Value = 24 Then  
ProgressBar2.Value = 0  
ProgressBar1.Value = 0  
Else  
ProgressBar2.Value = ProgressBar2.Value  
ProgressBar1.Value = ProgressBar1.Value  
End If  
End Sub
```

10.23.UpDown (Aşağı Yukarı) Kontrolü

VB' nin önceki versiyonlarında olamayan bir değerin artırılıp azaltılmasını sağlayan ve başka kontrollerle bağlantı kurarak onun içindeki değeri değiştirebilen bir kontroldür.

10.23.1.Properties

Orientation

Bu kontrol hem yatay hem de dikey olarak yerleştirilebilir. Bu özelliğin değeri 0 ise kontrol yatay, 1 ise kontrol dikey olarak görüntülenir.

BuddyControl

Bu kontrol aracılığı ile programımız içindeki başka bir kontrolün içeriğini değiştirebilirsiniz. Örneğin programınızdaki bir Text kutusu veya Label içindeki sayıyı bu kontrol aracılığı ile değiştirmek isterseniz bu özelliğe o kontrolün ismini vermeniz gerekir.

```
UpDown1.BuddyControl=Text1
```

BuddyProperty

BuddyControl özelliği ile UpDown kontrolünün kullanacağı kontrolü belirledikten sonra bu özellik ile de o kontrolün hangi özelliğinin değiştirileceğini belirlemeniz gerekir. Text1 kutusunun içindeki sayısı UpDown1 ile yüksekliğini de UpDown2 ile değiştirmek için;

```
UpDown1.BuddyControl=Text1  
UpDown1.BuddyProperty="Text"  
UpDown2.BuddyControl=Text1  
UpDown2.BuddyProperty="Height"
```

Aligment

BuddyControl ile bir kontrole bağlanan UpDown düğmeleri o kontrolün sağına veya soluna yerleştirilebilir. Aligment özelliğinin değeri 0 ise kontrolün soluna, 1 ise sağına yerleşecektir.

Min, Max

UpDown kontrolünün değiştirebileceği değer aralığı bu iki özellik ile belirlenir. Normalde bu değer 0-10 arasındır ve 0 ile 10 arasında değer ayarlanabilir. Bu iki özellik değiştirilerek aralık yeniden ayarlanabilir.

Increment

UpDown kontrolünün her seferinde ne kadarlık bir değişim gerçekleştireceği bu özellik ile belirlenir. Normalde bu değer 1' dir ve artış ve azalış miktarı da 1' dir.

Value

UpDown kontrolünün temsil ettiği değer bu özellik ile öğrenilebilir.

10.23.2.Events

UpClick

Kontrol üzerindeki artırma düğmesi tıklandığında bu olay meydana gelir.

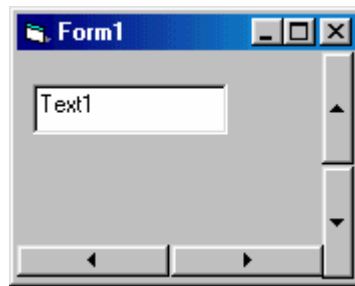
DownClick

Kontrol üzerindeki azaltma düğmesi tıklandığında bu olay meydana gelir.

Change

UpDown kontrolünün temsil edildiği değer değiştiğinde bu olay meydana gelir. Buddy özellikleri ile bu kontrol ile başka bir kontrol arasında bağlantı varsa herhangi bir kod yazmak gerekmez. Ancak bir bağlantı yok ise bu olaylar kullanılarak değer değişimlerinde yapılacak işlemler için gerekli kodlar bu olaylara yazılmalıdır.

ÖRNEK: Bir Text kutusunu form üzerinde sağa, sola, yukarı, aşağı hareket ettiren bir programı hem Buddy özelliği hem de olay alt yordamlarını kullanarak yazalım. Program için aşağıdaki gibi bir form oluşturalım ve kodları yazalım.



Private Sub Form_Load()

UpDown2.BuddyControl = Text1 'Buddy kullanarak.

UpDown1.BuddyControl = Text1

UpDown2.BuddyProperty = "Left"

UpDown1.BuddyProperty = "Top"

End Sub

Private Sub UpDown1_DownClick() 'Olayları kullanarak.

Text1.Top = Text1.Top + 4

End Sub

Private Sub UpDown1_UpClick()

Text1.Top = Text1.Top - 4

End Sub

Private Sub UpDown2_DownClick()

Text1.Left = Text1.Left - 4

End Sub

Private Sub UpDown2_UpClick()

Text1.Left = Text1.Left + 4

BÖLÜM 11

11.KARAKTERSEL VE SAYISAL FONKSİYONAR

11.1. LEN() Fonksiyonu

Len() fonksiyonu yardımı ile kendisine parametre olarak verilen karaktersel veya başka bir tip bilginin uzunluğu bulunur.

Yazılımı:

LEN(Bilgi veya Bilgi İçeren Değişken)

Örnek 11.1 :

```
A="Ahmet Turan"  
U=Len(A)           ' U 5 olur.  
U=Len("Ahmet")     ' U 5 olur.
```

11.2. LEFT() Fonksiyonu

Bu fonksiyon dışarıdan parametre olarak karaktersel bir bilgiyi, karaktersel bilgi içeren bir değişkeni veya Variant tipli alabilmektedir. Bu fonksiyon kendisine parametre olarak verilen Variant veya karaktersel (String) tipli değişken veya sabit bilginin soldan itibaren istenilen kadarını ayırmak, çıkarmak amacıyla kullanılmaktadır.

Yazılımı:

LEFT(Variant veya Karakterse ifade, uzunluk)

Örnek 11.2 :

```
Ad="Ahmet Turan"  
Sonuç=Left(Ad,5)           'Sonuç "Ahmet" olur.  
Sonuç=Left("Ahmet Turan",8) 'Sonuç "Ahmet Tu" olur.
```

Visual Basic'te Left() fonksiyonu ile benzer işleve sahip olan Left\$() adında bir fonksiyon daha vardır. Left() fonksiyonu geriye Variant tipte bilgi döndürürken, Left\$() fonksiyonu geriye String tipte bilgi döndürmektedir.

Örnek 11.3:

```
Dim Sayı As Integer  
Sayı=12345  
Karakter=6789  
Sonuç=Left(Sayı,2)           'Sonuç 12 olur.  
Sonuç=Left$(Sayı,2)         'Hata Verir.
```

Sonuç=Left(Karakter,3)	'Sonuç 123 olur.
Sonuç=Left\$(Karakter,3)	'Sonuç 123 olur.

11.3. RIGHT() Fonksiyonu

Bu fonksiyon, kendisine parametre olarak verilen karaktersel veya Variant tipli bilgi veya değişkenin içeriğinin sağdan itibaren istenilen kadarını ayırmak, çıkarmak amacıyla kullanılır.

Yazılımı:

RIGHT(Karaktersel veya Variant ifade,uzunluk)

Örnek 11.4:

Ders="Bilgisayar"	
Sonuç=Right(Ders,4)	'Sonuç "ayar" olur.
Sonuç=Right(Bilgisayar,5)	'Sonuç "sayar" olur.

Right() Fonksiyonundan başka Right\$() adında bir fonksiyon daha vardır. Right\$() Fonksiyonu dışarıdan parametre yalnızca karaktersel ve Variant tipli bilgi ve bu tipteki bilgileri içeren değişkenleri alabilmektedir.

Örnek 11.5:

Dim Sayı As Integer	
Sayı=1998	
Karakter=1999	
Sonuç=Right(Sayı,3)	'Sonuç 998 olur.
Sonuç=Right\$(Sayı,3)	'Hata Verir.
Sonuç=Right(Karakter,2)	'Sonuç 99 olur.
Sonuç=Right\$(Karakter,2)	'Sonuç 99 olur.

11.5. LTRIM() ve LTRIM\$ Fonksiyonları

LTRIM() Fonksiyonu kendisine parametre olarak verilen herhangi bir tipteki, LTRIM\$() ise karaktersel tipteki veya Variant bilgilerin baş tarafında bulunan boşlukları çıkarmak amacıyla kullanılır.

Yazılımı:

LTRIM(KarakterSel Bilgi)

Örnek 11.6:

Sehir="Erzurum" " " " " " "
Sonuç=LTRIM(Sehir) "Sonuç "Erzurum" olur.

11.6. RTRIM() ve RTIM\$() Fonksiyonları

RTRIM() fonksiyonu parametre olarak aldığı karakterisel bir bilginin arka tarafında yani sağında bulunan boşlukları çıkarmak amacıyla kullanılmaktadır. Parametre olarak verilen karakterisel bilginin arka tarafında hiçbir boşluk yoksa, geriye bilginin ilk hali dödürülür.

Yazılımı:

RTRIM(Karakterisel Bilgi)

11.7. TRIM() Fonksiyonu

Bu fonksiyon, parametre olarak aldığı karakterisel bilginin hem başında hem de sonunda yer alan boşlukları çıkarıp bilginin yeni şeklini geriye döndürür.

Yazılımı:

TRIM(Karakterisel Bilgi)

11.8. LCASE\$() Fonksiyonu

Bu fonksiyon parametre olarak aldığı karakterisel bilginin içinde bulunan alfabetik karakterlerin büyük olanlarını küçük harfe çevirmektedir. LCASE\$() Fonksiyonu kendisine parametre olarak verilen karakterisel bilginin yalnızca alfabetik olan karakterleri üzerinde etkili olmaktadır.

Örnek 11.7:

Ad="AHMET"
Sehir="SİVAS"
Print LCASE\$(Ad) "Çıktısı "ahmet"
Print LCASE\$(Sehir) "Çıktısı "sıvas"

Burada Tükçe'ye özgü "İ" harfini küçük harfe çevirmedi.

11. 9. UCASE\$() Fonksiyonu

Bu fonksiyon parametre olarak aldığı karakterisel bilginin içinde bulunan küçük harfleri büyük harfe çevirir. UCASE\$() Fonksiyonu kendisine parametre olarak verilen karakterisel bilginin yalnızca alfabetik olan karakterleri üzerinde etkili olmaktadır.

Yazılımı:

UCASE\$(Karaktersel Bilgi)

Örnek 11.8:

```
Ad="ismail"  
Sehir="İğdır"  
Print LCASE$(Ad)           'Çıktısı "ISMAIL"  
Print LCASE$(Sehir)        'Çıktısı "İĞDIR"
```

Burada Türkçe'ye özgü "i", "ı", "ğ" harflerini büyük harfe çevirirken yanıldı.

11.10. MID ve MID\$ Fonksiyonları

Bu iki fonksiyon kendilerine parametre olarak verilen karaktersel veya Variant bir bilginin belirlenen bir kısmını ayırıp çıkarmak amacıyla kullanılır. MID() ve MID\$() fonksiyonları dışarıdan parametre olarak üç bilgi almaktadırlar. Bunun karşılığında elde edilen alt bilgi geriye döndürülmektedir. Birinci **Parametrede** verilen bilginin **Başlama Yeri**'nden itibaren **Alınacak Karakter Sayısı** kadarı alınarak geriye döndürülür.

Yazılımı:

MID(Karakter)

Örnek 11.9

```
Ad="Ahmet Turan"  
Print MID$(Ad, 1, 5)           'Çıktısı "Ahmet"  
Print MID$(Ad, 7, 5)           'Çıktısı "Turan"
```

11.11. INSTR() Fonksiyonu

Instr() Fonksiyonu karaktersel bir bilgide başka bir karaktersel bilgiyi aramak amacıyla kullanılmaktadır.

Yazılımı:

INSTR(Başlama Yeri, Karaktersel Bilgi, Aranacak Bilgi)

Örnek 11.10:

Ad="ahmet turan"
Sonuç=INSTR(1, Ad, "a") 'Sonuç 1 olur.
Sonuç=INSTR(6, Ad, "t") 'Sonuç 7 olur.
Sonuç=INSTR(5, Ad, "a") 'Sonuç 10 olur.

11.12. STRCOMP() Fonksiyonu

Bu fonksiyondan karaktersel veya Variant tipteki iki sabit bilgiyi veya bilgi içeren iki değişkeni birbirleri ile karşılaştırmak için yararlanılır. Bu fonksiyon dışarıdan biri seçilerek olan 3 parametre almaktadır.

Yazılımı:

STRCOMP(1. Bilgi, 2. Bilgi, Karşılaştırma Şekli)

Örnek 11.12 :

Dim Ad As String
Dim Ad1 As String
Ad="Ahmet"
Ad1="ahmet"
Sonuç=StrComp(Ad,Ad1) 'Sonuç 1 olur.
Sonuç=StrComp(Ad,Ad1,1) 'Sonuç 0 olur.

Bu fonksiyon karşılaştırma yaparken büyük – küçük harf ayrımı yapar. Eğer karşılaştırma sırasında büyük – küçük ayrımı yapılması istenmiyorsa StrComp Fonksiyonuna 3. Parametre olarak 1 sayısal değeri verilmelidir.

11.13. STR\$() Fonksiyonu

STR\$() fonksiyonu ile sayısal bir bilgi karaktersel bir bilgi tipine çevrilir. Geriye döndürülen bilginin Variant tipte olması isteniyorsa STR\$() yerine STR() Fonksiyonu kullanılmalıdır.

Yazılımı:

STR(Sayısal Bilgi)

Örnek 11.13:

Sayı=975.31028
Sonuç=STR(Sayı) 'Sonuç "975.31028" olur.

Bilindiği gibi karaktersel bilgiler ile sayısal bilgilere "+" operatörü ile toplanmazlar. Buna bir örnek verecek olursak;

Dim No As Long

No=9728
Ad="Ahmet"
Print STR(9728) + " " + Ad 'Çıktısı "9728 Ahmet" olur.

Görüldüğü gibi sayısal bilgi içeren değişkenin içeriği STR() Fonksiyonu ile karaktersele yapıldığı için hata oluşmaz.

11.14. STRIN\$() ve STRING() Fonksiyonları

Yaptıkları işlev bakımından bir birlerine çok benzeyen String\$() ve String() fonksiyonları, bir karakterin istenilen sayıda ekrana veya yazıcıya arka arkaya tekrarlanarak yazılmasını sağlarlar. String\$() fonksiyonu geriye karaktersele bilgi döndürürken String() fonksiyonu geriye Variant tipte bilgi döndürmektedir.

Yazılımı:

STRING\$(Sayı, Karakter)

Örnek 11.14:

Print String\$(5,"+")	'+++++
Print String\$(5,Chr(65))	'AAAAA
Print String\$(5,"Bilgisayar")	'BBBBB

11.15. CHR() Fonksiyonu

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

Yazılımı:

CHR(Sayı)

Örnek 11.15

Sonuç=CHR(65)	' Sonuç "A" olur.
Sonuç=CHR(126)	' Sonuç "~" olur.

11.16. TAB() Fonksiyonu

GW ve QBASIC'te Print deyimi ile yapılacak yazma işlemi öncesi ekranın bilgi yazılacak yerini belirlemek için LOCATE deyiminden yararlanılıyordu. Ekranın bilgi yazılacak yerini satır bazında belirlemek için istenirse TAB() Fonksiyonundan yararlanılır.

Yazılımı:

TAB(Ekranın Bilgi Yazılacak Sütunu)

11.17. SPC() Fonksiyonu

Bu fonksiyon boşluk oluşturmak amacıyla kullanılır. Program içinde ne kadar boşluk oluşturmak isteniyorsa, boşluk sayısı SPC() fonksiyonuna parametre olarak verilmelidir.

Yazılımı

SPC(Boşluk Sayısı)

11.18. ASC() Fonksiyonu

Bu fonksiyon, herhangi bir karakter sel bilginin ilk karakterinin ASCII kodunu bulmak amacıyla kullanılmaktadır.

Yazılımı:

ASC(Karakter sel Bilgi)

Örnek 11.16:

Sonuç=ASC("S")	' Sonuç 83 olur.
Sonuç=ASC("Sivas")	' Sonuç 83 olur.

11.19. VAL() Fonksiyonu

Bu fonksiyon karakter sel bir bilginin sayısal bilgiye dönüştürülmesi amacıyla kullanılmaktadır. VAL() Fonksiyonu parametre olarak aldığı karakter sel bilginin sayısal bilgiye çevrilebilen kısmını ondalık noktaları ile birlikte sayısal bilgiye çevirerek geriye döndürür. Sayısal bilgi tipine çevrilmek istenen karakter sel bilginin içinde alfabetik karakterler varsa çevirme işlemi ilk alfabetik karaktere rastlayıncaya kadar yapılır.

Yazılımı:

VAL(Karakter sel Bilgi)

Örnek 11.17:

Sonuç=VAL("123.4567")	'Sonuç 123.4567 olur.
Sonuç=VAL("123Sivas")	'Sonuç 123 olur.
Sonuç=VAL("A123.4567")	'Sonuç 0 olur.

Sonuç=VAL("12ERzurum3.4567") 'Sonuç 12 olur.

11.20. ABS() Fonksiyonu

Bir sayının mutlak değerini bulmak için kullanılır. Bir sayının artı veya eksi olmasından çok büyüklüğünün arandığı durumlarda kullanılır.

Yazılımı:

ABS(Sayı)

Örnek 11.19:

Sonuç=ABS(10) 'Sonuç 10 olur.
Sonuç=ABS(-10) 'Sonuç 10 olur.

11.21. INT() Fonksiyonu

Ondalık nokta içeren bir sayıyı tam sayıya çevirmek amacıyla kullanılır. INT() Fonksiyonu kendisine parametre olarak verilen ondalık nokta içeren pozitif sayılarda yuvarlama yapmayıp noktadan sonrasını atmaktadır. Ancak INT() Fonksiyonuna parametre olarak verilen sayı negatif ise sayı bir sonraki küçük sayıya yuvarlatılmaktadır.

Yazılımı:

INT(Ondalık Sayı)

Örnek 11.20 :

Sonuç=INT(10.41) 'Sonuç 10 olur.
Sonuç=INT(10.95) 'Sonuç 10 olur.
Sonuç=INT(-10.41) 'Sonuç -11 olur.
Sonuç=INT(-10.80) 'Sonuç -11 olur.

11.22. SQR() Fonksiyonu

Bu fonksiyon pozitif bir sayının karekökünü bulmak amacıyla kullanılır.

Yazılımı:

SQR(Karekökü Alınacak Sayı)

Örnek 11.21: Sonuç=SQR(16) 'Sonuç 4 olur.

11.23. SIGN() Fonksiyonu

Sayısal bir sabit veya sayısal bilgi içeren değişkenin pozitif veya negatif olup olmadığını öğrenmek için SIGN() Fonksiyonundan yararlanılmaktadır. Parametre olarak verilen sayısal ifade negatif değere sahipse SIGN() fonksiyonu geriye -1, pozitif ise 1 değerini geriye döndürür. SIGN() fonksiyonuna parametre olarak verilen bilgi 0 değerini içeriyorsa SIGN() fonksiyonu bu kez de 0 değerini geriye döndürür.

Yazılımı:

SIGN(Sayısal Bilgi)

Örnek 11.22:

Sonuç=SIGN(10)	' Sonuç= 1 olur.
Sonuç=SIGN(-10)	' Sonuç= -1 olur.
Sonuç=SIGN(0)	' Sonuç= 0 olur.

11.24. SHELL() Fonksiyonu

Shell fonksiyonu ile bilgisayara kurulu olan herhangi bir program Visual Basic projesi dahilinde çalıştırılabilir.

Yazılımı:

SHELL(Program Adı)

Örnek 11.23:

```
Private Sub Command1_Click()  
    Shell ("C:\Progra~1\Winamp.EXE")  
End Sub
```

11.25. ChDrive

Aktif sürücüyü değiştirir.

Yazılım

ChDrive "Sürücü_Adı"

11.26. ChDir

Aktif dizini değiştirmeye yarar.

Yazılım

ChDir "Dizin" (ChDir "c:\VB")

11.27. MkDir

Dizin Oluşturmak için bu komut kullanılır.

Yazılım

MkDir "Dizin_Adi"

11. 28. Rmdir

İsmi verilen dizini siler.

Yazılım

Rmdir "Dizin_Adi"

11.29. Kill

İsmi verilen dosyayı siler.

Yazılım

Kill "Dosya_Adi" (Kill "c:\VB\Ornek.Bas")

11.30. Name

Verilen dosyanın adını değiştirir.

Yazılım

Name "Dosya_Adi" As "Yeni_Dosya_Adi"

11.31. CurDir

Aktif dizini öğrenmemizi sağlar.

Yazılım

CurDir “Sürücü”
11.32. FileCopy

Bir dosyayı bir yerden başka bir yere kopyalamamızı sağlar.

Yazılım

FileCopy “Kaynak_Dosya”, “Hedef_Dosya”

Dosya adı, sürücü ve yol içerebilir içerebilir. Joker karakterler (?,*) içeremez.

11.33. FileLen

Dosya boyutunu byte olarak bildirir.

Yazılım

FileLEn (“Dosya_Adı”)

11.34. FileDateTime

Dosyanın oluşturulduğu tarih ve saati verir.

Yazılım

FileDateTime (Dosya_Adı)

11.35. GetAttr

Dosyanın Attributesini öğrenmeye yarar.

Yazılım

GetAttr (“Dosya_Adı”)

Geri dönen değer aşağıdaki sayılarla And işlemine tabi tutularak o özelliğin bulunup bulunmadığı kontrol edilir.

Anlamı	Sembolik	Sayısal Değer
Yalnız Okunabilir Dosya (Read Only)	VbReadOnly	1
Gizli Dosya (Hidden)	VbHidden	2
Sistem Dosyası	VbSystem	4
Volum Etiketi	VbVolum	8
Dizin	VbDirectory	16
Arşiv Dosyası	VbArchive	32
Normal	VbNormal	0
Alias İsmi	VbAlias	64

Tablo-11.1. Dosyaların Özellikleri

Bir dosya aynı anda birden çok özelliğe sahip olacağı için özellikleri öğrenmek için GetAttr fonksiyonunda dönen değeri tablodaki sayısal değerlerle AND işlemine tabi tutmak gerekir.

Örnek 11.24 :

```
Dim attr
attr=GetAttr("\DOS\MSDOS.SYS")
Print "MSDOS.SYS";
If attr AND 1 Then Print "R";
If attr AND 4 Then Print "S";
```

11.36. SetAttr

Dosyanın Attributesini Değiştirir.

Yazılım

SetAttr "Dosya_Adl", attribute

Attribute parametresinin alabileceği değerler ve anlamları şöyledir:

Sembol	Sayısal	Anlamı
VbNormal	0	Normal
VbReadOnly	1	Read-Only
VbHidden	2	Hidden
VbSysytem	4	System
VbArchive	32	Arşiv

Tablo-11.2. Attribute parametresinin alabileceği değerler ve anlamları

Bu değerler Or işlemine tabi tutularak bir kaçı aynı anda aktif yapılabilir.

Örnek 11.25:

SetAttr "c:\dos\msdos.sys",vbHidden

11.37. Dir

Dosya ve dizinleri listelemeye yarayan komuttur.

Yazılım

Dir [(Dosya_Adı[,attr])]

Örnek 11.26:

Dir (“*.Bas”,1+2+4)

11.38. SavePicture

Bir nesnenin içindeki resmi dosyaya kaydedebiliriz.

Yazılım

SavePicture resim_bilgisi, Dosya_adı

Resim_bilgisi : Kaydedilecek resmin bulunduğu kontrolün özelliği. Bu özellik bir kontrolün picture veya image özelliği olabilir.

Örnek 11.28:

SavePicture Picture1.Picture, “fırat.bmp”

11.39. Date

Bugünün tarihini öğrenmeye ve değiştirmeye yarar.

MsgBox (“Bugün”, &Date) ‘Bugünün tarihini gösterir

Date = “12/8/1999” ‘Tarihi değiştirmek için

11.40. CVDate

İki tarih arasındaki farkı bulmak için kullanılır.

Dim a
a=Date-CVDate(“17/05/1999”)
MsgBox (“Fark”, & a)

11.41. Time

Saati öğrenmeye ve değiştirmeye yarar

MsgBox (“saat:”, &Time)

11.42. WeekDay

Günü öğrenilecek tarihi bulmak için kullanılır. Geri dönen değer haftanın gününü belirten bir sayıdır. Bu sayı 1 ile 7 arasındadır.

Yazılım

WeekDay (Tarih)

11.43. Day

Verilen tarihin gününü verir.

Yazılım

Day (Tarih)

11.44. Month : Tarihin ayını verir.

Yazılım

Month (Tarih)

11.45. Year

Tarihin yılını verir.

Yazılım

Year (Tarih)

11.46. Hour

Zamanın saatini verir.

Yazılım : Hour (Zaman)

11.47. DatePart

Verilen tarihin yılın kaçınıcı haftası, günü, kaçınıcı çeyreğinde olduğunu bulur.

Yazılım

DatePart (İstenen, Tarih)

İstenen:

q: Verilen tarihin yılın kaçınıcı çeyreğinde olduğunu bulur.

ww : Verilen tarihin yılın kaçınıcı haftası olduğunu bulur.

y : Verilen tarihin yılın kaçınıcı günü olduğunu bulur.

12.VİSUAL BASİCTE VERİ TABANI UYGULAMALARI

12.1.Visual Basic ile Veri Tabanı Oluşturma

Veri tabanı işlemleri tüm programcılar tarafından merak edilen ve yapılmak istenen işlemlerdir. Günümüzde özellikle ticari amaçlı programlar, bir şirketin veya kuruluşun verilerini tutabilen programlar revaçta bulunmaktadır. Dolayısı ile bu tür gereksinimleri karşılamak için veri tabanı yani kısaca dosyalama olarak bilinen yapıyı iyi bilmek gerekmektedir.

Veri tabanı işlemlerinde oldukça değişik dosya tipleri kullanılabilir ama en çok bir Office dosya tipi olan MDB uzantılı Access veri tabanı dosyaları bu işlemde kullanılmaktadır. Özellikle de Visual Basic ile oluşturulan veri yapısı ile Access programının veri yapısı birbirine çok benzediğinden bu konu üzerinde veri tabanı işlemlerini anlatacağım.

Access' dan başka Dbase, Foxpro, Clipper, Paradox vb. diğer programlarda da hazırlanan dosyalar ile veri tabanı oluşturulabilir.

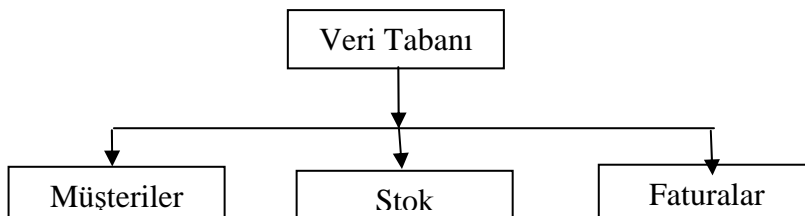
Visual Basic 6.0 ile beraber gelen veritabanı tasarım ortamı kullanıcıya büyük kolaylıklar sağlamaktadır. Gerek tasarım gerekse çalışma zamanında Visual Basic ortamından çıkmadan veri tabanı dosyası tasarlamak, tablolar oluşturmak dolayısı ile form tasarımını kolay yapmak ve rapor oluşturmak oldukça zevkli bir hale getirilmiştir.

12.2.Veritabanını Oluşturan Elemanlar

Bir veri tabanı dosyası belli temel elemanların bir araya getirilmesi ile meydana gelir. Veritabanının temelini veri alanları oluşturur. Veri alanları bir araya gelerek tabloları oluşturur. Bir yada birden fazla tablo bir araya gelerek veritabanı dosyası oluşturur. Bu elemanların tamamını Visual Basic programı bir nesne olarak algılamaktadır. Dolayısı ile bu elemanlar hem birbirinden bağımsız hem de birbirine bağımlı olarak çalışmaktadırlar. Bunları kısaca şöyle açıklayabiliriz.

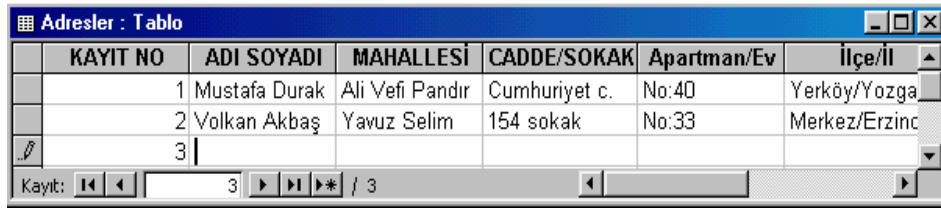
12.2.1. Database (Veritabanı) Nesnesi

Bir yada birden fazla tabloyu içeren ve aynı zamanda birbiri ile ilişkili olan tablo gruplarını bünyesinde bulunduran tablo grubuna denir. Database bir dosya türüdür. Yukarıda da açıkladığımız gibi Access,Foxpro,Database IV,V vb dosya türlerinden seçilebilir. Örneğin her hangi bir veri tabanına ait tablolar şu şekilde olabilir.



12.2.2. Table (Tablo) Nesnesi

Bir grup veri alanını içeren yapıya table (tablo) denir. Örneğin adresler ile ilgili bilgilerin yer aldığı tablo aşağıdaki gibi olabilir.

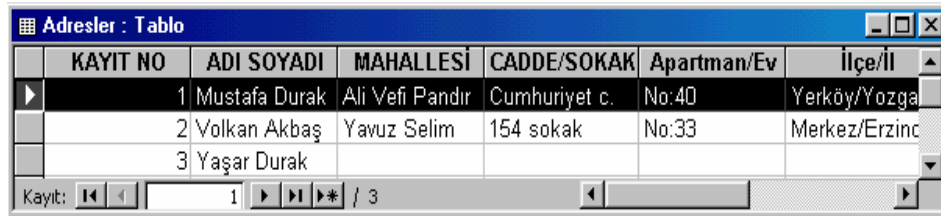


KAYIT NO	ADI SOYADI	MAHALLESİ	CADDE/SOKAK	Apartman/Ev	İlçe/İl
1	Mustafa Durak	Ali Vefi Pandır	Cumhuriyet c.	No:40	Yerköy/Yozga
2	Volkan Akbaş	Yavuz Selim	154 sokak	No:33	Merkez/Erzinc
3					

Şekil 12.1. Table (Tablo) Nesnesi

12.2.3. Record (Kayıt)

Bir tabloda yer alan ve veritabanını oluşturan sahaların veya alanların içerdiği bilgilerin oluşturduğu yapıya Record (Kayıt) denir. Şekil-12.1’de yer alan Kayıt no, Adı Soyadı, Mahallesi, Cadde/Sokak, Apartman/Ev ve İlçe/İl olarak adlandırılmış bölümlerin içerdiği bilgiler kayıtları oluşturur. Örnek olarak Şekil-12.2’de yer alan seçili satır bir kayıttır.



KAYIT NO	ADI SOYADI	MAHALLESİ	CADDE/SOKAK	Apartman/Ev	İlçe/İl
1	Mustafa Durak	Ali Vefi Pandır	Cumhuriyet c.	No:40	Yerköy/Yozga
2	Volkan Akbaş	Yavuz Selim	154 sokak	No:33	Merkez/Erzinc
3	Yaşar Durak				

Şekil-12.2. Record (Kayıt)

12.2.4. Field (Veri Alanı)

Bir tablodaki her bir kaydı oluşturan en temel yapıya veri alanı veya veri sahası denir. Veri alanları bizim Veritabanımızın değişkenleridir. Örneğin Adresler tablosundaki Kayıt no, Adı Soyadı, Mahallesi, Cadde/Sokak, Apartman/Ev ve İlçe/İl diye adlandırılmış her bölüm bir alanı oluşturmaktadır.(Şekil-12.3)

Anahtar Saha →

	Alan Adı	Veri Türü
🔑	KAYIT NO	Otomatik Sayı
	ADI SOYADI	Metin
	MAHALLESİ	Metin
	CADDE/SOKAK	Metin
	Apartman/Ev	Metin
	İlçe/İl	Metin

Şekil-12.3. Veri Alanları

12.2.4. Index (Anahtar Kayıt)

Her bir kaydı temsil eden ve kayıtların sıralanmasında önemli bir rol alan veri alanına anahtar kayıt denmektedir. Şekil-12.3'de görüldüğü gibi anahtar alan Kayıt no alanıdır.

12.2.5. Query (Sorgu)

Bir yada birden fazla tablodan belli şartlara bağlı olarak istenilen kayıtların listelenmesini sağlayan komutlara Query denir. Belli kayıtlara erken ulaşmamızı sağlar.

12.3. Veri Tabanının Özellikleri

Genellikle veri tabanı programlarının şu özelliklere sahip olması istenir.

- 1.)Kayıt ekleme
- 2.)Kayıt silme
- 3.)Kayıt bulma
- 4.)Bir önceki kayda gitme
- 5.)Bir sonraki kayda gitme
- 6.)İlk kayda gitme
- 7.)Son kayda gitme

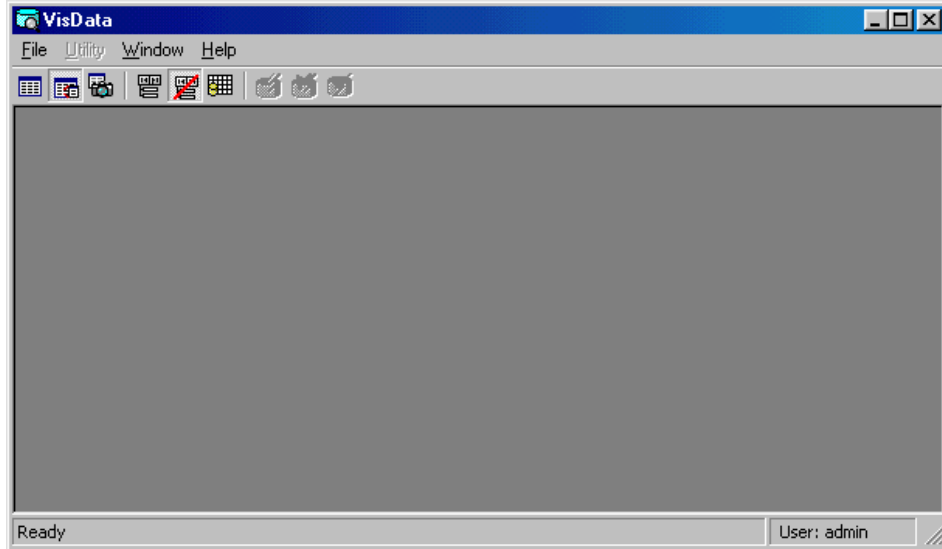
Bu sıraladığımız işlemler program kodu yazarak yapılacak işlemlerdir. Program kodu tamamı ile programcının bilgi ve becerisi dahilinde değişebilen bir sistemdir. Her programcı kendine ait bir metot ile veri kaydını vb. işlemleri gerçekleştirebilir. Şimdi bu özellikleri uygulayarak kitabımızın ilerleyen bölümlerinde öğrenmeye çalışacağız.

12.4. Visual Data Manager ile Veri Tabanı Tasarımına Giriş

Kitabımızın 4.bölümünde açıklandığı gibi Visual Data Manager (Visdata) ile veri tabanı oluşturma işleminde bize kolaylık sağlar. Visual Data Manager ile Access(2.0,7,0), Database(III,IV,V) ,Foxpro(3.00,4.00,5.2), Paradox(5.0,4.X,3.X) ODBC veri tabanı programlarının desteklediği veri tabanı dosyaları oluşturmak mümkündür. Gerek bu veri tabanı programlarına ait bir dosya ve gerekse direkt

Visdata programı vasıtası ile bu formatlarda oluşturulan bir dosya Visual Basic tarafından rahatlıkla kullanılabilir.

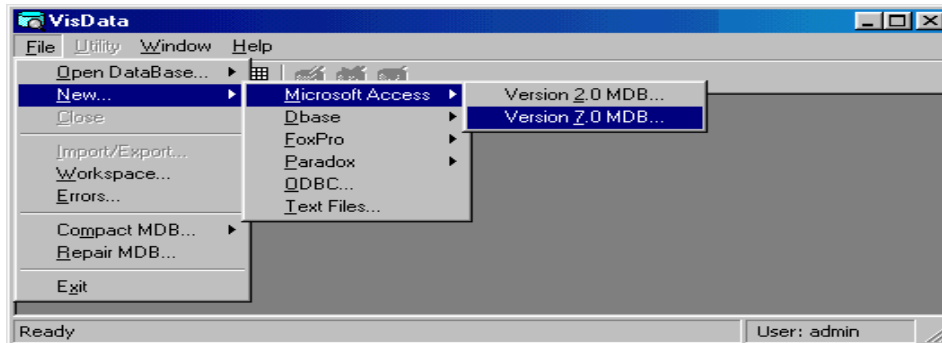
Programa girmek için Visual Basic ortamından Add-ins/Visual Data Manager menü seçeneklerini kullanırız.Bu işlemten sonra Şekil-12.4’de görülen program penceresi görülecektir.



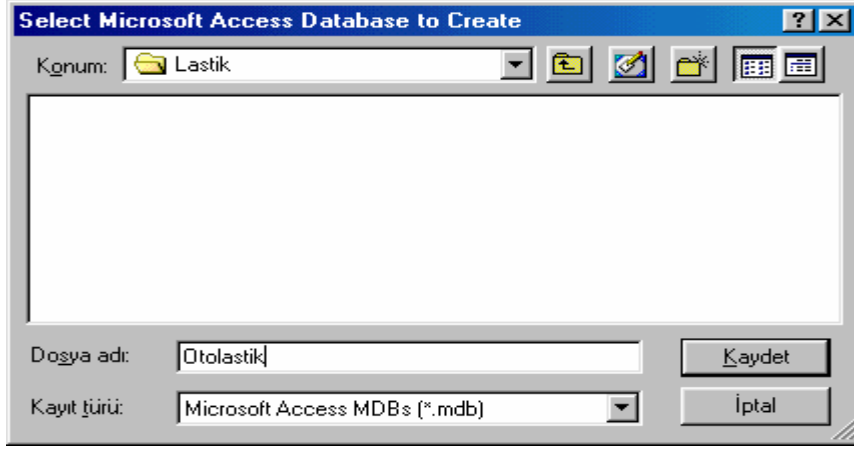
Şekil-12.4 Visdata Penceresi

12.4.1. VisData ile Veri Tabanı Tasarımı

İlk veritabanı dosyamızı oluşturmak için Şekil-12.5’de görüldüğü gibi sırasıyla File-New-Microsoft Access-Version 7.0 mdb seçeneklerini kullanarak işleme başlayalım.Bu seçenekler girildikten sonra veritabanı dosyamızın ismini yazarak kaydedeceğimiz konumu seçmemiz için Şekil-12.6’da görülen diyalog penceresi karşımıza çıkacaktır.Aynen Access ile açılan veri tabanı dosyaları gibi önce dosya adı sonrada konumunu seçerek kaydetme işlemini gerçekleştirmeliyiz. Örnek olarak Otolastik.mdb dosyasını “C:\Lastik\” konumuna kaydettim.

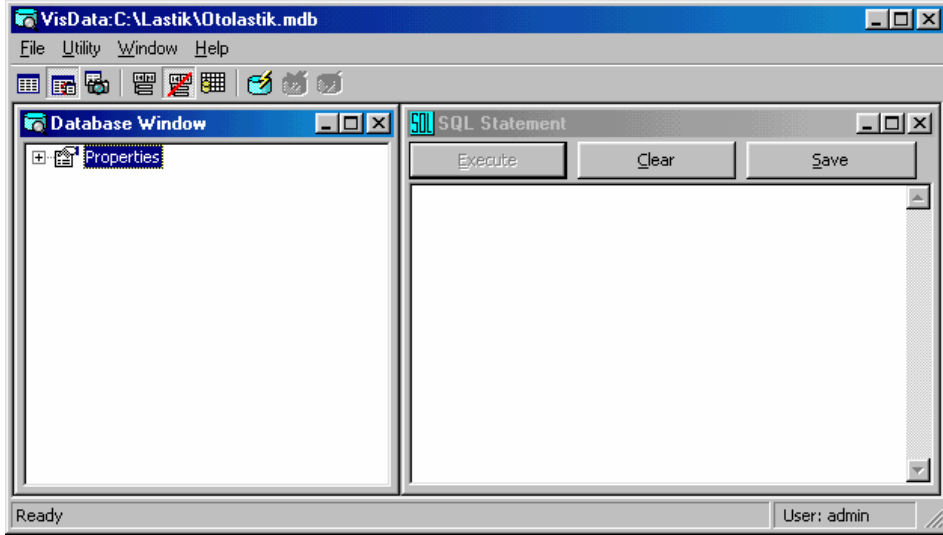


Şekil-12.5 Yeni Dosya Oluşturma



Şekil-12.6 Veri Tabanı Dosyasını Kaydetme

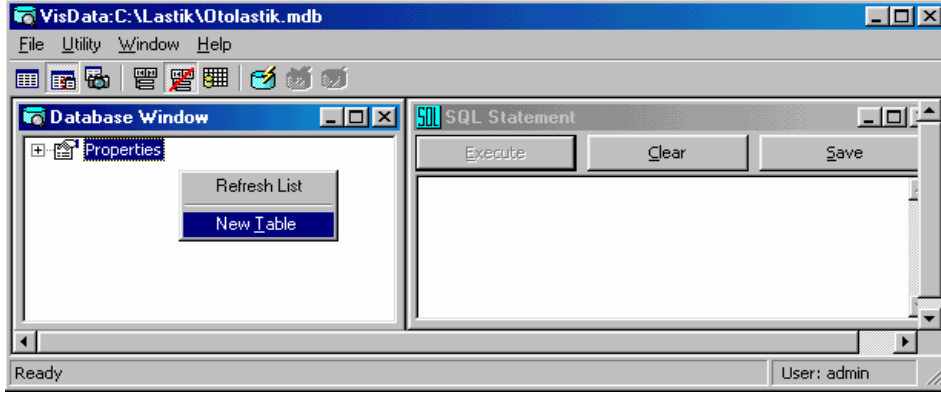
Visdata programı yardımı ile Otolastik.mdb olarak oluşturduğumuz dosyamız bizim veri tabanı dosyamızdır. Şimdi bu veri tabanı dosyamıza tablo, index, kayıt,sorgu ve veri alanı ekleme işlemlerini şimdi sırası ile gerçekleştireceğiz.Dosyamızı kaydettikten sonra Şekil-12.7’de görülen pencere karşımıza çıkacaktır.



Şekil-12.7. VisData Programına Tablo Tasarım Bölümü

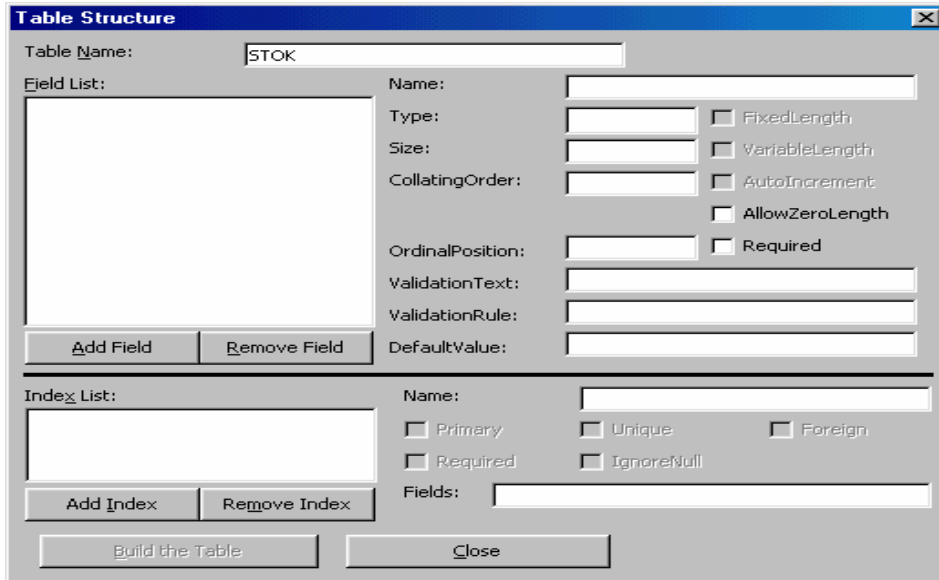
Yukarıda görülen şekil Visdata programına ait veri tabanı dosyalarının özelliklerinin değiştirilip,yenilenebileceği tasarım alanının şeklidir. Tablo oluşturma ve değiştirme bu alanda yapılmaktadır.

Bir yanda Database window diğer yanda ise SQL Statement pencereleri yer almaktadır. Şimdi oluşturduğumuz Otolastik veritabanına ait olacak olan tabloları oluşturmak için Database window penceresinde iken Mouse ile sağ tıklayıp çıkan menüden New Table seçeneğini seçelim (Şekil-12.8).



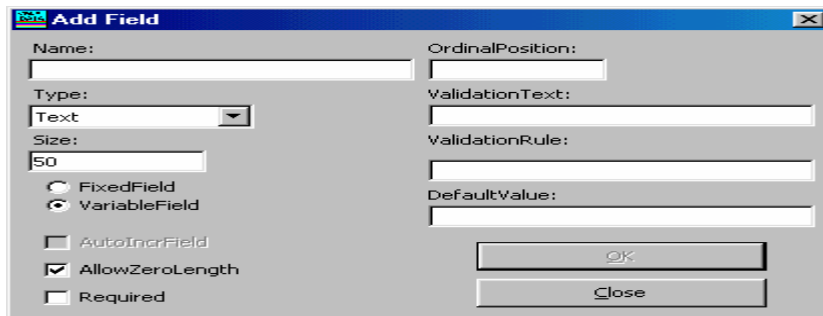
Şekil-12.8 Tablo Oluşturma

New Table seçeneğini seçtikten sonra karşımıza aşağıdaki veri tabanına ait tablo tasarımının yapıldığı tasarım penceresi çıkacaktır.



Burada Table Name yazılan kutuya tablomuza vereceğimiz ismi yazalım.Örneğin ben Stok isimli bir tablo ismi verdim.

Bundan sonra Stok tablosuna ait olacak olan veri alanlarını ekleme işlemini yapalım. Bunun için Table Structure penceresinde yer alan Add Field düğmesine tıklayarak aşağıda görülen Add Field penceresine ulaştım.



Add Field penceresinde yer alan seçenekler ve alanların anlamlarını şu şekilde açıklayabiliriz.

Name:Veri alanı isimlerinin belirlendiği bölümdür. Örneğin Stok tablosu için STOKKODU, LASTIKMARKASI, LASTIKTIPI, LASTIKEBADI, ALISFIYATI, KDV, SATISFIYATI, STOKADEDİ ve STOKTARIHI gibi veri alanı isimlerini bu bölüme gireceğiz.

Type:Bu kısımda veri alanının türü seçilir.Örnek olarak LASTIKMARKASI için Text, ALISFIYATI için Double tipleri verilebilir.

Size: Bu kutuya veri alanı uzunluğu girilir. Eğer buraya 15 girilmiş ise en fazla 15 karakterlik bir bilgi girilecektir.

FixedField seçeneği seçili ise alana girilecek olan her kaydın uzunluğu sabit olacaktır.

VariableField seçeneği seçili ise alanların uzunlukları farklı olabilecektir.

AutoIncField seçeneği seçili ise ilgili alan otomatik olarak değerini birer artıracaktır. Bu sadece sayısal alanlar için aktif hale gelir.

AllowZeroLenght seçeneği seçili ise ilgili alana hiçbir şey girilmese de işlem devam edebilecektir.

Requried seçeneği seçili ise ilgili alana bilgi girişi yapılmadan bir sonraki kayıta geçiş mümkün olmayacaktır.

OrdinalPosition kutusuna veri alanı sırası girilir. Eğer her hangi bir işlem yapılmaz ise 0'dan başlayarak veri alan numaralarını ayarlar.

Validation Text kutusuna alana girilecek bilgi türüne ait açıklama yazılır.

Validation Rule kutusuna ise alana girilecek olan sınır değeri belirlenir. Örneğin STOKKODU alanı için 1-1000 arası bir sayı girilmesi için >1 AND <1000 ifadesi bu kısma yazılmalıdır. Eğer bu formata aykırı bir bilgi girilirse Validation text kutusuna yazılan açıklama karşımıza çıkacaktır.

Default Value kutusuna ise girilecek alana ait varsayılan değer girilecektir.Örneğin LASTIKMARKASI her zaman XXXX marka ise bu kısma XXXX yazarak bir daha bu veri alanına aynı bilgiyi yazmamıza gerek kalmayacaktır.

Stok tablomuz için Add Field penceresine girilecek alanlarımız şu şekilde olacaktır.

ALAN ADI	Type	Size	AutoIncrement
----------	------	------	---------------

STOKKODU	Long	4	Evet
LASTIKMARKASI	Text	30	
LASTIKTIPI	Text	15	
LASTIKEBADI	Text	30	
ALISFIYATI	Double	8	
KDV	Byte	1	
SATISFIYATI	Double	8	
STOKADEDİ	Integer	2	

Şekil-12.9. Table Structure Penceresi

12.4.2. Tablo Yapısını Değiştirmek

Tablo yapısını değiştirmek, yeni alanlar eklemek veya varolan alanlardan istenileni silmek için Database window penceresindeki Stok tablosu üzerinde iken sağ tıklayıp açılan menüden Design seçeneğini seçmemiz gerekmektedir. Bundan sonra karşımıza Table Structure çıkar burada veri tabanı tablosu üzerinde istenilen değişiklikler yapılabilir.

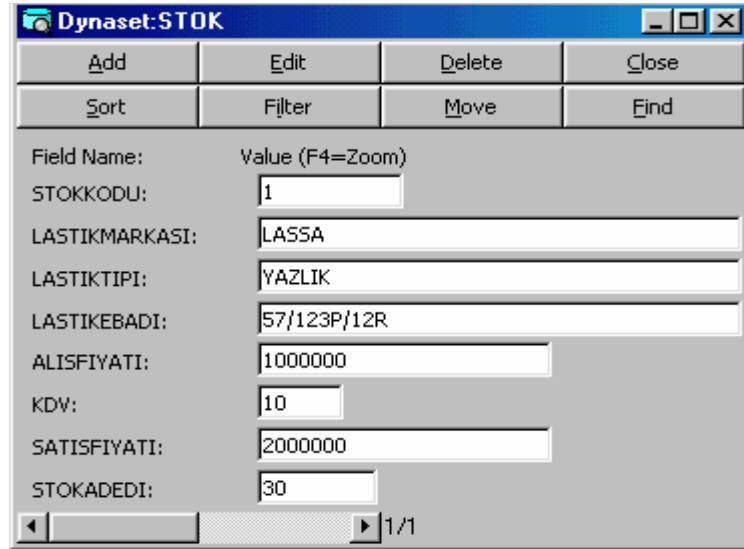
12.4.3. Tabloda Index Tanımlamak

Index tanımlamak verilerin saklanması, silinmesi, bulunması, kayıtlar arasında dolaşma gibi işlemlerde kolaylık ve hız sağlar. Index tanımlamak için Table Structure penceresinde yer alan Add Index düğmesine basılır. Bu işlemten sonra aşağıdaki pencere karşımıza çıkacaktır.

Burada index yapılacak veri alanı seçilerek Indexed Fields kutusuna aktarılır. Name kutusuna ise index yapılacak alana bir isim verilir. Örnek olarak Stokkodu veri alanına index işlemini uygulayalım. Bunun için öncelikle Name yazan kutuya KODINDEX adını verip sonrada STOKKODU alanını işaretlememiz yeterli olacaktır. İndexleme esnasında Primary, Enuque, ve IgnoreNulls alanlarının pasif olması yani seçili olmaması gerekmektedir. Bu işlemleri gerçekleştirip OK düğmesine basarsak oluşturduğumuz index dosyaları Table Structure ekranında görülecektir.(Şekil-12.9)

12.4.4. Tabloya Kayıt (Bilgi) Girmek

Oluşturulan tablolara bilgi girmek için Database window penceresinde yer alan tablolardan hangisine bilgi gireceksek o tablo üzerinde sağ tıklayıp açılan menüden Open seçeneği seçilir. Karşımıza aşağıdaki bilgi giriş formu gelir.



The screenshot shows a window titled "Dynaset:STOK" with a menu bar containing "Add", "Edit", "Delete", "Close", "Sort", "Filter", "Move", and "Find". Below the menu bar, there is a list of fields and their corresponding values:

Field Name:	Value (F4=Zoom)
STOKKODU:	1
LASTIKMARKASI:	LASSA
LASTIKTIPI:	YAZLIK
LASTIKEBADI:	57/123P/12R
ALISFIYATI:	1000000
KDV:	10
SATISFIYATI:	2000000
STOKADEDİ:	30

At the bottom of the form, there is a navigation bar with a left arrow, a right arrow, and the text "1/1".

Bu pencerede yer alan;

Add düğmesiyle yeni kayıt eklenir.

Edit düğmesi ile aktif kayıt üzerinde değişiklik yapılır.

Update düğmesiyle alanlara girilen bilgiler güncelleştirilir. (Update düğmesi Add veya Edit düğmesine basıldıktan sonra görünmektedir.)

Delete düğmesiyle aktif (görünen) kayıt silinir.

Close düğmesi ile bilgi giriş formu kapatılır.

Sort düğmesi ile istenilen alanlar sıralanabilir.

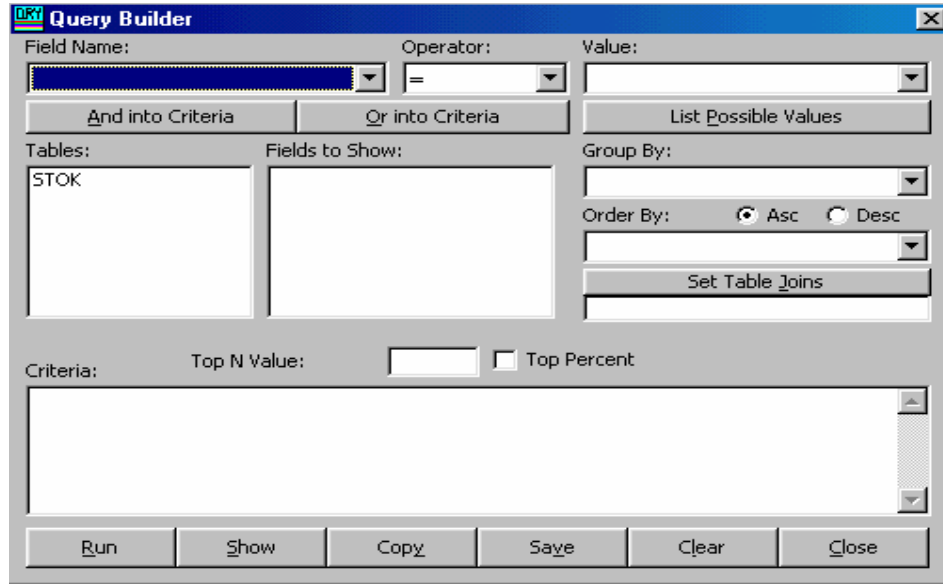
Filter düğmesi ile istenilen alanlara Süz işlemi uygulanabilir.

Move düğmesi yardımı kayıtlar taşınabilir.

Find düğmesi ile istenilen bir kayıt bulunabilir.

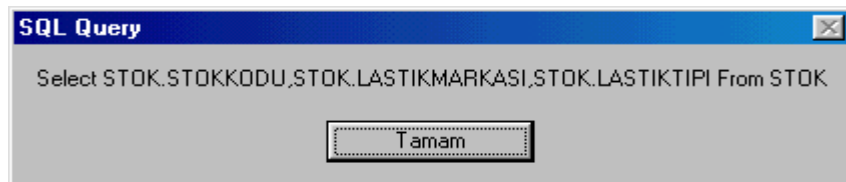
12.4.5. Query Builder (Sorgu Oluşturmak)

Sorgular sadece verilen kritere bağlı çalışan süzgeçlerdir. VisData programı ile oluşturduğumuz veri tablolarına ait sorgular oluşturabilirsiniz. Bunun için Utility-QueryBuilder menü seçeneği ile QueryBuilder penceresine gelinir.(Şekil-12.12)



Şekil-12.12. Query Builder Penceresi

Buradan table kutusunda yer alan STOK tablosunu seçelim. Bu seçildiğinde buna ait veri alanları Fields to shows penceresine aktarılacaktır.Eğer biz Fields to shows penceresinde yer alan alanlardan bazılarını seçer ve Show düğmesine basarsak karşımıza seçili alanları seçtiğimizi onaylatan aşağıdaki pencere gelecektir.



Bu işlem bittikten sonra Run düğmesine basarak yaptığınız sorgunun sonucunu görebilirsiniz. Sorgu işleminin amacı veri alanlarının sadece bazılarının görüntülenmesi ve sadece görüntülenen veri alanlarına bilgi girişi yapmaktır. Burada yapılan sorgunun sonucu aşağıda görüldüğü gibi olacaktır.

The screenshot shows a window titled "SQL Statement" with a menu bar containing "Add", "Edit", "Delete", "Close", "Sort", "Filter", "Move", and "Find". Below the menu bar, there is a "Field Name:" label and a "Value (F4=Zoom)" label. The "Field Name:" label is followed by a list of fields: "STOKKODU:", "LASTIKMARKASI:", and "LASTIKTIPI:". The "Value (F4=Zoom)" label is followed by a text box containing "2". To the right of the text box, there is a callout box with the text "Sorgu için seçilen alanlar". Below the text box, there are two more text boxes: one containing "GOODYEAR" and another containing "YAZLIK". At the bottom of the window, there is a status bar showing "1/2".

Şekil-12.13.Query Sonucu Oluşan Bilgi Formu

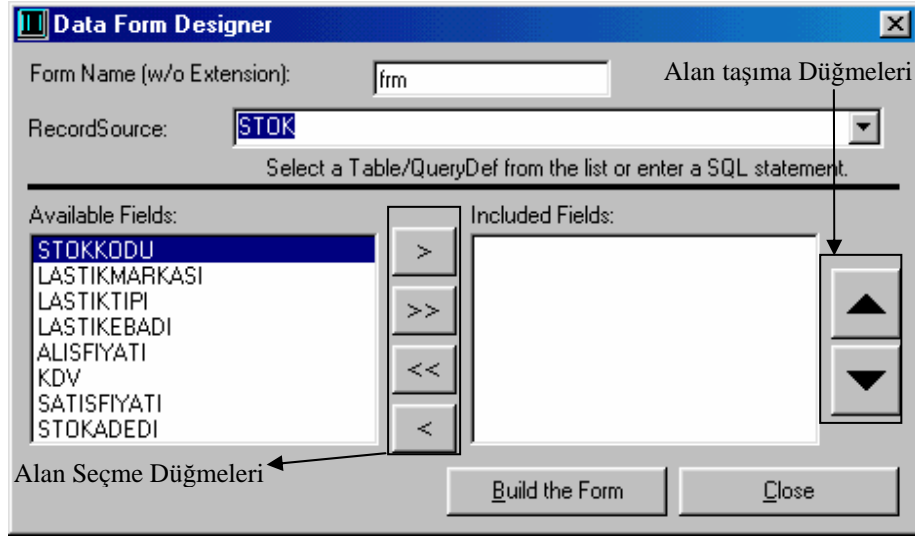
12.5. VB Ortamına Otomatik Veri Formu Aktarmak

VisData programı ile oluşturulan tablolar Visual Basic ortamına otomatik olarak aktarılabilir. Bu işlem ile hiçbir programlama bilmeden kolayca bir veri tabanı yapılabilir. Bu işlem için Utility-Data Form Desinger menü seçenekleri girilir ve aşağıda görülen pencere görüntüsü elde edilir.

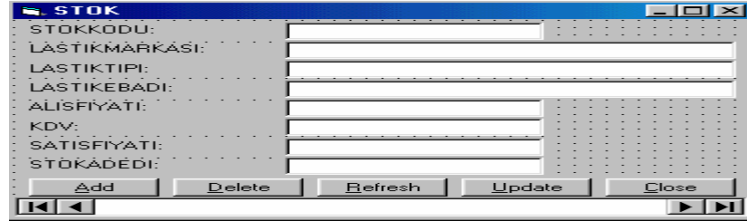
The screenshot shows a window titled "Data Form Designer". It has a "Form Name (w/o Extension):" label and a text box. Below it is a "RecordSource:" label and a dropdown menu. Below the dropdown menu is a label "Select a Table/QueryDef from the list or enter a SQL statement." Below this label are two text boxes: "Available Fields:" and "Included Fields:". Between these two text boxes are four buttons: ">", ">>", "<<", and "<". To the right of the "Included Fields:" text box are two buttons: "▲" and "▼". At the bottom of the window are two buttons: "Build the Form" and "Close".

Şekil-12.14. Data Form Desinger Penceresi

Form Name kutusuna veri tabanına ait olan alanların görüntüleneceği Form ismini verilir. RecordSource kutusuna ise hangi veri tabanı tablosu için form hazırlanacak ise o veri tablosunun ismi yazılır veya seçilir. Bu işlemlerden sonra veri alanlarını alan seçme düğmeleri ile seçerek form üzerine aktarma işi gerçekleştirilir. Seçilen alan Included Fields penceresinde görülür. Veri alanlarının yerlerini değiştirmek için şekilde görülen Alan taşıma düğmeleri kullanılır.



Veri alanı seçme işlemleri tamamlandıktan sonra Build the Form düğmesiyle form oluşturma işlemi başlatılır. Kısa bir süre sonra form oluşturulur.(Şekil-12.15).



Şekil-12.15 Tablonun VB Ortamındaki Görünüşü

Form oluşturulduğunda ver tabanı ile ilgili bazı işlemleri yapmak için hazır komutlar da form üzerinde yer alacaklardır. Form oluşturulduktan sonra program aynen Visual Basic(VB) derleyicisinde yazılmış gibi çalıştırılabilir. Form üzerinde bulunan düğmeler şu işlemleri sağlar.

Add düğmesi ile ver tabanı dosyasına yeni kayıt eklenir.

Delete düğmesi ile aktif kayıt silinir.

Refresh düğmesi ile alan içerikleri güncellenebilir.

Update düğmesi le girilen veriler güncelleştirilir.

Kayıt gezinti düğmesi ile kayıtlar arasında dolaşılabilir.

Close düğmesi ile programdan çıkılabilir.

Düğmelerin başlıkları tasarım anında iken daha önceki bölümlerde öğrendiğimiz özellikler yardımı değiştirilebilir.

VERİ ERİŞİM YÖNTEMLERİ

Nedir veri erişim yöntemi ? diye sorarsanız hemen cevap verelim. Veri erişim yöntemleri (Data Access objects) programlama aracılığıyla veri tabanı dosyalarına erişmek amacı ile kullanılan tekniklerdir. Tabi bu teknikleri kullanırken tanımlamamız gereken nesneler vardır. Bu nesnelere ise veri erişim nesneleri diyoruz.

Microsoft Visual Basic veri erişimi için çok sayıda arabirim geliştirmiştir. Şimdi sırası ile bu yöntemleri incelemeye başlayalım.

- VBSQL: SQL server için üretilmiş bir API-arabirimidir. Microsoft SQL Server'a erişim sağlar
- DAO(Date Access Objects): Microsoft Jet veri tabanlarına erişim geliştirilmiş ilk arabirimdir.
- RDO(Remote Access Objects): ODBC verilerine ulaşmak için kullacağımız diğer bir arabirimdir. Jet veritabanlarına erişim için kullanılmaz.
- ADO(Active X Data Objects): ADO,RDO ve DAO nun en gelişmiş biçimidir.
- ODBC: Çok sayıda ilişkisel veri tabanına erişim için kullanılır.
- OLE DB: ODBC daha gelişmiş biçimidir.
- RDS: Web verilerine erişim için kullanılır.

Şimdi nedir bu veri erişim yöntemleri diye kafanız karıştı gibi olmuştur. Aslında bu veri erişim yöntemleri arasında sürekli değişim yapılarak kullanılmazlar. Örnek olarak bir projenizi ADO temelli yaptığınızda diğer erişim yöntemlerine ihtiyacınız genelde olmaz. Biz Başlangıç olarak DAO veri erişimi ile işe başlamaktan yanayız. Daha sonraki yöntemimiz ile Visual Basic 6.0 ile gelen ADO teknolojisini i olacaktır. Şimdi DAO ile derslerimize başlıyoruz. İyi dersler

DAO VERİ ERİŞİM YÖNTEMİ

DAO programlama ile programlarımızı iki teknik ile yaparız. Bunlardan birincisi Data Kontrol nesnesi ile çalışma diğeri ise kendi tanımladığımız nesneler ile verileri erişimdir. Hangisi mi? diye sorarsanız. İleri seviyedeki programlarda zaten ikinci yöntemi kullanma zorunluluğumuz olacaktır. İmdi biz Data kontrol ile işimize devam edelim.

Derslerimizi bir örnek program üzerinde anlatmaya devam edeceğiz. Bu örnek programımız öğrenci takip programı olsun. Veri tabanımız ile ilgili bilgiler ise şunlardır;

Veri tabanı adı: ogrtakip.mdb

Tablolar: kisisel,devam

Kisisel tablosundaki alanlar ise;

ad	metin 20
soyad	metin 20
numara	metin 10
sinif	metin 10
tckimlikno	metin 20
cinsiyet	metin 10
dtarihi	tarih
toplamdevam	sayi

devam tablomuzdaki alanlar ise;

tarih	tarih
nedeni	metin 30
gunsay	sayi
numara	metin 10

Kırmızı alanlar ise ilişkili alanlarımız olacaktır. İlişkili alanların tanımı ileri derslerimizde göreceksiniz.

Devam edelim. Şimdi Visual Basic programımızda form tasarımını yapalım. Form tasarımımız başlangıç aşamasında aşağıdaki gibi oldu.

ÖĞRENCİ TAKİP PROGRAMI VER.1.0

ÖĞRENCİ KİŞİSEL BİLGİLERİ

TC. Kimlik Numarası: Öğrenci Numarası:

Öğrenci Adı Soyadı:

Öğrenci Sınıfı: Cinsiyeti:

Öğrenci Doğum Tarihi:

■ Data1

ÖĞRENCİ DEVAM BİLGİLERİ

Şimdi formumuz üzerinde ilk olarak kişisel tablomuzdaki veriler ile çalışarak devam edeceğiz. Form üzerine eklenecek text kutularını genelde size tavsiyemiz veri tabanınızdaki alanların isimleri ile yapılandırmanızdır kod ortamında hangi text kutusunun hangi alanı temsil ettiğini daha kolay tespit edebilmeniz için. Şimdi nasıl veri tabanına erişeceğiz ve verilerimiz text kutularına nasıl getirip göstereceğiz.

Data kontrolü formumuza ekledikten sonra ikinci aşamada data1 kontrolümüzü artık veri tabanımıza bağlamamız gerekir.

Properties - Data1

Data1 Data

Alphabetic Categorized

Data

BOFAction 0 - Move First

Connect Access

DatabaseName ...

DefaultCursorType 0 - DefaultCursor

DefaultType 2 - UseJet

EOFAction 0 - Move Last

Exclusive False

Options 0

ReadOnly False

RecordsetType 1 - Dynaset

RecordSource

- properties penceresindeki category kısmında data1 kontrolünün *databasename* özelliği tıklanır. Bu pencerede;
- ... düğmesi tıklanır. Açılan pencereden veri tabanı dosyanızın yeri belirtilir ve aç butonu tıklanır.
- Bir veri tabanında birden fazla tablo olacağından daha sonra hangi tablo ile çalışılacaksa o tablonun seçilmesi gerekir. (Örneğin bizim programda biz şu anda sadece kişisel tablosunu kullanıyoruz.) seçim işlemi için ise *recordsource* özelliği seçilir.
- Recordsource özelliği kişisel olarak atanır.

RecordSource

Font devam

Font kişisel

Misc NOTLAR

Artık data1 kontrolümüz veri tabanına bağlanmıştır. Bizim veri erişim yöntemimiz artık data1 nesnesidir. Peki programı çalıştırsak text kutularında bilgileri görebilir miyiz? Text kutuları şu aşamada data1 nesneli ile bağlantılı olmadığından sadece boş kutucuklardır. Şimdi sırada text kutuları nasıl verileri gösterecek o var; Text kutularını bağlamak için ise; (örneğin tckimlikno kutumuz) textboxların properties penceresinde category kısmında data bölümüne gelinir. Bu bölümde il olarak *datasource* ile data kaynağı belirtilir. Daha sonra işlemimiz bitmez. Çünkü data1 bağlamakla hangi alana bağlanacağını söylemiş olmayız. Tckimlik no alanının hangi alana bağlı olacağını ise data field özelliğinden belirtiriz. Ancak arkadaşlar bu işlemleri sırası ile yapmadığınız takdirde örneğin data1 kontrolü için recordsource vermediniz, bu durumda text kutularında data field kısmında alanları göremezsiniz. Yada datasource verilmeden alanları göremezsiniz. Sıralamaya dikkat ediniz.

Data

DataField

DataFormat tckno

DataMember adi

DataSource soyadi

DDE

LinkItem numara

LinkMode sinifi

dtarihi

cinsiyet

Properties - tckimlikno

tckimlikno TextBox

Alphabetic Categorized

RightToLeft False

ScrollBars 0 - None

TabIndex 3

TabStop True

Visible True

Data

DataField

DataFormat General

DataMember

DataSource

DDE

LinkItem

LinkMode 0 - None

LinkTimeout 50

LinkTopic

Font

Font MS Sans Serif

Şimdi bütün alanların datasource özellikleri data1 ve datafield özellikleride hangi alanı temsil ediyorsa ona uygun alan ile seçelim. Artık verilerimize ulaşabiliriz. Programımız çalıştığında veri tabanından kayıt girdi isek daha önceden kayıtlarımız gözükecektir. Aksi takdir yine kayıtlarımız göremeyeceğiz sizce?

ÖĞRENCİ TAKİP PROGRAMI VER.1.0

ÖĞRENCİ KİŞİSEL BİLGİLERİ

T.C. Kimlik Numarası: 50230076238 Öğrenci Numarası: 9198

Öğrenci Adı Soyadı: atalay ay

Öğrenci Sınıfı: a11c

Öğrenci Doğum Tarihi: 21.05.1985 Cinsiyeti:

Data1

ÖĞRENCİ DEVAM BİLGİLERİ

programımız çalıştırdık ve ekran text kutularında kayıtlarımızı görmüş olduk. Bu aşamada sadece kayıtlar arasında ulaşabiliriz. Şimdi işimiz bittimi sizce ?. kayıt işlemlerimize başlamanın zamanı geldi. Buyrun kayıt işlemlerine.

ÖĞRENCİ TAKİP PROGRAMI VER.1.0

ÖĞRENCİ KİŞİSEL BİLGİLERİ

T.C. Kimlik Numarası: Öğrenci Numarası:

Öğrenci Adı Soyadı: Soyadı:

Öğrenci Sınıfı: Cinsiyeti:

Öğrenci Doğum Tarihi: Cinsiyeti:

Data1

ÖĞRENCİ DEVAM BİLGİLERİ

YENİ KAYIT KAYIT GÜNCELLE KAYIT SİL GÜNCELLEME İPTAL KAYIT ARAMA

Şimdi formumuz gördüğümüz gibi command butonlar ekledik. Sırası ile bu butonların kod ortamlarını yazıyoruz. Kayıt işlemleri için arkadaşlar data1 nesnemizi kullanıyoruz. Artık verilere erişim için sürekli bu nesnemizi kullanmak durumundayız. Data1 nesnesi için kayıt işlemlerini temsil eden özelliğimiz ise recorset özelliğimizdir.

Data1 kontrolü özellikleri:

Data	
BOAction	0 - Move First
Connect	Access
DatabaseName	C:\Documents a
DefaultCursorType	0 - DefaultCurs
DefaultType	2 - UseJet
EOAction	0 - Move Last
Exclusive	False
Options	0
ReadOnly	False
RecordsetType	1 - Dynaset
RecordSource	kisisel

- databasename:** Data kontrolü için kullanılacak veri tabanı dosyasını seçeri.
- Recordsource:** Veri tabanı dosyası içerisindeki hangi tablo ile çalışılacağını belirler
- options:** Aşağıdaki kullanımım formatına göre şu değerleri alır.
Format: data1.options=dbforwardonly
Dbdenywrite 1 : çoklu otamlarda diğer kullanıcılar kayıt setindeki kayıtlara değiştiremezler
Dbdenyread 2 : çoklu otamlarda diğer kullanıcılar kayıt setindeki kayıtlara okuyamazlar
Dbreadonly 4 : kayıtlar değiştirilemez.
Dbappendonly 8 : Kayıt eklenebilir. Ancak var olan kayıtlar okunamaz.
DbInconsistent 16: Tüm Kayıtlar güncellenebilir.

- Readonly:** True yada false değerlerine göre sadece okunur olması özelliği verilir.
- EOAction:** Data kontrolü son kayıta geldiğinde ileri düğmeye basıldığında ne tür bir işlem yapılacak belirtir. Aşağıdaki değerlerden biri verilerek davranış belirtilir.
0 : son kayıta kalınır.
1: sonraki düğmesi pasif
2: yeni kayıt eklenir.

Arkadaşlar veri tabanında en önemli kurallardan biride veri tabanı açıldığında mutlaka işiniz bittiğinde kapatmanız önerilir. Aksi takdirde veri tabanında bozulmalar olabilir. Data1 kontrolü açıldığında örneğin programdan çıkıldığında data kontrolü de kapatılmalıdır.

```
Private Sub Form_Unload(Cancel As Integer)
Data1.Recordset.Close
End Sub
```

Data Kontrolünü Yenileme:

```
Private Sub Command2_Click()
Data1.Refresh
End Sub
```

Kayıt Arama:

Kayıt Arama işlemlerinde iki türlü arama vardır.

1. İndexli arama: index verilen alanlara göre arama işlemi yapılır. Veriler index dosyalarında arandığından hızlıdır. İndex dosyalarında arama işlemlerinde data1 *kontrolünün recordsettype özelliği mutlaka table olmalıdır.* Ancak indexli aramada like operatörü kullanılmaz.

```
Private Sub Command2_Click()
Data1.RecordsetType = 0
Data1.Refresh
Data1.Recordset.Seek "=", ogrno.Text
If Data1.Recordset.NoMatch = True Then
MsgBox "kayıt bulunamadı"
End If
End Sub
```

2. Find ile Arama yöntemi: indexli arama yönteminin aksine verilen ana tabloda aranır. Hız olarak daha yavaştır. Arama sonucundaki bilgiler sıralı değildir. Ancak like metodu ile detaylı arama yapabilirsiniz. Örneğin baş harfi f ile başlayan adlar gibi.

```
Private Sub Command2_Click()
Data1.RecordsetType = 1
Data1.Refresh
Data1.Recordset.Findfirst "tckimlikno=" & tckimlikno.Text & ""
If Data1.Recordset.NoMatch = True Then
MsgBox "kayıt bulunamadı"
End If
End Sub
```

Not find ile arama işlemlerinde format bilgisi olarak aşağıdaki format kullanılır.

Data1.recordset.find "tablo_alan_adı=değişken"

Uyarı: değişken bölümü aşağıdaki durumları alabilir.

*Eğer aradığınız alan bir karakter ise data1.recordset.findfirst "alanadı='& değişken &'"

*Eğer aradığınız alan bir sayısal alan ise data1.recordset.findfirst "alanadı=& değişken &"

*Eğer aradığınız alan tarih ise data1.recordset.findfirst "alanadı=#& değişken &#"

Eğer like kullanılacak ise data1.recordset.findfirst "alanadı like '& değişken &'" (bu bölümde like ile kullanılan * joker karakterdir. * değişkenin başına veya sonunda veya her ikisinde de kullanılabilir. Ayrıntılı bilgi için operatörler bölümüne bakınız.)

YENİ KAYIT	düzenle	ÖNCEKİ KAYIT	SONRAKİ KAYIT	İLK KAYIT	SON KAYIT	KAYIT SİL	TÜM DEVAM RAPORU	YEDEKLE
------------	---------	--------------	---------------	-----------	-----------	-----------	------------------	---------

KİŞİSEL BİLGİLERİ	DEVAMSIZLIK BİLGİLERİ	NOT BİLGİLERİ
--------------------------	-----------------------	---------------

ÖĞRENCİ KİŞİSEL BİLGİLERİ	
TCKNO.....:	50230076238
ADI.....:	atalay
SOYADI.....:	ay
SINIFI.....:	a11c
NUMARASI.....:	9198
SINIF ÖĞRETMENİ.....:	rahit akın
GELDİĞİ OKUL.....:	mehmet akif ersoy iöo
YABANCI DİLİ.....:	ingilizce
CİNSİYETİ.....:	SEÇİNİZ
DOĞUM TARİHİ.....:	21.05.1985
DOĞUM YERİ.....:	k.maraş
ANA ADI.....:	ipek
BABA ADI.....:	ali
ADRESİ.....:	namık kemal mahallesi 23. sokak no:11 kahraman maraş

yukarıdaki formda verilen programın kodlarına göre ;

Option Explicit
'nesnelerin türleri belirlendi'

Public dosya As Database
Public tablokisi As Recordset
Public tablodevam As Recordset
Public tablonor As Recordset

Private Sub Command1_Click()
sakla
Command1.Visible = False
Command2.Visible = False

End Sub

Private Sub Command2_Click()
Command1.Visible = False
Command2.Visible = False
sifirla
goster
End Sub

Private Sub Command3_Click()
duzenler
Command3.Visible = False
SSTab1.SetFocus
End Sub

Private Sub Form_Activate()
devam_puan_goster
End Sub


```
t_ogr.Text = tablokisi.Fields("sogr")
t_gelokul.Text = tablokisi.Fields("gelokul")
t_yabdil.Text = tablokisi.Fields("yabancidil")
t_anaadi.Text = tablokisi.Fields("anaadi")
t_babaadi.Text = tablokisi.Fields("babaadi")
t_dyeri.Text = tablokisi.Fields("dyeri")
t_dtarihi.Text = tablokisi.Fields("dtarihi")
t_adresi.Text = tablokisi.Fields("adresi")
combo1.Text = tablokisi.Fields("cinsiyet")
```

End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)

 Select Case Button.Key

 Case "onceki":

 once_kayit

 Case "YENI":

 sifirla

 tckimlikno.SetFocus

 Command1.Visible = True

 Command2.Visible = True

 Case "SONRAKI":

 sonra_kayit

 Case "duzenle":

 tckimlikno.SetFocus

 Command3.Visible = True

 Case "ILK":

 tablokisi.MoveFirst

 sifirla

 goster

 devam_puan_goster

 Case "SON":

 tablokisi.MoveLast

 sifirla

 goster

 devam_puan_goster

 Case "RAPOR":

 Case "YEDEKLE":

 YEDEKLEME.Show

 Case "sil":

 Dim soru

 soru = MsgBox("kayit silinsin mi?", vbYesNo + vbQuestion + vbDefaultButton2)

 If soru = 6 Then

 tablokisi.Delete

 sonra_kayit

 sifirla

 goster

```

        devam_puan_goster
    End If
End Select
End Sub

Public Sub once_kayit()
    If tablokisi.BOF Then
        MsgBox ("ilk kayıttasınız")
        tablokisi.MoveNext
        sifirla
        goster
        devam_puan_goster

    Else

        tablokisi.MovePrevious
        sifirla
        goster
        devam_puan_goster

    End If
End Sub
Public Sub sonra_kayit()

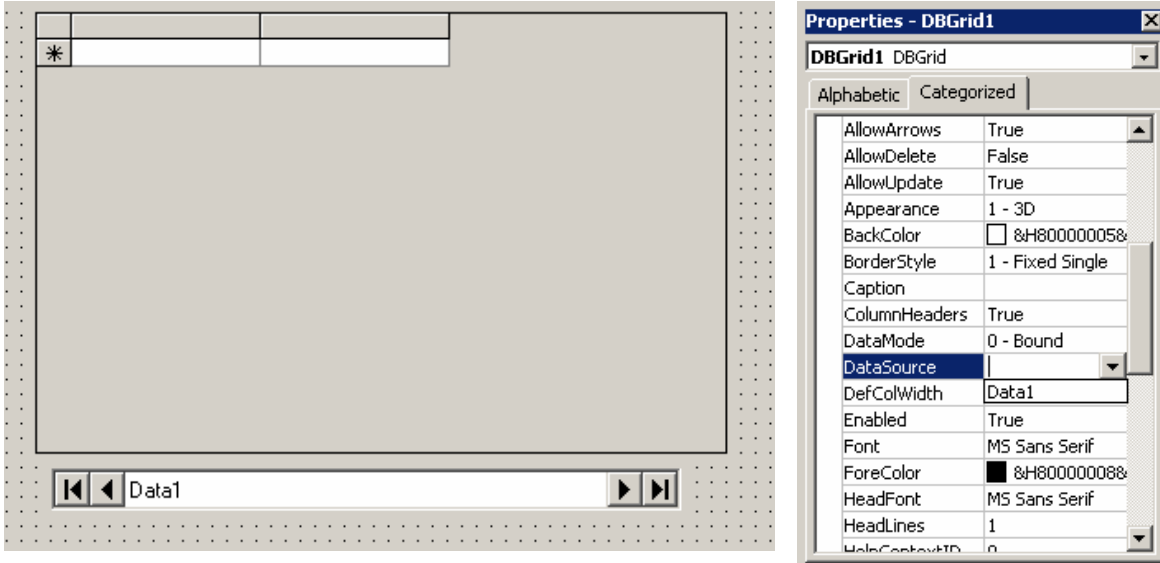
    If tablokisi.EOF Then
        MsgBox ("son kayıttasınız")
        tablokisi.MovePrevious
        sifirla
        goster
        devam_puan_goster
    Else
        tablokisi.MoveNext
        sifirla
        goster
        devam_puan_goster

    End If
End Sub
Public Sub duzenler()
    tablokisi.Edit
    tablokisi.Fields("tckno") = tckimlikno.Text
    tablokisi.Fields("adi") = t_adi.Text
    tablokisi.Fields("soyadi") = t_soyadi.Text
    tablokisi.Fields("sinifi") = t_sinifi.Text
    tablokisi.Fields("numara2") = t_numara.Text
    tablokisi.Fields("sogr") = t_ogr.Text
    tablokisi.Fields("gelokul") = t_gelokul.Text
    tablokisi.Fields("yabancidil") = t_yabdil.Text
    tablokisi.Fields("anaadi") = t_anaadi.Text
    tablokisi.Fields("babaadi") = t_babaadi.Text
    tablokisi.Fields("dyeri") = t_dyeri.Text
    tablokisi.Fields("dtarihi") = t_dtarihi.Text
    tablokisi.Fields("adresi") = t_adresi.Text
    tablokisi.Fields("cinsiyet") = combo1.Text
    tablokisi.Update
End Sub

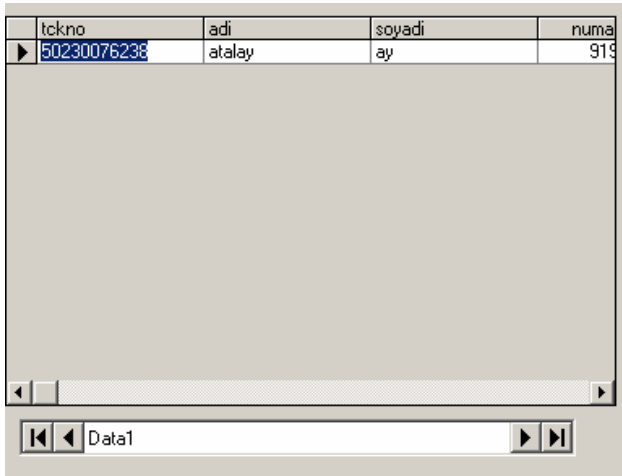
```

DBGRID KONTROLÜ

Dbgrid kontrolü toolbar'a eklemek için compenetslerden *Microsoft Databoundgrid controlü* eklemeniz gerekir. Kontrolü ekledikten sonra data nesnemize bağlamamız gerekir. Bu işlem için ise;



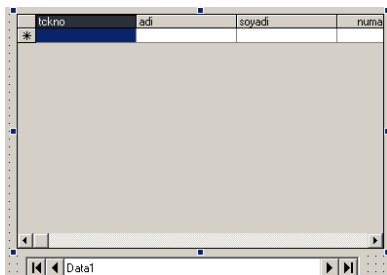
Dbgrid nesnesinin data source özelliği form üzerindeki data kontrolünüze eşitlenir. Örneğimizde data1 olarak bağlanmıştır. Eğer data1 nesnesi olmadan datasource seçeneği seçilir ise data1 nesnesini dbgrid özelliklerinde göremezsiniz. Bu aşamada programı çalıştırmanı dbgridin çalışmasını sağlayacaktır. Bütün alanlar dbgrid default ayarlarına göre dbgrid üzerine gelecektir.



Görüldüğü gibi ogrtakip.mdb veri tabanı dosyasındaki kisisel tablosuna ait tüm kayıtlar dbgrid bileşeni içinde yer aldılar. Şimdi kayıtlar arasında yön tuşlarını kullanarak dolaşmak mümkündür. Peki dbgrid kontrolü ile yeni kayıt girişi, istenilen sütunları gösterme gibi işlemler için ne yapmamız gerekir. Bu durumda artık dbgrid özelliklerini öğrenelim.

- Sadece İstenilen sütunları gösterme:**

Dbgrid kontrolü üzerinde sağ tıklanır. Bu bölümden retrieve fields seçilir. Bunu yaptığımızda tabloya ait tüm alan başlıkları dbgrid bileşenine gelir. Bundan sonraki işlem artık hangi sütun olmayacak gibi işlemlerdir. Bunun için yine dbgrid üzerinde sağ tıklanır. Bu bölümde edit seçeneği seçilir. Edit seçeneği ile artık dbgrid düzenleme moduna girmiştir. Bu moda iken herhangi bir sütun seçildiğinde;



Sağ tıklayarak aşağıdaki işlemlerden birisi yapılabilir.

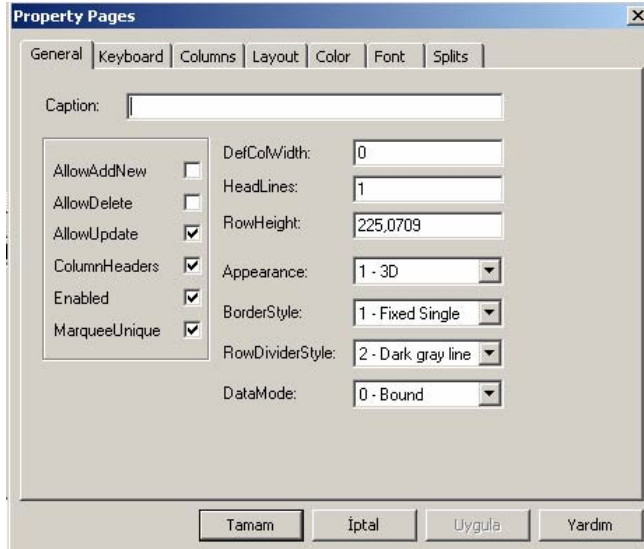
Delete: seçili sütunu siler

Insert :Seçili olan sütundan bir önceki sütuna yeni bir sütun ekler.

Append: en sonra yer alan sütunun yanına yeni bir sütun ekler.

Split: Tabloyu seçili yerden ikiye böler.

Clearfields: sütun başlıklarını siler.

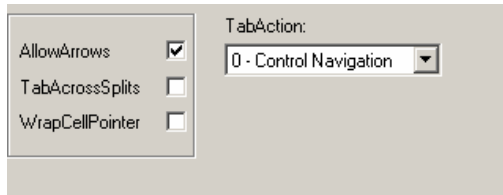


Şimdi sırası ile

1. General:

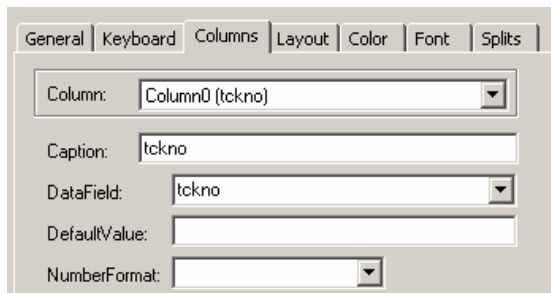
- Caption: Dbgrid kontrolü üzerine yazılacak olan başlığı belirtir.
- Allowaddnew: yeni kayıt girişi olanağını sağlar
- Allowupdate: Kayıt güncelleme işini sağlar
- Allowdelete: silme işine izin verir.
- ColoumnHeaders: Sütunları çalışma ortamında genişletme – daraltma iznini verir yada vermez.
- Enabled: Dbgrid kontrolü çalışma ortamında aktif yada pasif olur.
- Headlines: Sütun Başlığı genişliği
- Rowheight: satır yüksekliği

2. Klavye



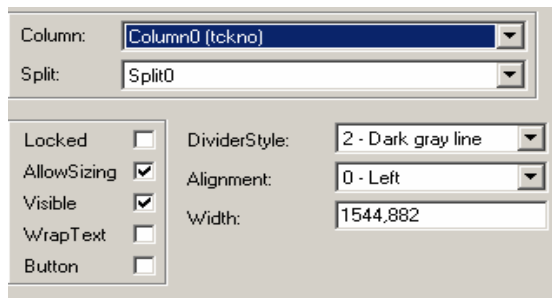
AllowArrows: işaretli ise yön tuşları ile kayıtlar arasında dolaşılır
TabAcrossSplits: dbgrid bölünen kısım içinde tab tuşunun kullanılmasını sağlar.
WrapCellPointer: hücreler arasında dolaşırken imleç otomatik olarak bir sonraki satırın ilk hücresine konumlanır.

3. Cloumns:



Column: sütun seçme bu bölümde veri alanları hangi sütunda yer almıştır o belirtilir.
Caption: Seçili kolonun başlığı belirtilir.
DataField: Seçili sütuna hangi alan atanacak belirtilir.
Default Value: Yeni kayıt eklendiğinde sütunun alacağı varsayılan değeri belirtir.
NumberFormat: Seçili sütunun alan türü belirtilir.

4. Layout



Locked: Seçili sütuna bilgi girişi yapılamaz
Allowsizing: sütun fare yardımı ile istenilen genişliğe getirilebilir.
Visible: sütun görünür yada gözükmez.
Wraptext: Hücreye yatay olarak sığmayan yazılar bir alt satırdan devam eder:
Driverstyle: sütun çizgilerinin biçimi ayarlarını.

5. Splits:
Locked: İşaretli ise ilgili bölüme split bilgi girişi yapılmaz
Allowfocus: ilgili boyut yeniden boyutlandırılabilir.
Scrollbars: kutusunda dbgrid bileşenine yatay ve dikey kaydırma çubukları eklenir.
Marguestyle: hücreler ile ilgili ayarlamalar yapılır.

DBGRID KONTROLÜ OLAY YORDAMLARI (EVENTS)

AfterColEdit	ButtonClick
AfterColUpdate	Change
AfterDelete	Click
AfterInsert	ColEdit
AfterUpdate	ColResize
BeforeColEdit	DblClick
BeforeColUpdate	DragDrop
BeforeDelete	DragOver
BeforeInsert	Error
BeforeUpdate	GotFocus
ButtonClick	HeadClick
Change	KeyDown

AfterColEdit(by Val colindex as integer):

Herhangi bir hücreye bilgi girişi tamamlandıktan sonra bu olay meydana gelir.

AfterColUpdate(by Val colindex as integer):

Dbgrid içerisinde bir verinin taşınması olayında meydana gelir.

AfterDelete

Herhangi bir kayıt silindikten sonra meydana gelir.

Afterinsert

Dbgrid bileşenine yeni bir kayıt eklendikten sonra meydana gelir

Afterupdate

Dbgrid içerisinde yapılan değişiklik veri tabanına yazıldıktan sonra meydana gelir.

ButtonClick

Layout kısmında sütunlara verilen buon seçeneği ile aşağı doğru bir ok eklenir. Buttonclick olayında ise bu aşağı doğru ok tıklandığında neler yapılacak ise kodlar eklenir.

Örnek:

```
Private Sub DBGrid1_ButtonClick(ByVal ColIndex As Integer)
```

```
If ColIndex = 3 Then
```

```
    DBGrid1.Columns(0) = "ali"
```

```
End If
```

```
End Sub
```

Yukarıdaki örnekte olduğu gibi dbgrid üzerindeki 4. sütunumuza buton eklenmiş ve buton tıklanığında 0. sütun ali olarak değiştirilmektedir.

Change

Dbgrid içerisinde herhangi bir değişiklik yapıldığında meydana gelir.

Click:

Dbgrid üzerinde herhangi bir yeri tıkladığınızda meydana gelir.

Colresize

Dbgrid içerisinde Sütun yeniden boyutlandırıldığında meydana gelir.

HeadClick

Dbgrid içerisinde sütun başlıklarından herhangi birisi tıklandığında meydana gelir.

Afterupdate

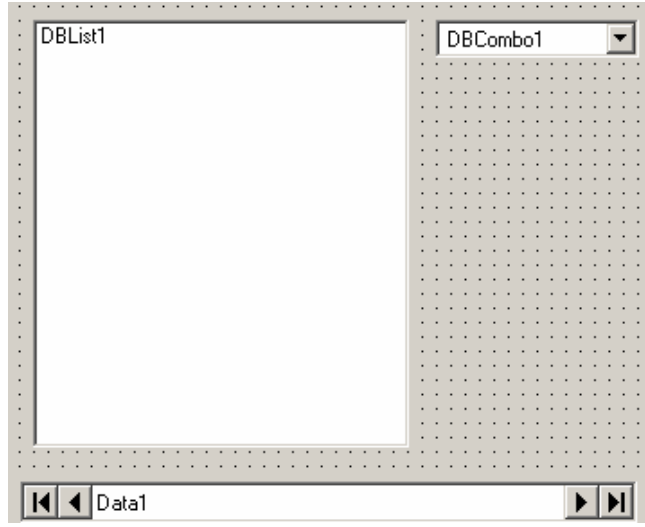
Dbgrid içerisinde yapılan değişiklik veri tabanına yazıldıktan sonra meydana gelir.

OnAddnew()

Yeni bir kayıt girişi olduğunda meydana gelir.

DBCOMBO KONTROLÜ

Dbgrid kontrolü toolbar'a eklemek için compenetslerden *Microsoft Databoundlist controlü* eklemeniz gerekir.



Özellikleri:

Rowsource, Listfield:

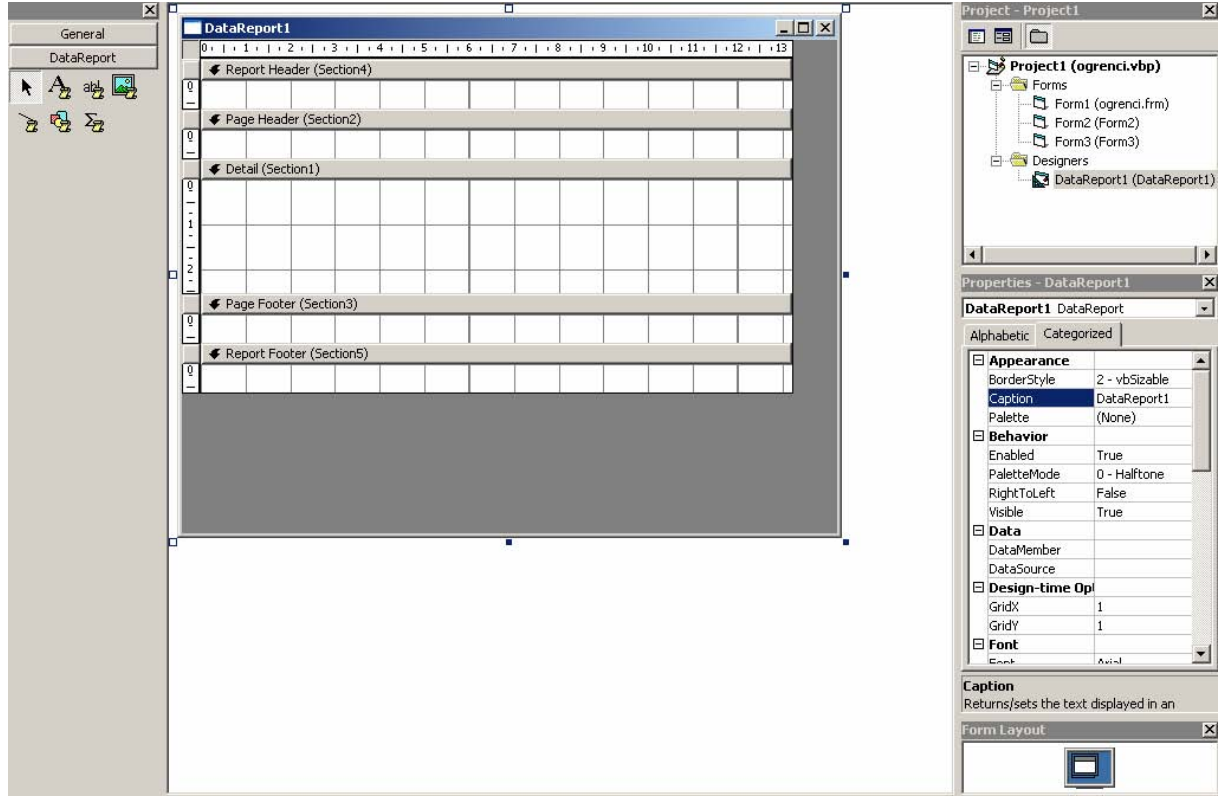
Listeleme işleminin hangi veri tabanından alınacağını belirler.

Datasource-Datafield:

Liste üzerinde seçili bir elemanın herhangi bir alana yazılmasını isterseniz Datasource-datafield özellikleri seçilir.

RAPORLAR İLE ÇALIŞMAK

Rapor oluşturmak bir çok ticari uygulamanın en Önemli özelliklerinden biridir. İyi bir programı ön plana çıkaran özelliklerin başında şüphesiz iyi bir raporlama tekniği yatmaktadır. Diğer vb versiyonlarından farklı Olarak raporlar VB 6.0 programı ile entegre edilebilecek hale getirilmiştir. Aşağıda raporun tasarım anındaki görüntüsü yer almaktadır.)Görüntüyü almak için Project menüsünden Add Data Report seçeneğini vermelisiniz.)



Şekil-13.1 Rapor Dosyasının Tasarım Görüntüsü



Data Report Designer (DRD) Özellikleri

Data Report Designer' in en önemli özelliği Drag and Drop (Sürükle ve Bırak) olarak bilinen özelliktir. Data Environment Designer (DED) üzerindeki veriler sürüklenip bırakılması ile otomatik olarak Text kutuları oluşmaktadır. Aynı zamanda da bu alanlara ait DataMember ve DataField özellikleri de otomatik ayarlanmaktadır. (Data Environment Designer' i görmek için Project*More ActiveX Designer*Data Environment seçeneğini vermelisiniz.)

Data Report Designer (DRD) Kontrolleri

Rapor tasarımında kullanılan elemanlar Şekil-13.1' de görülmektedir. Bu kontrollerin bulunduğu alana Data Report Toolbox adı verilir. Toolbox elemanları şunlardır.

	Rpt Label :Rapor başlıkları ve sabit ifadeler için kullanılan elemandır. Tasarım anında düzenlenebilir.
	Rpt TextBox : Bu bileşen çalışma zamanında ve sadece Text formatındaki veri alanlarının içeriğini görüntüler. Veri girişi yapılamaz.
	Rpt Image :Rapora resim eklemek için kullanılır. Standart Image kontrolüne benzer.
	Rpt Line : Rapor üzerinde değişik çizgiler (yatay, dikey, çapraz) çizmek için kullanılır.

	RptShape: Rapor üzerinde değişik geometrik şekiller çizmek için kullanılır. Bunlar arasında kare, dikdörtgen, oval, çember sayılabilir.
	RptFunction: Bu bileşen raporun sadece Footer bölümünde yer alabilir. Bununla rapor üzerinde yer alan veriler üzerinde işlem yapılabilir. (Toplama vb.)

Bir Raporun Yapısı

Bir rapor aşağıdaki bölümlere sahiptir.

- Report Header (Rapor Başlığı)
- Page Header (Sayfa Başlığı)
- Group Header (Grup Başlığı)
- Group Footer (Grup Sonu)
- Details (Gövde Bölümü)
- Page Footer (Sayfa Sonu)
- Report Footer (Rapor Sonu)

Şimdi bu bölümleri inceleyelim.

Report Header

Her raporun başlığında çıkacak olan ifade burada yazılır. Buraya genellikle hangi alanda döküm yapılacaksa o isim verilir. Normalde bu başlık tüm sayfalarda görülmektedir. Eğer sadece birinci sayfada görülüp diğer sayfalarda görülmesini istemiyorsak Report Header seçili iken Properties penceresinden ForcePageBreak özelliğinin değerini RptPageBreakAfter yapılmalıdır. (Bu alana başlık yazmak için önce RptLabel kontrolünü bu alana yerleştirmeliyiz. Rapor üzerinde bir kontrol eklemek için ekleme yapacağımız alanda sağ tıklayıp açılan menüden Insert Control seçeneği seçip istediğimiz kontrolü ekleyebiliriz.)

Page Header

Her sayfada çıkmasını istediğimiz ifadeler bu bölümde yer almalıdır yani veri tabanı dosyama ait kayıtların bulunduğu sahaların isimleri bu bölümde yer almalıdır.

Group Header / Footer

Burası DRD' nin tekrarlanan bölümlerini içerir. Her bir grup başı, grup sonu ile eşittir. Örneğin bir müşterinin tüm alımlarını ve toplam borcunu bulmak için kullanılabilir. (Eğer bu bölüm görünmüyor ise aynen kontrol ekler gibi sağ tıklayıp açılan menüden Group Header veya Group Footer seçilmelidir.)

Details

Bir raporun temel gövdesini burası oluşturur. Veri tabanı dosyamızda bulunan kayıtlar bu bölümde yer almalıdırlar. Buradaki alanlar RptTextBox ile oluşturulmalı. Bu TextBox içerikleri Properties penceresinde yer alan DataField ve DataSource değerlerinin seçilmesi ile belirlenmektedir.

Page Footer

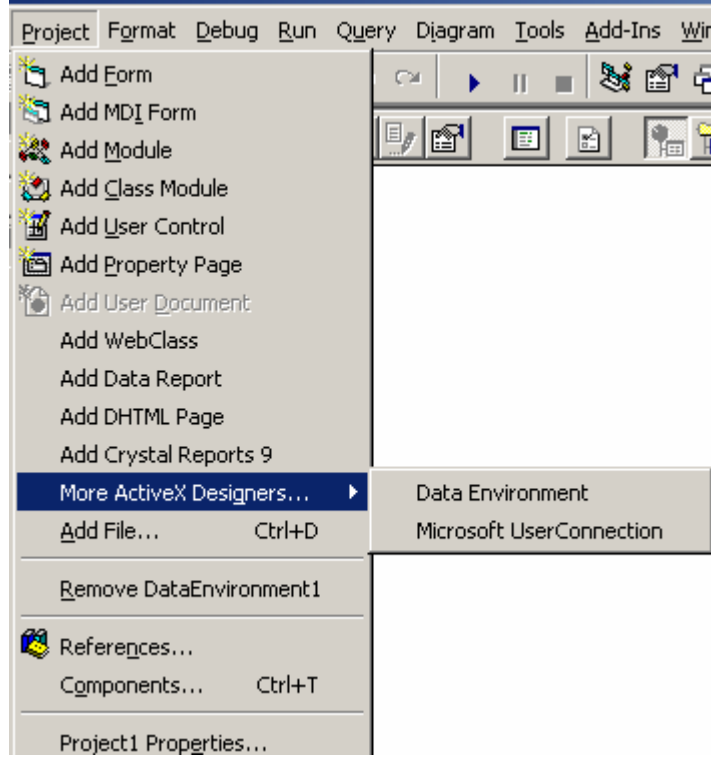
Rapora ait her sayfanın alt kısmında çıkacak olan bilgiler burada yer alır. Örneğin sayfa numarası, tarih, saat, şirket ismi bu alan sayesinde eklenebilir.

Report Footer

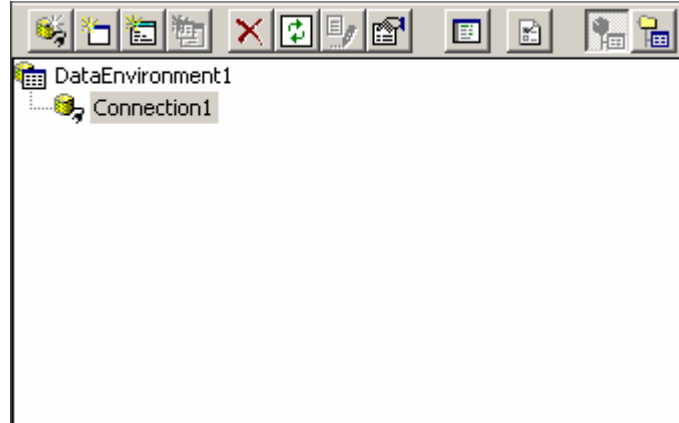
Raporun sonunda yer alması gereken bilgiler burada yer alır. Rapor özeti, toplam sayfa sayısı gibi bilgiler bu kısımda yer almalıdır.

Bir Rapor Oluřturmak

Yeni bir proje bařlatalım ve Project menüsünden More ActiveX Control seçeneğinden Data Environment seçeneğini tıklayalım. Bu seçenek ile veri tabanı tasarımcısı projeye eklenecektir.

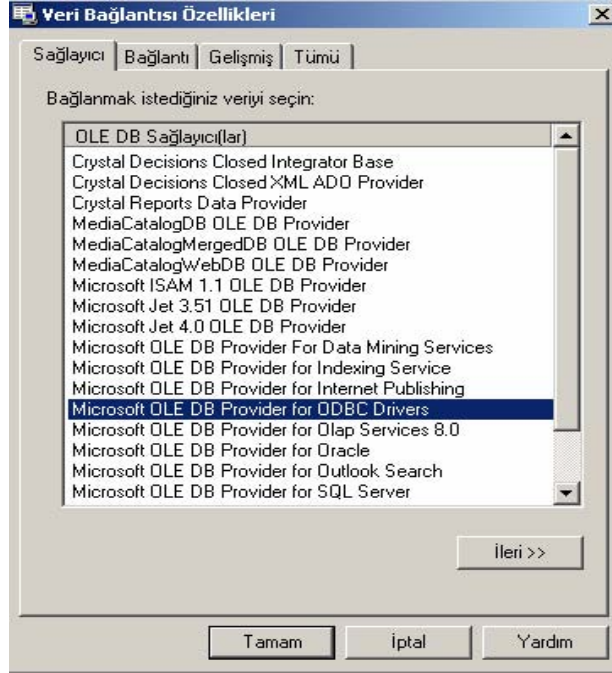


Data Environment' in Eklenmesi

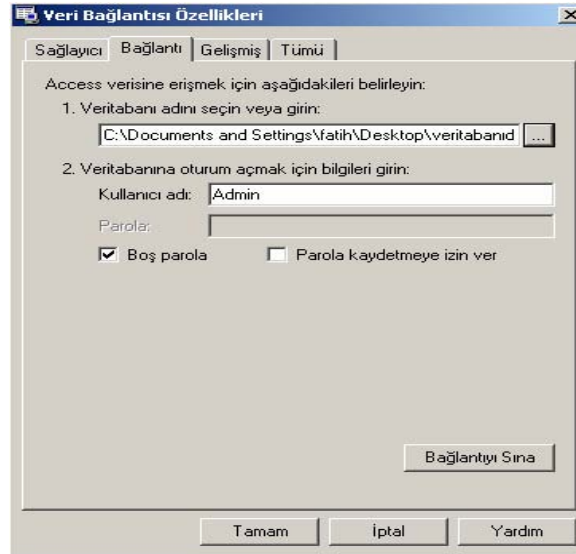


Data Environment Penceresi

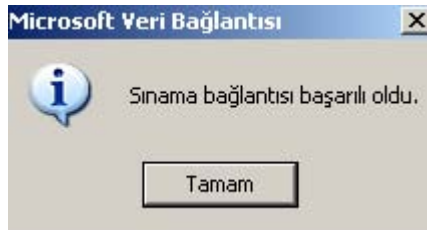
Data Environment eklendikten sonra pencerede yer alan Connection üzerinde sağ tıklayarak sağığıdaki ekran görüntüsüne ulaşalım.




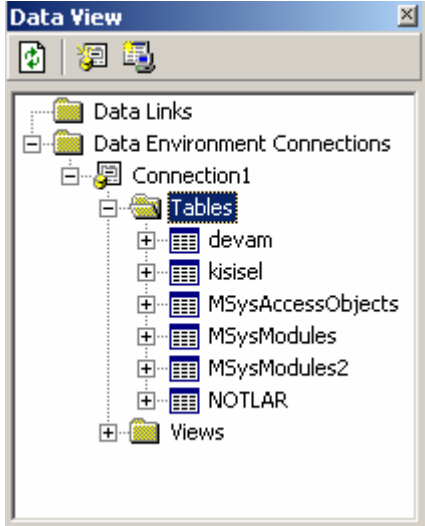
Bu pencerede veri bağlantımızın özelliklerini belirleriz. Genelde bu alandan ilk seçenek olan Microsoft Jet 3.51 OLE DB Provider seçeneği seçilir. Seçim işlemi bittikten sonra İleri butonuna basarak bir sonraki aşamaya geçelim. Bir sonraki aşamada veri bağlantısını yapacağım dosyayı belirleyeceğiz. Veri tabanı dosyasını belirlemek için pencerede yer alan ... butonuna basmalıyım. Bu butona basarak veri tabanı dosyama olan bağlantı işlemlerini tamamlamış olurum.



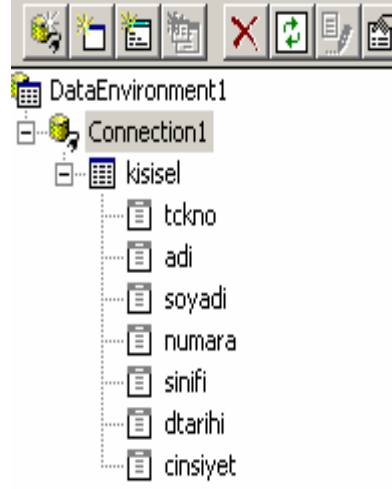
Bağlantı işlemi bittikten sonra Bağlantıyı Sına düğmesi ile yaptığımız bağlantıyı kontrol edebiliriz. Eğer bağlantıyı doğru yaptıysak aşağıdaki mesaj penceresi karşımıza çıkacaktır.



Bağlantı işlemi tamamlandıktan sonra bağlantı yaptığımız veri tabanı dosyasına ait tablolar ve bu tablolara ait veri alanları Data View görülecektir. Burada yer alan tablolardan hangisi ile çalışacaksak o tabloyu sürükleyip Data Environment penceresine yer alan Connection üzerine bırakıyoruz ve tablomozo Data Environment penceresinde görüyoruz.(Data View penceresine ulaşmak için araç çubuğunda bulunan  ikona basmamız yeterli olacaktır.)

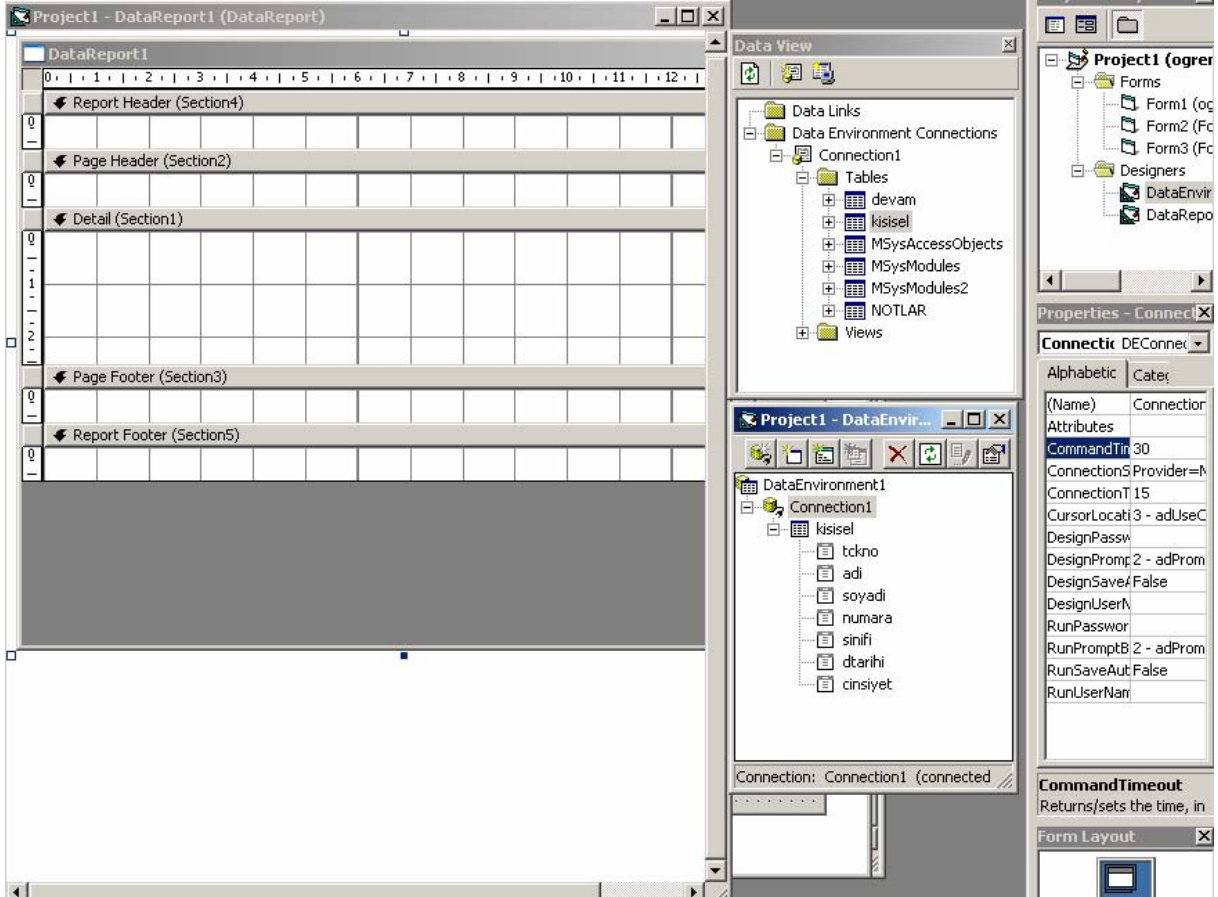


Data View Penceresi

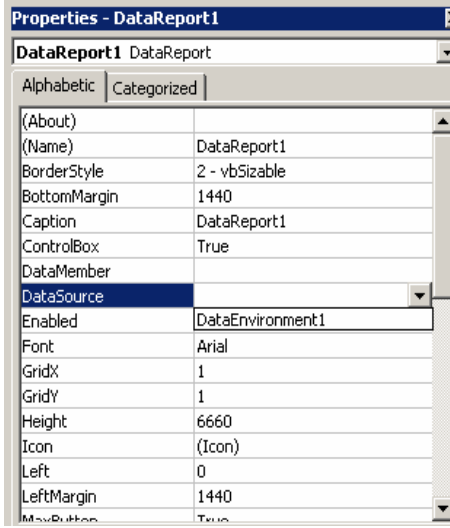


Data Environment Penceresi

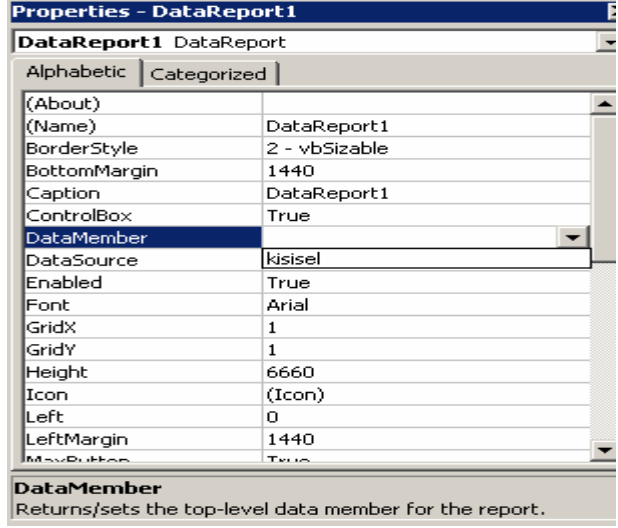
Data Environment tasarımcısını oluşturduktan sonra Project menüsünden Add Data Report seçeneğini tıklayarak raporumuza bir form, Data Report Designer (DRD) ekleyelim. DDR eklendikten sonra önceki konularda anlattığımız şekilde dizayn edilerek gerekli değişiklikler uygulanmalıdır. Yapılan değişikliklerden sonra projemizin son görüntüsü aşağıdaki gibi olacaktır.



Veri tabanında bulunan tabloların içeriklerini göstermek için Data Report' un üç önemli özelliğini kullanacağız. Bunlar DataSource ve DataMember özellikleridir. Data Report seçili iken Properties penceresinde DataSource özelliğine bağlantı kurduğumuz Data Environment adını (burada ben Data Environmet1' i kullandım.) yazmalı veya seçmeliyiz. DataMember özelliğine ise bağlantı kurduğumuz tablonun adını yazmalıyız. (Ben burada Stok tablosuna bağlantı kurdum.)

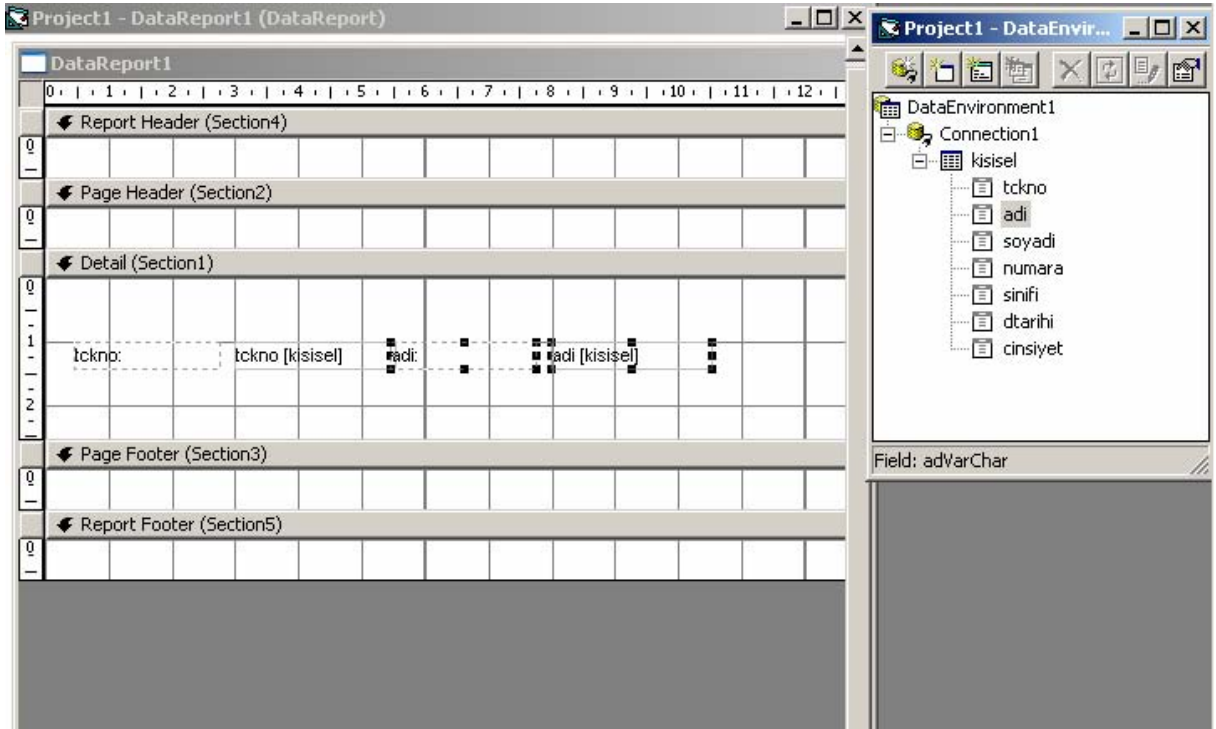


DataSource Değeri



DataMember Değeri

Bundan sonra yapılacak iş ise Data Environment penceresindeki tabloya ait alanları fare ile sürükleyip rapor üzerindeki Detail bölümüne bırakmaktır



Raporumuz üzerinde gerekli deęişiklikler ve ayarlamalar yapıldıktan sonra son görünümü şu şekilde olacaktır.

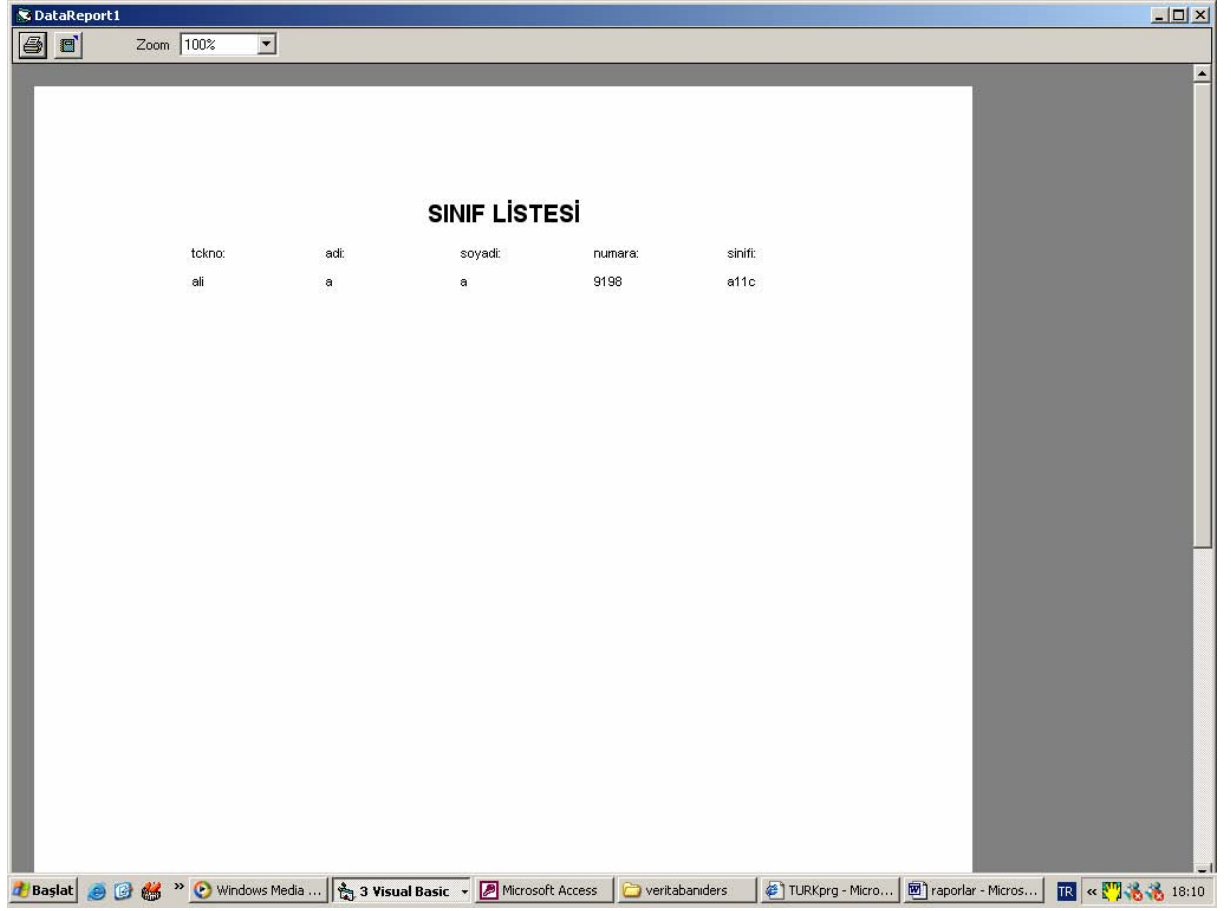
DataReport1															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Report Header (Section4)															
0	SINIF LİSTESİ														
1	Page Header (Section2)														
0	İtkno:		adi:		soyadi:		numara:		sinifi:						
1	Detail (Section1)														
0	İtkno [kisisel]		adi [kisisel]		soyadi [kisisel]		numara [kisisel]		sinifi [kisisel]						
1	Page Footer (Section3)														
0	Report Footer (Section5)														
1															

Raporumuzda görülen %T ve %D saat ve tarihi, %p sayfa numarasını göstermektedir.(Rapor dizayn edilirken sayfa büyüklüğüne dikkat etmeliyiz.)



Rapor hazırlandıktan sonra raporu kullanıcıya göstermek için projemiz dahilindeki forma "RAPOR" Caption' ılı bir CommandButton ekleyelim ve Click yordamına aşağıdaki program kodunu yazalım

```
Private Sub Command1_Click( )  
DataReport1.Show ' Hazırlanan DataRepot1' i ekranda gösterir.  
End Sub
```

Bu kod projeye dahil edildikten sonra proje çalıştırılıp RAPOR butonuna basılırsa aşağıdaki gibi bir ekran görüntüsünü elde ederiz.



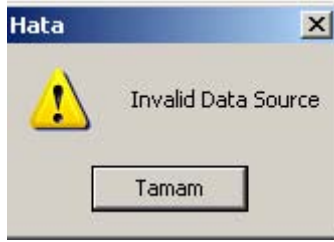
Raporumuz normal olarak çalışmaktadır. Rapor ekranında yer alan Zoom kutusu ile rapor belli oranlarda %' ilk olarak büyütölüp küçültülebilir. Ayrıca raporumuz üzerinde yer alan diğer simgelerde şu anlama gelmektedir.

	Raporu yazdırmak için kullanılır.
	Raporumuzun verilerini HTLM formatında kaydetme olanağı verir.

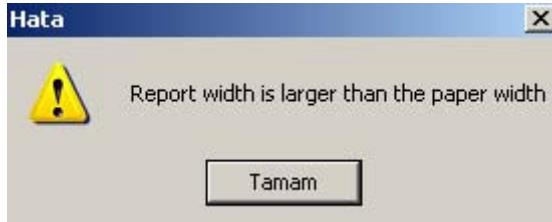
DATA REPORT HATALARI



Data Report Seçeneğinde Data Memberi seçmediniz.



data report seçeneğinde data source seçilmedi.



rapor genişliği sayfa genişliğini taşı. Raporunuzu daraltmanız gerekir.

DAO İLE SQL KOMUTLARI

Sql komutlarını artık veri tabanında kullanmaktan başka çaremiz yok arkadaşlar. Şimdi bu sql derslerimize başlayalım.

SQL-1 SELECT



En basit SQL cümlecığı oluşturmak için **SELECT** sözcüğü kullanılmaktadır.

Bu sözcük database içerisinde istediğin herhangi bir table'da ki kayıtları senin kullanımına açacaktır.

Yazılışı:

SELECT (virgüllerle ayrılmış alan adları veya * kullanılır) **FROM** Tabloismi

ÖRNEK

-- Musteri tablosunda ki tüm alanlar geri döndürülür.

SELECT * FROM Musteri

SQL-2 Where cümlecığı kullanmak



Where kullanımı ver tablolarını sorgulayarak geriye sınırlı sayıda kayıt döndürmenizi sağlar.

Mesela: müşteri kayıtlarını içeren bir veri tablomuz var ve siz belirli müşterilerin bilgilerini getirmek istiyorsunuz O zaman where cümlecığını Select ifadenize eklemelisiniz:

SELECT * FROM Musteri **WHERE** SoyAdi = 'Yorulmaz'

Bu ifade müşteri tablomuzda ki tüm kayıtlarda soyadı Yorulmaz olanları bulur ve geriye döndürür.

Eğer tüm kayıtlardan soyadı sadece 'C' harfi ile başlayanları bulmak istiyorsanız:

SELECT * FROM Musteri **WHERE** SoyAdi **LIKE** 'C%'

Bu yazılışın üstekinden farkı eklenen **LIKE** anahtar kelimesi ve % işaretidir.

% işareti joker karakteridir ve **LIKE** anahtar kelimesine beni gördüğün zaman ne olursa olsun farketmez manasındadır.

Yukarıdaki Select ifadesinde sadece Soyadı C ile başlayanlar geri döndürülmektedir.

SQL-3 Insert ile Kayıt Ekleme



Database'den nasıl kayıtlar seçilerek uygulamamıza aktarıldığını gördük. Şimdi de veritabanına nasıl kayıt ekleneceğini öğrenelim.

Insert ifadesi şöyle kullanılır;

INSERT INTO Tabloismi (Alanadı1, Alanadı2, Alanadı3)
VALUES ('Değer1','Değer2','Değer3')

Tabloismi veritabanında ki tablonun adıdır.

Bu tabloya kayıt eklenmek istendiğinde tablo içerisinde ki Alanadı1, 2, alanları birbirini karşılayan values (değer1, 2, 3, ...) değerleri ifade içerisine sırayla yazılmalıdır.

ÖRNEK

'-- Aşağıda ki kod Musteri tablosuna, VALUES bilgileri ve bu verileri karşılayan alan adları ile bir kayıt ekler.
INSERT INTO Musteri (Adi,SoyAdi,Cinsiyeti) VALUES ('Tamer','Yorulmaz','Erkek')

SQL-4 Update ile Kayıt Güncelleme



Update ifadesinin yazılışı şöyle:

UPDATE TabloAdı SET AlanAdı1=Deger1 WHERE AlanAdı2 = Deger2

Bu örnek yazılışı farklı şekilde de düşünebiliriz:

Eğer Where koşul cümlesini kullanmayacaksak tabloda ki tüm kayıtlar update güncelleme işlemine tabi tutulacaktır.

AlanAdı1 ve AlanAdı2 aynı kolon olabilir. Deger1 ve Deger2 string için " işareti içerisine alınır sayı için ise tırnak kullanılmaz.

ÖRNEK

'-- Bu Update ifadesi SoyAdı "Yorulmaz" olan müşterilerin adını "Ali" olarak günceller.
UPDATE Musteri SET Adi = 'Tamer' WHERE SoyAdi = 'Yorulmaz'

SQL-5 Delete Kayıt Silme



Delete ifadesinin yazım şekli:

DELETE * FROM TabloAdı WHERE AlanAdı = Deger

Kullanımı SELECT ifadesinde olduğu gibidir.

Kayıtları silmenin en iyi yolu normal SELECT ifadesi gibi kullanarak doğru kayıtları seçtikten sonra SELECT kelimesi yerine DELETE kelimesini yazmaktır.

ÖRNEK

'-- Soyadı Yorulmaz olan müşterileri siler.
DELETE * FROM Musteri WHERE SoyAdi = 'Yorulmaz'

|

SQL-6 GROUP BY



ASP dosyalarımızda Veritabanımızdaki kayıtları listelediğimiz bölümleri GROUP BY komutu ile istediğimiz kategoride gruplaya biliriz.

Örneğin;

dbornek adlı Veritabanımızın içerisinde tblornek isimli tablomuz ve bunun içerisinde Ad, Soyad ve Yaş alanları olsun. Kayıtları Adlarına göre gruplamak istediğimizde aşağıdaki gibi bir komut cümlecisi yazabiliriz.

SELECT * FROM tblornek GROUP BY Ad;

SQL-7 BETWEEN...AND



Bu SQL ifadesini bizim belirleyeceğimiz aralığa göre kayıtlarımızı süzmek için kullanabiliriz.

Örneğin;

dbornek adlı Veritabanımızın içerisinde tblornek isimli tablomuz ve bunun içerisinde urun, fiyat ve tarih alanları olsun. Kayıtlar arasından belirlediğimiz tarih arasındaki kayıtları seçebilmek için bir komut cümlecisi yazabiliriz.

SQL-8 LIKE



Tablolardan Seçtiğimiz kayıtları sıralamak için kullanılır.

Örneğin: `SELECT * FROM tblornek WHERE ad LIKE 'E%';`

Bu örneğimiz içerisindeki % işareti E harfinden sonra gelen harflerin ve uzunluğunun önemsiz olduğunu gösterir. Yani tblornek tablosu içerisindeki adı E ile başlayan ve uzunluğu önemli olmayan tüm adları seçmiş oluruz. Buna göre tablomuz aşağıdaki gibi oluşur.

Örneğin: `SELECT * FROM tblornek WHERE ad LIKE '%M?N';`

Bu Örneğimizde kullanılan ? işareti bir karakter yerine geçer. Bu karakterin türü önemli değildir. O halde SQL cümlemizi açarsak; tblornek tablosu içerisindeki ad alanları içerisinde, önceki karakterleri önemli olmayan son harfi N, ondan öncesinde herhangi bir karakter kullanılan ve onun öncesinde M harfi olan kayıtları bana göster. (Üff amma da uzunmuş yahu!). tblornek tablomuzu hatırlarsak buna uyan kayıt hangisidir? Elbetteki Egemen kaydı değil mi? Yani bunun sonucunda tablomuzun yeni hali yine yukarıdaki gibi olur, değişmez.

SQL-9 COUNT



Bir tabloda seçili alan içerisindeki değerleri sayar. Bunu özellikle siteniz için yapacağınız anket uygulamalarında, anket için verilen oyların sayımında kullanabilirsiniz.

Bir tabloda seçili alan içerisindeki değerleri sayar. Bunu özellikle siteniz için yapacağınız anket uygulamalarında, anket için verilen oyların sayımında kullanabilirsiniz.

Örnek-1: `SELECT COUNT(evet) AS ToplamEvet FROM tblanket;`

Bu Örneğimizde yazımıza girişte bahsettiğim gibi bir anket uygulamamız olsun ve bu anket uygulamamızın da tablosunun isminin tblanket olduğunu varsayalım. Ayrıca bu örneğimizin içerisinde de AS komutunun kullanımını öğrenmiş olalım. Şimdi gelelim SQL cümlemizin tercümesine. tblanket tablosu içerisinde evet alanındaki değerleri topla, AS komutu ile oluşturulan ToplamEvet değişkeninin içerisine aktar. Örneğin Anket uygulamamızın konusu olarak evet ve hayır seçenekleri bulunuyorsa, COUNT ve AS komutlarını kullanarak ne kadar oy verildiğini bulabilirsiniz. Buna göre aşağıdaki tablomuzu bir inceleyelim.

id	evet	hayır
--	---	----
1	1	
2	1	
3		1
4	1	
5	1	
6		1

Anket örneğimizde Anket Formundan gelen value(değer) ların 1 ifade ettiğini varsayarak, SQL cümlemizi hem evet, hem hayır için kullanırsak sonuçta karşımıza ToplamEvet in değeri 4, ToplamHayır in değeri 2 olarak çıkacaktır.

SQL-10 MAX



Tablo içerisinde seçilen sayısal bir alan değerinin en büyüğünü bulur.

Örnek-1: `SELECT MAX(yas) FROM tblornek GROUP BY ad;`

tblornek adında bir tablomuz olsun, yazmış olduğumuz SQL cümlesinde tblornek tablosundaki yas alanında, yaşı en büyük olanı seç. Tabi bir de GROUP BY komutunu kullandık. Bunun amacıda; tblornek tablosundaki yas alanında, yaşı en büyük olanların sayısı fazla ise, onları ad larına göre grupla. Bu na göre tablomuz nasıl bir görüntü alır? Aşağıda...

tblornek tablosu:

id	ad	soyad	Web Sitesi	yas
4	Tahsin	Özadam	www.icsehguc.net	23
1	Egemen	Mede	www.programci-tr.com	22
3	Yasin	Efe	www.soruvecevap.com	20

Sql cümlemizden sonra

id	ad	soyad	Web Sitesi	yas
4	Tahsin	Özadam	www.icsehguc.net	23

SQL-11 MIN



Tablo içerisinde seçilen sayısal bir alan değerin en küçüğünü bulur.

Örnek-1: `SELECT MIN(yas) FROM tblornek GROUP BY ad;`

tblornek adında bir tablomuz olsun, yazmış olduğumuz SQL cümlesinde tblornek tablosundaki yas alanında, yaşı en küçük olanı seç. Tabi bir de GROUP BY komutunu kullandık. Bunun amacıda; tblornek tablosundaki yas alanında, yaşı en küçük olanların sayısı fazla ise, onları ad larına göre grupla. Bu na göre tablomuz nasıl bir görüntü alır? Aşağıda...

tblornek tablosu:

id	ad	soyad	Web Sitesi	yas
4	Tahsin	Özadam	www.icsehguc.net	23
1	Egemen	Mede	www.programci-tr.com	22
3	Yasin	Efe	www.soruvecevap.com	20

Sql cümlemizden sonra

id	ad	soyad	Web Sitesi	yas
3	Yasin	Efe	www.soruvecevap.com	20

SQL ile Top List Mantığı



Bir veri tabanından değeri en yüksek belirli sayıda kayıt okutma:

Bir veri tabanı düşünün, elinizde binlerce kayıt var. Ama siz, hit'i en yüksek ilk 10 kaydı okutup ötesi ile muhatap olmak istemiyorsunuz;

İşte böyle bir durumda, şu SQL cümleciği yardımımıza yetişiyor:

`SELECT TOP 10 * FROM tblVeriTablom WHERE grup=5 ORDER BY hit DESC, id DESC;`

Dikkat edilmesi gereken nokta, bu cümlecik ile Veriler sıralandığında, 10 kayıt seçilmesi. Ancak kesin sonuç, seçilen kayıtların ilk 10'unun diğer kayıtlardan bariz olarak ayrılması ile mümkün. Aksi halde işe yaramaz.

Örneğin, Veri Tabanı Yönetici yazılımı ilk 15 kayıt için ayırt edici bir unsur bulamazsa, hepsi eşit diye ilk 10 kayıt yerine tüm kayıtlar listeler.

Bundan dolayı da sıralama yan cümleciğinin (ORDER BY) kayıtları bariz olarak sıralayacak şekilde seçilmesi gerekir

BU SQL KOMUTLARINI NASIL KULLANACAĞIM

1. data kontrolü ile:

```
Private Sub Form_Load()
```

```
' * karakteri sonu ne olursa olsun anlamındadır . örnek f ile başlayan isimler gibi  
Data1.RecordSource = "SELECT * FROM kisisel where ad like '" & ad.Text & "*"  
Data1.Refresh  
End Sub
```

2. dbgrid üzerinde sorgu sonucu liste nasıl alınır.

Arkadaşlar hatırlarsanız dbgrid kontrolü veri erişimi olarak data kontrolü kullanır. Bundan dolayı data kontrolüne yapılan sorguların sonucunda dbgrid nesnesi de bu sorgu sonucundaki kayıtları gösterecektir. Liste alırken data kontrolüne sorgu yazmanız yeterlidir.

3. nesneler için SQL kullanma

eğer nesne kullanarak veri tabanı açılmış ise;

örnek:

```
set dosya=openatabase(curdir &"\ogrtakip.mdb")
```

```
set tablo=dosya.openrecordset("kisisel")
```

şimdi yukarıdaki bölme sql komutlarını nasıl koyacağız.

```
Set tablo=dosya.openrecordset ("select * from kisisel where ad like '" & ad.text & "*")
```

Gibi örnek vererek sorunuza cevap verebiliriz.

KONU BAŞLIKLARI

İLİŞKİNİN TANIMI: BİR PROGRAM ÜZERİNDE İLİŞKİLİ TABLO KULLANIMI İLİŞKİLİ TABLOLARDAYENİ KAYIT İLİŞKİLİ TABLOLARDAYENİ SİLME

İLİŞKİNİN TANIMI:

Bir programda bir kişi kayıtları için birden fazla bilgi tutulabilecek kayıtlar varsa bu durumda ilişkili tablo kullanmak zorundayız. Bu tanımı daha da açarsak örneğin öğrenci takip programı için düşünelim. Bir öğrencimiz olsun. Bu öğrencinin kişisel bilgilerini tutuyoruz.

Öğrenci takip mdb dosyasında Kişisel tablosu (öğrenci kişisel bilgilerini tutacağız)

Peki bu öğrenciye ait devamsızlık bilgilerini tutalım dersek nasıl olacak. Alanlarımızı inceleyelim;

Ad	Soyad	Numara	Sınıf	Cinsiyet

Yukarıdaki tabloya göre alanlarımız varsa ve eğer artık devamsızlık da girecek isek tablomuzu şu formata dönüştürelim. Devamsızlık tarihi , gün sayısı ve nedeni diye alanlarımızı ekleyelim.

Ad	soyad	numara	sınıf	cinsiyet	devtarihi	nedeni	Gunsay

Kişisel tablomuzu bu şekilde alanları ekledik. İlk aşamada doğru gibi. Kişiyi ait ad soyad ve devamsızlık bilgisini girdik. Ancak düşünelim, bu kişinin ikici veya daha çok devamsızlıkları olacaktır. Bu durumda yeni bir devamsızlık girmek için yeni kayıt açmamız gerekir. Bu durumda ad, soyad, sınıf ve numara alanlarına da girmemiz gerekecektir. Girelim dersiniz , hem sınıftaki öğrenci sayısı artacak hem de veri tabanında kayıt sayımız artacaktır. Bu durumla ilgili daha çok sakıncalı durum meydana gelecektir. Bir düşünün.

İşte yukarıdaki durumlarda tablolarımızın sayısını artırmamız gerekecektir. Yani kişisel bilgilerimizi bir tabloda devamsızlık bilgilerimizi de ayrı bir tabloda tutacağız. Şimdi bu durumu inceleyelim. Tablolarımız;

Kişisel tablomuz:

Ad	Soyad	Numara	Sınıf	Cinsiyet

Devamsızlık tablomuz:

devtarihi	nedeni	Gunsay

Şimdi yukarıdaki tablolara kayıt girelim

Kişisel tablomuz:

Ad	Soyad	Numara	Sınıf	Cinsiyet
Fatih	Şanlı	1000	11B	Erkek
İhsan	Sevimli	2000	10C	Erkek

Devamsızlık tablomuz:

devtarihi	nedeni	Gunsay
10/11/2000	Rahatsız	1
11/11/2000	Sevкли	2
12/12/2000	sevкли	2
01/01/2005	Sevкли	1
02/02/2004	sevкли	1

Şimdi devamsızlık tabloya kayıt giriyoruz peki devamsızlık bilgileri hangi öğrencilere ait diye sorsam? Cevabımız olmayacaktır. İşte bu durumda işin en can alıcı noktası karşımıza çıkıyor. Nedir bu ?

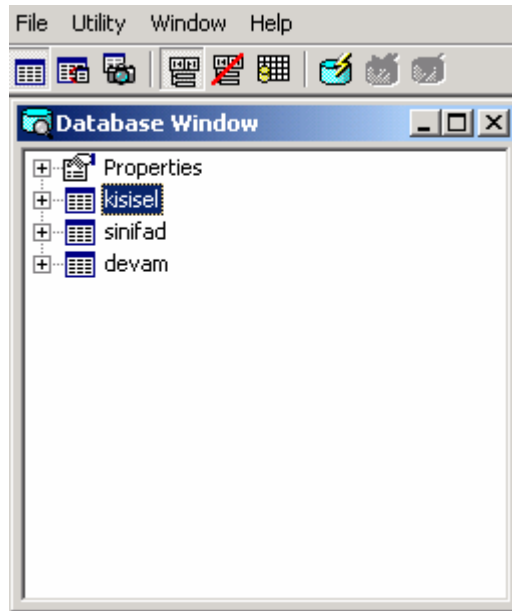
Artık tablomuz sayıları arttı. İki tablomuz var. Ancak bu tablolarımız birbiri ile ilişkili değil. Şimdi bu tablolara arasında bağlantı kurmak zorundayız. Bu bağlantıyı kurmak için tablolarımızı master ve detail tablolar olarak ayıracam. Master tablomuz ana tablomuzdur. O olmazsa diğer tabloların bir anlamı yoktur. Bizim örneğimizde öğrenci bilgisi olmaz ise devamsızlık ve diğer bilgiler de olmaz. O halde ana tablomuz kişisel tablomuzdur.

Master ve detail tablolarımıza göre şimdi ilişkimizi kuralım. İlişki kurmanın yolu master tablomuzdaki bir alanımızı devamsızlık tabloya ekleyeceğiz. Ancak bu alanımız master tablomuz için özel bir alandır. Sadece o kayıta ait olan bilgi olacaktır. Örneğin. Öğrenci programı için kişisel tablomuzdaki numara alanı en uygun alandır. Numara sadece o kişiye aittir. Şimdi devamsızlık tabloya bu numara alanını ekleyelim.

devtarihi	nedeni	Gunsay	Numara
10/11/2000	Rahatsız	1	1000
11/11/2000	Sevklı	2	1000
12/12/2000	sevklı	2	2000
01/01/2005	Sevklı	1	1000
02/02/2004	sevklı	1	2000

Öğrenci Takip Programı üzerinde ilişkili dosyaların incelenmesi

Formumuzun ekran görüntüsü



visdata ekran görüntüsü

Şimdi iki tablomuz kisisel ve devam tablomuzu ilişki tablomuzdur. Bu tablolarımızda numara alanları ortak alan olarak belirledik. Şimdi her ileri geri kayıt ilerlediğimize kişiye ait devamsızlıkları da tablodan sorgulamamız gerekir. Aksi takdirde dbgrid üzerinde kişiye ait kayıtları getirmemiz olanaksızdır. Şimdi sorgumuzu nasıl yapacağız ve hangi olay yordamına yapacağız.

```
Public Sub goster()  
    temizle  
    If rst.Fields("ad") <> "" Then AD.Text = rst.Fields("ad")  
    If rst("soyad") <> "" Then soyad.Text = rst.Fields("soyad")  
    If rst("numara") <> "" Then NUMARA.Text = rst.Fields("numara")  
    If rst("sinif") <> "" Then sinif.Text = rst("sinif")  
    If rst("babaad") <> "" Then babaad.Text = rst("babaad")  
  
End Sub  
  
Public Sub temizle()  
    AD.Text = ""  
    soyad.Text = ""  
    NUMARA.Text = ""  
    babaad.Text = ""  
    sinif.Text = ""  
End Sub
```

Yukarıdaki alt programlarımız text kutularını temizleyip o anki aktif kayıtları gösteren alt programlardır. Şimdi herhangi bir öğrenci bilgisini gösterdiğimiz anda devamsızlık tablosunu da açıp sorgulamamız gerekir. Devamsızlık bilgileri dbgrid üzerinde ise, dbgrid nesnemiz de verileri data1 kontrolünden aldığımızı göre devamsızlık sorgumuzu data1 kontrolüne yapıyoruz.

```
Public Sub goster()  
    temizle  
    If rst.Fields("ad") <> "" Then ad.Text = rst.Fields("ad")  
    If rst("soyad") <> "" Then soyad.Text = rst.Fields("soyad")  
    If rst("numara") <> "" Then NUMARA.Text = rst.Fields("numara")  
    If rst("sinif") <> "" Then sinif.Text = rst("sinif")  
    If rst("babaad") <> "" Then babaad.Text = rst("babaad")  
  
    Data2.RecordSource = "select * from devam where numara=" & NUMARA.Text & ""  
    Data2.Refresh  
  
End Sub
```

Yukarıdaki goster alt programımızı data2 sorgusunu ekledik. Böylelikle dbgrid kayıtları alırken sadece o anki kişiye ait kayıtları gösterecektir.

YENİ KAYIT

Peki yeni bir devamsızlık girdiğimizde ne yapacağız. Yeni devamsızlık bilgisi girerken arkadaşlar devamsızlık bilgisi o anki aktif kayıta ait olacağından devamsızlık tablomuzdaki numara alanı otomatik olarak kisisel tablomuzdaki numara alanına eşit olması gerekir. Yani yeni kayıt ortamında bu numara alanına bilgiyi biz atamalıyız. Örneğin biz dbgrid üzerine numara alanı ekledik. Ama gizli yaptık. Şimdi dbgrid üzerinde yeni kayıt girdiğimiz anda numara bilgisini otomatik olarak alalım..

```
Private Sub DBGrid1_OnAddNew()  
    DBGrid1.Columns(3) = rst("numara")
```

End Sub

Böylelikle artık her yeni devamsızlık bilgisi girildiğinde o anki kişiye ait devamsızlık girmiş oluruz.

KAYIT SİL

Peki silme işleminde bir sorun olacak mı? Düşünün biz silme işleminde sadece kişisel tablomuzdan kayıtları diliyorduk daha önce. Şimdi ise devamsızlık tablomuz da da kayıtlar var. Kişisel tablomuzdaki kayıtları silmeden devamsızlık tablosundan da o kişiye ait kayıtları silmemiz gerekir. Bunu yaparken ister sql sorgu olarak yapın. İsterseniz doğrudan döngü kurarak sırası ile kayıtlara bakın. Eşleşen kayıtları devamsızlıktan sildikten sonra ana master tablonuzdaki öğrenci bilgisini de silebilirsiniz.

ADO VERİ ERİŞİM YÖNTEMİ

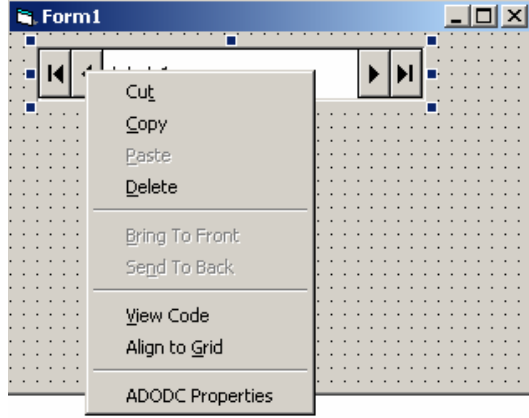
Visual Basic uygulamalarındaki veri tabanı öneminden dolayı Visual Basic 6.0 ile birlikte artık yenilikler arasında ADO (ActiveX Data Object) uygulamasıdır. Bu uygulama kullanırken hatırlarsınız DAO yönteminde olduğu gibi iki tekniğimiz olacaktır. Birincisi ADO Kontrol Elemanı, ikincisi ise nesne tanımlama yolu ile erişim olacaktır

ADO Data Control

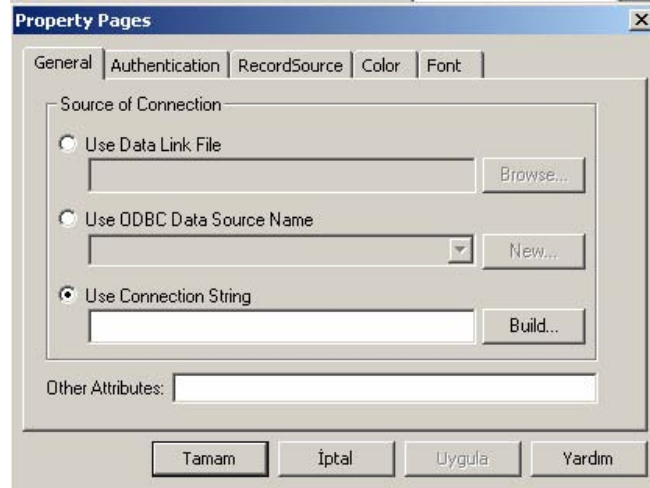
ADO Data kontrol veri kaynakları ile iletişimi sağlar. Daha önce kullanılan DATA kontrol yerine geliştirilmiştir. Kullanıcı arabirimi ve veri arasındaki iletişimi sağlama görevi üstlenmiştir.

ADO Data Kontrolü Toolbox'a eklemek:

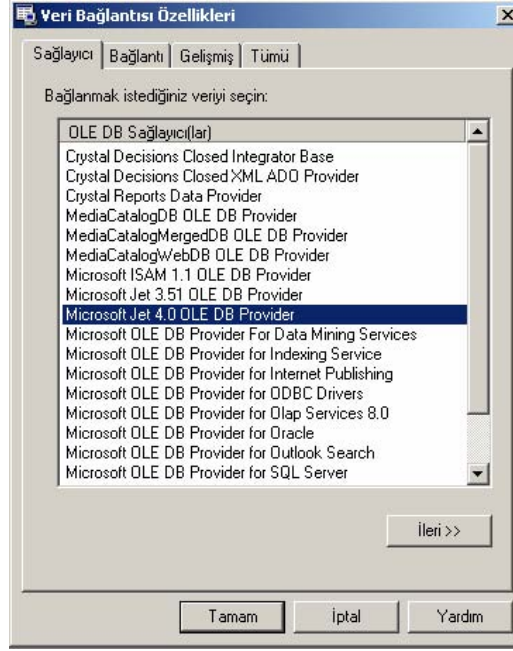
ADO Data Kontrolü proje eklerken **Compenents** bölümünden **Microsoft ADO Data Control 6.0** seçilir. Bu seçimden sonra artık ADO kontrolümüzü form üzerine ekleriz.



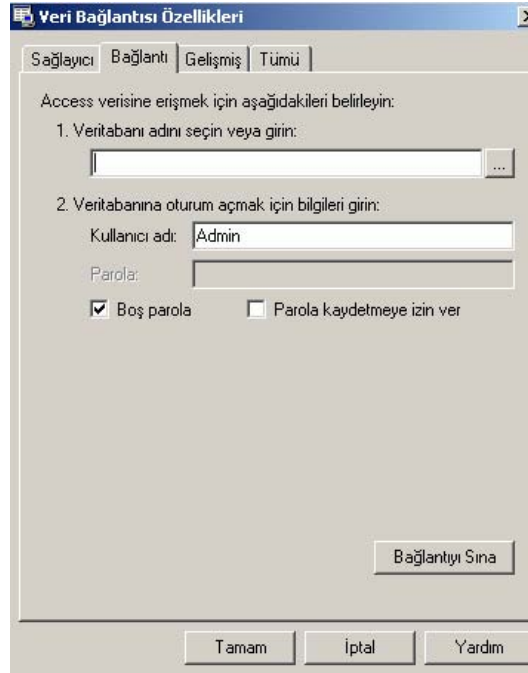
Yukarıdaki form üzerine eklenen ADO kontrolü artık veriye erişim için ayarlanmasını beklemektedir. Bunun için ADO kontrolü üzerinde sağ tıklanır bu bölümden *ADODC Properties* bölümü seçilir.



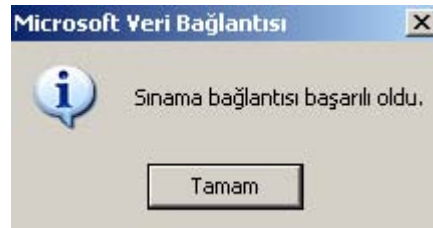
Açılan Yukarıdaki pencerede ise bağlantı için gerekli bilgiler istenmektedir. Bu bölümde şu an için bizim bilgimiz Use Connection String olacaktır. ODBC bölümümüz de diğer iki seçenek anlatılacaktır. **Build** seçeneği tıklanır.



Bu Bölümde Veri erişimi için kullanacağınız ara birimi seçersiniz. Access kullanığzdan dolayı biz bu bölümü Microsoft JET 4.0 OLE DB Provider seçiyoruz. Daha sonraki uygulamalarınızda ODBC tanımladığımızda bu bölümde ODBC drivers seçeneği seçeceğiz. Artık *İleri* Düğmesi tıklandığında;

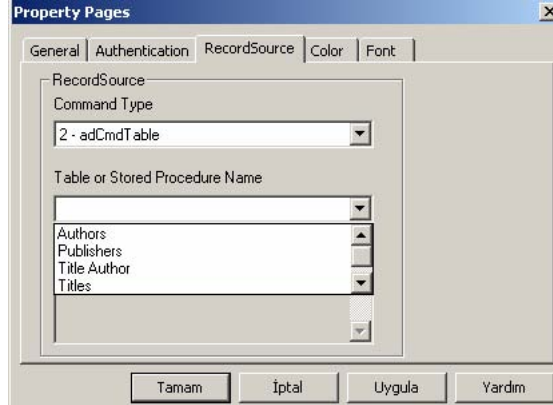


Veri tabanı seçmemiz için yukarıdaki pencere gelir. Bu pencerede Veri tabanı adı seçin bölümünde... seçeneği ile veri tabanı dosyanız seçilir. Daha sonra Bağlantıyı sına seçeneği tıklanır. Bundaki amaç veri tabanına bağlandı mesajını almaktır.



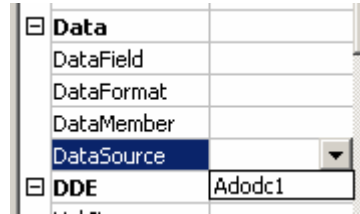
Bundan sonra artık veri tabanında hangi tablo ile ADO Controlü veri iletişimi sağlayacak bilgisini vermeye geldi. Bunun için ise Veri Tabanı bağlandıktan sonra penceremizi tamam düğmesi tıklanarak bir önceki pencereye döneriz. Bu pencerede

General Authentication RecordSource Color Font RecordSource bölümü seçilir.



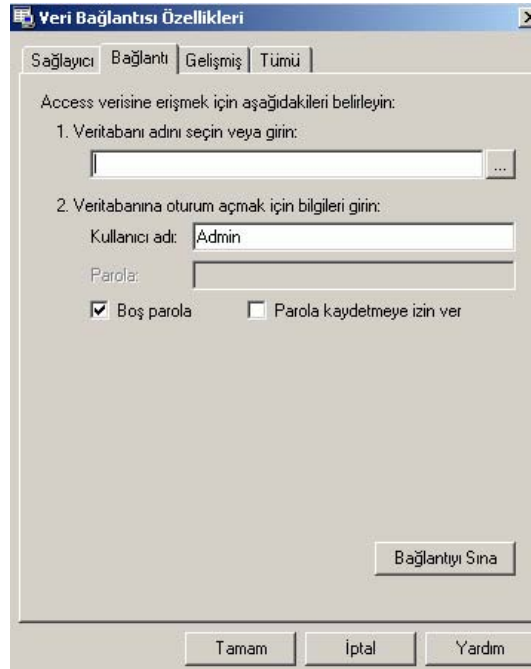
CommanType kısmından Table seçeneği seçilir. Daha sonra artık tablonuzu seçebilirsiniz. Table Or Stored Procedure Name kısmında tablonuzu seçiniz. Bu seçimden sonra artık ADO Controlünüz ile veri erişimi için bağlantı kurdunuz.

Form Üzerindeki Kontrolleri ADO Data Controle Bağlama

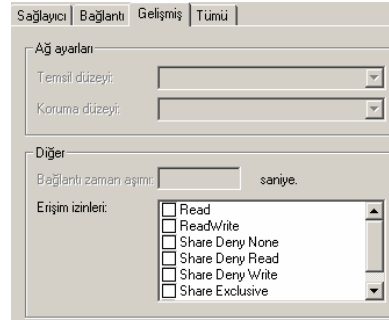


ADO Control ile veri erişimi sağlandıktan sonra textbox,combobox,listbox,grid gibi kontroller ile birlikte kullanılırlar. Ayrıca sonraki dersimizin konusu olan ve ADO için geliştirilmiş datagrid,datalist,datacomco kontrolleride kullanılır. Başlangıç olarak text kutuları DATA kontrolünde olduğu gibi Text kutusunun data kısmındaki Data source ve Data field alanları bağlanır. Grid için ise bir sonraki dersimizi bekleyin.

ADO Data Control Diğer özellikleri



Yandaki pencerede hatırlarsanız ADO kontrolü için veri tabanı seçimi yapmıştık. Bu bölümde Gelişmiş seçeneğimizde ise;



Admodeunknown: İzin düzenlenmemiş

Admoderead: Sadece okumak için açılır.

Admodewrite: Sadece yazmak için açılır.

Admodereadwrite: Hem okuma hem yazma işlemi için açılır.

Admodesharedenyread: Diğer kullanıcıların okuma izni engellenir.

Admodesharedenywrite: Diğer kullanıcıların yazma izinleri engellenir.

Admodeshareexclusive: diğer kullanıcıların bağlantı açması engellenir.

Admodesharedenynone: diğer kullanıcıların izinle bağlantı açması engellenir.

➡ !!! Visual Basic Uygulamalı Ders !!!

Merhabalar ,

Evet Örnek Derslerimize Başlıyalım İsterseniz ;

İlk Olarak Bir Button'a Tıklayınca Belirliyeceğimiz Bir URL'ye Nasıl Gider ?

Cevap ;

```
Private Sub Command1_Click()
```

```
Shell "explorer.exe http://www.wardom.org" ' Bu Wardom.Org 'Gitmesini İstedığınız URL'yi  
Yazabilirsiniz..
```

```
End Sub
```

Bir Exe (Başka Bir Uygulama) Nasıl Çalıştırılır ?

Cevap;

```
Private Sub Command1_Click()
```

```
Shell "Çalıştırmak istediğiniz Program.exe"
```

```
End Sub
```

CD Rom Nasıl Açılır ? Nasıl Kapanır ?

Cevap;

Cd-Rom u açıp kapatmak 2 Command Button Ekleyin Form'un üzerine

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

```
Private Sub Command1_Click()
```

```
mciExecute ("Set CDAudio door Open")
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
mciExecute ("Set CDAudio door closed")
```

```
End Sub
```

Kaynak: [Wardom http://www.wardom.org/showthread.php?t=106918](http://www.wardom.org/showthread.php?t=106918)

Speakerdan beep Sesini Çıkarmak ?

Cevap;

Bir Command Button Ekleyin Form'un üzerine

```
Private Sub Command1_Click()
```

```
Beep
```

```
End Sub
```

Bir ComboBox'a Rakkam Veya Harf Nasıl Eklenir ?

Cevap;

```
Private Sub Combo1_Change()
```

Combo1.AddItem " Değer "
End Sub

Bir Proje Nasıl Exe'ye Çevirilir ?
Cevap;

File Menüünden Make Project1.Exe ile Projenizi Exe'ye Çevirebilirsiniz..

Projeme Nasıl OCX Eklerim ?
Cevap;

Form Üzerinde İken **CTRL + T**'yi Tıklayarak Çıkan Pencere'de Daha Önceden Windows'ta Olan OCX'leri Ekliyebilir , Browse'yi Tıklayarak'ta Kendi OCX'inizi Yükliyebilirsiniz..

Projeme Flash (SWF) Uzantılı Dosyaları Nasıl Eklerim ?
Cevap;

[Cevabın Resimli Anlatımı için Tıklayın](#)

Bir Button'a Tıklayınca **MSGBOX** Nasıl Çıkartılır ?
Cevap;

Private Sub Command1_Click()
MsgBox " Merhaba Dünya "
End Sub

[MSGBOX'un Resimli Anlatımı için Tıklayın](#)

Uyari isareti olan (X) mesaj kutusu nasıl Yapılır ?
Cevap;

MsgBox "Mesaj Buraya!!", vbCritical, "Önemli"

Resim Nasıl Yazdırılır ?
Cevap;

Printer.PaintPicture Picture1.Picture
Printer.EndDoc

Bir Form Nasıl Gizlenir ?
Cevap ;

Form1.Hide

 **Visual Basic Uygulamalı Ders 2**

Merhaba uygulamalı derlserimize devam ediyoruz..

1-) Bir TextBox'un içeriği nasıl temizlenir (silinir) ?

Cevap ;

Bir textbox'un içeriğini silmek için aşağıdaki komutu kullanabiliriz..

Text1.Text = "" ' birden fazla textin içeriğini silmek için ise ;

text1.text = ""

text2.text = ""

vs...

2-) Timer çalışırken interval değeri nasıl değiştirilir ?

Cevap ;

Bir commandbutton koyalım form'umuza ve aşağıdaki komut ile timerin interval değerini değiştirebiliriz..

Timer1.interval = 1000 ' bu 1000 yazan yeri istediğiniz gibi değiştirebilirsiniz..

3-) Program çalışırken bir daha önceden belirlediğimiz bir resim nasıl değiştirilir ?

Cevap ;

Öncelikle forma 2 tane picture ekliyelim ardından içine resimlerimizi atalım programda çıkmasını istediğin resmi gizlemiyoruz ama 2.ci resmi gizliyoruz , program esnasında 2ci picture'nin içindeki resmi göstermek istersek ,

Kaynak: [Wardom http://www.wardom.org/showthread.php?t=115123](http://www.wardom.org/showthread.php?t=115123)

Kaynak: [Wardom http://www.wardom.org/showthread.php?t=115123](http://www.wardom.org/showthread.php?t=115123)

picture1.picture = picture2.picture ' bu komut ile değiştirebiliriz..

4-) HScrollbar'a (Hortizal) değer vermek ?

Cevap ;

Örneğin ses artırıp azaltmak için HScrollbar kullanıyorsunuz peki bir button'a basınca sesi istediğiniz düzeye nasıl getirirsiniz.. ?

HScroll1.LargeChange = 50 ' bu komut HScroll1'in ses değerini o anda %50 yapar , istediğiniz gibi değiştirebilirsiniz..

VScrollbar (Vertikal) ' aynı mantık uygulanabilir..

5-) Çıkışta Açık Formlarınız Kapatmak ?

Cevap ;

Programınız kapanırken açık formlardan bazıları kapanmıyor ve program açık görünüyorsa bunu ana formun unload kısmına ekleyiniz..

Dim Form As Form

```
For Each Form In Forms
Unload Form
Set Form = Nothing
Next Form
End
```

6-) Formu Nasıl Transparent hale getirebiliriz ?

Cevap ;

Formunuza bir modül ekleyin ve aşağıda ki kodu içine ekleyin

```
Private Declare Function SetLayeredWindowAttributes Lib "user32" (ByVal hwnd As Long,
ByVal crKey As Long, ByVal bAlpha As Byte, ByVal dwFlags As Long) As Long
Private Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" (ByVal
hwnd As Long, ByVal nIndex As Long) As Long
Private Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal
hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
```

```
Private Const GWL_EXSTYLE = (-20)
Private Const LWA_COLORKEY = &H1
Private Const LWA_ALPHA = &H2
Private Const ULW_COLORKEY = &H1
Private Const ULW_ALPHA = &H2
Private Const ULW_OPAQUE = &H4
Private Const WS_EX_LAYERED = &H80000
```

```
Public Function MakeTransparent(ByVal hwnd As Long, Perc As Integer) As Long
Dim Msg As Long
On Error Resume Next
If Perc < 0 Or Perc > 255 Then
MakeTransparent = 1
Else
Msg = GetWindowLong(hwnd, GWL_EXSTYLE)
Msg = Msg Or WS_EX_LAYERED
SetWindowLong hwnd, GWL_EXSTYLE, Msg
SetLayeredWindowAttributes hwnd, 0, Perc, LWA_ALPHA
MakeTransparent = 0
End If
If Err Then
MakeTransparent = 2
End If
End Function
```

Formunuza bir tane HScroll Bar ekleyin ve aşağıda ki kodu forma yazın

```
Private Sub Form_Load()
HScroll1.Min = 50
HScroll1.Max = 255
HScroll1.Value = 50
```

End Sub

```
Private Sub HScroll1_Change()  
MakeTransparent Me.hwnd, HScroll1.Value  
End Sub
```

Transparent form değerleri 0 - 255 aralığındadır. Ben 50 - 255 aralığı kullandım görülebilmesi için..

7-) Programımız çalışmasına sistem trayde devam etsin ..

```
Private Type NOTIFYICONDATA  
    cbSize As Long  
    hWnd As Long  
    uId As Long  
    uFlags As Long  
    uCallbackMessage As Long  
    hIcon As Long  
    szTip As String * 64  
End Type  
  
Private Const NIM_ADD = &H0  
Private Const NIM_MODIFY = &H1  
Private Const NIM_DELETE = &H2  
Private Const WM_MOUSEMOVE = &H200  
Private Const NIF_MESSAGE = &H1  
Private Const NIF_ICON = &H2  
Private Const NIF_TIP = &H4  
Private Const WM_LBUTTONDOWN = &H203  
Private Const WM_LBUTTONUP = &H201  
Private Const WM_RBUTTONDOWN = &H206  
Private Const WM_RBUTTONUP = &H204  
Private Const WM_RBUTTONDOWN = &H204  
Private Const WM_RBUTTONUP = &H205  
Private Declare Function Shell_NotifyIcon Lib "shell32" Alias "Shell_NotifyIconA" (ByVal dwMessage As Long, pnid As NOTIFYICONDATA) As Boolean  
Dim nid As NOTIFYICONDATA  
  
Private Sub Command1_Click()  
    nid.cbSize = Len(nid)  
    nid.hWnd = Form1.hWnd  
    nid.uId = vbNull  
    nid.uFlags = NIF_ICON Or NIF_TIP Or NIF_MESSAGE  
    nid.uCallbackMessage = WM_MOUSEMOVE  
    nid.hIcon = Form1.Icon  
    nid.szTip = "Görev çubugunda görünür" & vbNullChar  
    Shell_NotifyIcon NIM_ADD, nid  
    Me.Visible = False  
End Sub
```



```
Private Sub Form_Load()  
YigitsMenu.Visible = False  
Command1.Picture = Me.Icon  
'Command1 Buttonumuza ICONumuzu atiyoruz.  
End Sub
```

```
Private Sub Form_Terminate()  
Shell_NotifyIcon NIM_DELETE, nid  
' Her türlü hataya karsi burda da önlem almamız gerekiyor..  
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
Dim msg As Long  
msg = X / Screen.TwipsPerPixelX  
Select Case msg  
Case WM_LBUTTONDOWN  
Case WM_LBUTTONUP  
Case WM_LBUTTONDBLCLK 'sol tus çift klik  
MenuGoster_Click  
Case WM_RBUTTONDOWN 'sag tus basili  
PopupMenu YigitsMenu, , , MenuGoster  
Case WM_RBUTTONUP 'sag tus basilip bırakildi  
Case WM_RBUTTONDBLCLK 'sag tus çift klik  
End Select  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
Shell_NotifyIcon NIM_DELETE, nid  
'Formumuz kapandiginda Iconumuzu kaldiriyoruz...  
End Sub
```

```
Private Sub MenuGoster_Click()  
'Koydugumuz Menunun Iconken sol ok ile ortaya çıkmasını  
'sagliyoruz...  
Me.Visible = True  
Shell_NotifyIcon NIM_DELETE, nid  
End Sub
```

```
Private Sub MenuKapat_Click()  
'Burda da yine sol okla çıkan menudeki Kapatbuttonu...  
Unload Me  
End Sub
```

Visual Basic'de 3 Bilinmeyenli 3 Denklemin Köklerini Bulan Program

```
Private Sub hesap_Click()  
List1.Clear  
List2.Clear
```

 Döküman Araçları

 [Karma ver: dövləstəyung](#)

 [Kurallara Aykırı Mesajı Bildir](#)

Dim a11, a12, a13, a14, b11, b12, b13, b14, c11, c12,
c13, c14 As Long

a11 = Val(Text1)
a12 = Val(Text2)
a13 = Val(Text3)
a14 = Val(Text4)

b11 = Val(Text5)
b12 = Val(Text6)
b13 = Val(Text7)
b14 = Val(Text8)

c11 = Val(Text9)
c12 = Val(Text10)
c13 = Val(Text11)
c14 = Val(Text12)

'Değerler atandı"

List1.Clear

List1.AddItem " "

Kaynak: [Wardom](http://www.wardom.org/showthread.php?t=109913)

<http://www.wardom.org/showthread.php?t=109913>

List1.AddItem " Denklemler :"

List1.AddItem " " & a11 & "A + " & a12 & "B + " & a13
& "C " & "=" & a14

List1.AddItem " " & b11 & "A + " & b12 & "B + " &
b13 & "C " & "=" & b14

List1.AddItem " " & c11 & "A + " & c12 & "B + " & c13
& "C " & "=" & c14

List1.AddItem " "

List1.AddItem " Katsayılar Matrisi : "

List1.AddItem " " & a11 & " " & a12 & " " & a13 & " "
& ":" & " " & a14

List1.AddItem " " & b11 & " " & b12 & " " & b13 & " "
& ":" & " " & b14

List1.AddItem " " & c11 & " " & c12 & " " & c13 & " "
& ":" & " " & c14

List1.AddItem " "

List1.AddItem " Satır İşlemleri : "

'matris listboxa yazıldı"

Dim ab, ac, ca, cb, ba As Integer

'Matris işlemleri"

a21 = a11

a22 = a12

a23 = a13

a24 = a14

ab = b11 / a11



[Printer Çıktısını Göster](#)



[Sayfayı Emaile Yolla](#)



[Bu Dökümana Abone Ol](#)

```
b21 = b11 - (a11 / a11) * b11
b22 = b12 - (a12 / a11) * b11
b23 = b13 - (a13 / a11) * b11
b24 = b14 - (a14 / a11) * b11
```

```
ac = c11 / a11
```

```
c21 = c11 - a11 / a11 * c11
c22 = c12 - a12 / a11 * c11
c23 = c13 - a13 / a11 * c11
c24 = c14 - a14 / a11 * c11
'Birinci Sütun 0 landı'
List1.AddItem " " & "R2 = R2 - " & ab & "*" R1"
List1.AddItem " " & "R3 = R3 - " & ac & "*" R1"
List1.AddItem " "
```

```
List1.AddItem " " & a21 & " " & a22 & " " & a23 & " "
& ":" & " " & a24
List1.AddItem " " & b21 & " " & b22 & " " & b23 & " "
& ":" & " " & b24
List1.AddItem " " & c21 & " " & c22 & " " & c23 & " "
& ":" & " " & c24
List1.AddItem " "
```

```
If b22 = 0 And c22 = 0 Then GoTo sonsuz Else GoTo 1
MsgBox ("Eğer b22 ve c22 elemanları 0 ise bir
bilinmeyen bulunamayacağından denklemin çözümü
yoktur.")
```

```
1:
If b22 = 0 Then
d21 = b21
d22 = b22
d23 = b23
d24 = b24
'b satırı hafızaya alındı
b21 = c21
b22 = c22
b23 = c23
b24 = c24
'b satırı yerine c satırı yazıldı
```

```

c21 = d21
c22 = d22
c23 = d23
c24 = d24
'c satırı yerine hafızadaki b satırı yazıldı
'c satırı ile b satırı yer değiştirmiş oldu
List1.AddItem " " & "R2<==>R3"
List1.AddItem " "
List1.AddItem " " & a31 & " " & a32 & " " & a33 & " "
& ":" & " " & a34
List1.AddItem " " & b31 & " " & b32 & " " & b33 & " "
& ":" & " " & b34
List1.AddItem " " & c31 & " " & c32 & " " & c33 & " "
& ":" & " " & c34
List1.AddItem " "

```

'-----

Else: GoTo 2

End If

2:

a31 = a21

a32 = a22

a33 = a23

a34 = a24

b31 = b21

b32 = b22

b33 = b23

b34 = b24

'a ve b satırları aynen yazıldı

c31 = c21

c32 = c22 - b22 / b22 * c22

c33 = c23 - b23 / b22 * c22

c34 = c24 - b24 / b22 * c22

'2.satırda da 0 lanma oldu

'matrisimiz satırca indirgenmiş eşelon forma dönüştü

bc = c22 / b22

List1.AddItem " R3 = R3 - " & bc & "*R2"

List1.AddItem " "

List1.AddItem " " & a31 & " " & a32 & " " & a33 & " "
& ":" & " " & a34

List1.AddItem " " & b31 & " " & b32 & " " & b33 & " "
& ":" & " " & b34

List1.AddItem " " & c31 & " " & c32 & " " & c33 & " "
& ":" & " " & c34

List1.AddItem " "

```
List1.AddItem " Katsayılar Matrisi Satırca indirgendi "  
List1.AddItem " "  
List1.AddItem " "
```

'-----

3: If $c_{33} = 0$ Then GoTo sonsuz Else GoTo 4 ' $c_{33}=0$ ise sonsuz satırına git

```
4: a41 = a31  
a42 = a32  
a43 = a33 - c33 / c33 * a33  
a44 = a34 - c34 / c33 * a33
```

```
ca = a33 / c33
```

```
b41 = b31  
b42 = b32  
b43 = b33 - c33 / c33 * b33  
b44 = b34 - c34 / c33 * b33
```

```
cb = b33 / c33
```

```
c41 = c31  
c42 = c32  
c43 = c33  
c44 = c34
```

```
List2.Clear  
List2.AddItem ""  
List2.AddItem " R1 = R1 - " & ca & "*R3"  
List2.AddItem " R2 = R2 - " & cb & "*R3"  
List2.AddItem " "  
List2.AddItem " " & a41 & " " & a42 & " " & a43 & " "  
& ":" & " " & a44  
List2.AddItem " " & b41 & " " & b42 & " " & b43 & " "  
& ":" & " " & b44  
List2.AddItem " " & c41 & " " & c42 & " " & c43 & " "  
& ":" & " " & c44  
List2.AddItem " "
```

```
'a43 0 oldu  
a51 = a41  
a52 = a42 - b42 / b42 * a42  
a53 = a43  
a54 = a44 - b44 / b42 * a42
```

$ba = a_{42} / b_{42}$

$b_{51} = b_{41}$

$b_{52} = b_{42}$

$b_{53} = b_{43}$

$b_{54} = b_{44}$

$c_{51} = c_{41}$

$c_{52} = c_{42}$

$c_{53} = c_{43}$

$c_{54} = c_{44}$

List2.AddItem " R1 = R1 - " & ba & "*R2"

List2.AddItem " "

List2.AddItem " " & a₅₁ & " " & a₅₂ & " " & a₅₃ & " "
& ":" & " " & a₅₄

List2.AddItem " " & b₅₁ & " " & b₅₂ & " " & b₅₃ & " "
& ":" & " " & b₅₄

List2.AddItem " " & c₅₁ & " " & c₅₂ & " " & c₅₃ & " "
& ":" & " " & c₅₄

List2.AddItem " "

$a_{61} = a_{51} / a_{51}$

$a_{62} = a_{52}$

$a_{63} = a_{53}$

$a_{64} = a_{54} / a_{51}$

$b_{61} = b_{51}$

$b_{62} = b_{52} / b_{52}$

$b_{63} = b_{53}$

$b_{64} = b_{54} / b_{52}$

$c_{61} = c_{51}$

$c_{62} = c_{52}$

$c_{63} = c_{53} / c_{53}$

$c_{64} = c_{54} / c_{53}$

List2.AddItem " R1 = R1 / " & a₅₁

List2.AddItem " R2 = R2 / " & b₅₂

List2.AddItem " R3 = R3 / " & c₅₃

List2.AddItem " "

List2.AddItem " " & a₆₁ & " " & a₆₂ & " " & a₆₃ & " "
& ":" & " " & a₆₄

List2.AddItem " " & b₆₁ & " " & b₆₂ & " " & b₆₃ & " "
& ":" & " " & b₆₄

List2.AddItem " " & c₆₁ & " " & c₆₂ & " " & c₆₃ & " "
& ":" & " " & c₆₄

Kaynak: [Wardom](http://www.wardom.org/showthread.php?t=109913)

<http://www.wardom.org/showthread.php?t=109913>

List2.AddItem " "

```
List2.AddItem " Denklem Satırca indirgenmiş "  
List2.AddItem " Eşelon forma geldi "  
List2.AddItem ""
```

'katsayılar matrisi satırca indirgenmiş eşelon forma geldi

```
a = a64  
b = b64  
c = c64  
'a b ve c değerleri bulundu
```

```
List2.AddItem " Cevaplar :"  
List2.AddItem " A= " & a  
List2.AddItem " B= " & b  
List2.AddItem " C= " & c  
List2.AddItem ""
```

GoTo bitti

```
sonsuz:  
MsgBox ("Denkleminizin Tek bir çözümü yok ")  
bitti:  
End Sub
```

```
Private Sub kapat_Click()  
Unload Form1  
End Sub
```



Shell32.dll'in Yararları

İlk önce Project-References den shell32.dll dosyasını ekleyin.. bilmeyenler için Shell32.dll Windows\system32 dedir.

```
'declare kısmına  
Dim opt As New Shell  
Dim klasor As Folder
```

```
With klasor.Items.Item  
Text1.Text = .Path  
Text2.Text = "İsim: " & .Name & vbCrLf & _  
"Klasör Türü: " & .Type & vbCrLf & _  
"Konum: " & .Parent & vbCrLf & _  
"Son Değişiklik: " & .ModifyDate & vbCrLf
```

End With
End Sub

```
Private Sub Command2_Click()  
opt.ControlPanelItem "inetcpl.cpl" 'Internet seçeneklerini açar  
'desk.cpl 'Masaüstü özellikleri  
'sysdm.cpl 'Sistem özellikleri  
'appwiz.cpl 'Program ekle kaldır  
'daha fazlası için windows\system32 de *.cpl diye aratın  
End Sub
```

```
Private Sub Command9_Click()  
Opt.open "c:\rifocan.doc" 'Word Dosyası Açma Uzantısını Değiştirerek Başka Şeyler'de  
Çalıştırılabilir ..  
End Sub
```

```
Private Sub Command3_Click()  
opt.FindFiles 'Dosya arama  
End Sub
```

Kaynak: [Wardom http://www.wardom.org/showthread.php?t=110981](http://www.wardom.org/showthread.php?t=110981)

```
Private Sub Command4_Click()  
opt.FileRun 'Çalıştır penceresi  
End Sub
```

```
Private Sub Command5_Click()  
opt.Open "http://www.wardom.org"  
'opt.Open "mailto:mailadresiniz@mailadresiniz.com"  
'opt.open app.path  
End Sub
```

```
Private Sub Command6Click()  
opt.SetTime  
End Sub
```

```
Private Sub Command7_Click()  
opt.MinimizeAll 'Programı tam Ekran Yap 😊  
End Sub
```

```
Private Sub Command8_Click()  
If MsgBox("Windows u kapatacak mısınız?", 32 + 4) = vbNo Then Exit Sub  
opt.ShutdownWindows 'Windows Kapat..  
End Sub
```
